Neural Computing and Applications
https://doi.org/10.1007/s00521-024-10370-4

ORIGINAL ARTICLE

A new bi-level deep human action representation structure based on the sequence of sub-actions

Fardin Akhlaghian Tab[1] · Mohsen Ramezani[2] · Hadi Afshoon[1] · Seyed Amjad Seyedi[1] · Atefeh Moradyani[1]

Received: 31 January 2022 / Accepted: 8 August 2024

Abstract

Human action recognition is applicable in different domains. Previously proposed methods cannot appropriately consider the sequence of sub-actions. Herein, we propose a semantical action model based on the sequence of sub-actions. A technique is used to segment actions on the time axis based on body movements via an energy diagram. After dividing actions into sub-actions, a novel bi-level deep structure is used to extract their features. Then, the sequence of sub-action features is modeled by a deep network to create the action model. As extracted sub-actions have fewer variations in execution manner, their representation is more stable, and modeling their sequence would be an efficient model. Experimental results on UCF-YouTube, UCF-Sport, and Human Motion DataBase (HMDB) datasets indicate the sustainable performance of this method. Overall, the accuracy of the proposed method is 0.972 on average, while the value for the second-best method is 0.925.

Keywords Human action · Deep feature · Sustainable model · Spatiotemporal model

1 Introduction

Today, human action recognition is used in different domains such as content-based video retrieval, surveillance cameras, and patient movement monitoring [1, 2]. Body movements constitute human actions such as volleyball spike, dive, wave, and sit up [3, 4]. Thus, body movements can be modeled as representative of an action and then learned by a classifier and used to perform the action recognition task [5, 6]. To represent an action, certain features are extracted during the body's execution of such action. The performance of the action recognition method depends directly upon the extracted features, which can be categorized as traditional or deep [7, 8]. Traditional features used in action analysis tasks (e.g., action recognition and action retrieval) can be further categorized as local or global [9–11]. Global features seek to represent the human body gestures via the shape of the human body. Local features, however, seek to model important movements that occur during actions. Global features are sensitive to noise and the appearance of the human, but local features cannot consider the shape of the body or the global information of actions. Thus, traditional features cannot achieve a proper and stable performance [12].

Because convolutional neural networks are successfully used in different image processing applications, they are also used to outperform the action recognition methods [13]. These networks are used to extract deep features and learning discriminative representations considering both global and local body changes. Convolutional neural networks (CNNs) extract spatial features of actions and learn

✉ Mohsen Ramezani
m.ramezani@uok.ac.ir

Fardin Akhlaghian Tab
f.akhlaghian@uok.ac.ir

Hadi Afshoon
h.afshoon@uok.ac.ir

Seyed Amjad Seyedi
amjadseyedi@uok.ac.ir

Atefeh Moradyani
atefeh.moradyani@uok.ac.ir

[1] Department of Computer Engineering, Faculty of Engineering, University of Kurdistan, Sanandaj, Iran

[2] Department of Computer Science, University of Kurdistan, Sanandaj, Iran

Published online: 16 October 2024

them but ignore the temporal features of actions. Three-dimensional CNNs are used to extract both the spatial and the temporal information of actions. Additionally, autoencoder (AE) is used in some methods to extract temporal information of actions in an extra step [14]. Nonetheless, some studies have used long short-term memory (LSTM) to this end based on the features learned through a CNN [15]. Two-stream architecture is another solution in which spatial and temporal features of actions are separately extracted in two streams and combined in different manners [16].

Different action recognition methods use multiple frames, namely keyframes, in their procedure to control the volume of computations, because there is a vast number of frames in each video [3, 5]. Keyframes are usually selected randomly, which may result in no frames being picked up from an important part of the video. As a practical example, considerable overlap exists between the extracted features of diving and jumping actions, because in most videos, diving relates to the actor jumping and then falling into the water in a short time. Thus, failure to pick up a sufficient number of keyframes from this part of the dive action will lead to dive videos being classified as jump ones. Moreover, the sequence of the action parts (i.e., segments on the time axis), which is ignored in other methods, can be a proper feature for discriminating them. Each action can be segmented on the time axis into sub-actions, where the sequence of their features can be a discriminator component.

This paper proposes a novel semantical representation method based on modeling the sequence of sub-actions. Our method pays attention to all sub-actions equally, regardless of their duration. Furthermore, this method models the sequence of sub-actions as a discriminative model. In this method, deep spatiotemporal features of sub-actions are extracted independently and subsequently fused to create the final action representation. Clearly, sub-actions are simpler than an action from the viewpoint of execution manner, and the extracted features of sub-actions would be more stable than those of actions. In other words, the representation of a sub-action would have less diversity. Thus, segmenting each action into sub-actions and fusing the representation of these sub-actions helps improve human action recognition. Overall, we propose a new semantical action representation framework that comprises the three main steps of segmenting stream (extracting sub-actions), sub-action modeling, and modeling the sequence of sub-actions, respectively. The main contributions of this paper are as follows:

1. Each video is divided semantically into sub-actions (i.e., segmenting video on the time axis), and the spatiotemporal features of sub-actions are extracted for modeling their sequence. As sub-actions have fewer variations in execution manner, the final model would be more stable than other methods.
2. A novel method is utilized for detecting sub-actions based on a diagram (i.e., energy diagram) representing the movements per frame. The energy line is calculated, which inspires the response function (by Dollar [17]), the moments of that line are near zero, and the action must be broken to sub-actions.
3. A new bi-level deep representation method is proposed to use sub-actions' features to create the action model. The main goal of this structure is to model the action based on the sequence of deep features of sub-actions. The first level relates to extracting deep features of sub-actions, and the second level relates to creating the final model based on their sequence.
4. In addition to evaluating the proposed structure in human action recognition task, this structure is also evaluated in content-based retrieval tasks (i.e., action retrieval). In fact, we sought to evaluate the extracted sub-actions and their raw spatial features. The retrieval task lacks the training of a learning algorithm and can be considered as a hard task for evaluating a model.

The remainder of this paper is organized as follows. In section two, related works are introduced and described in detail. The proposed method is presented in section three. The proposed method is evaluated in section four by describing the experimental results on the datasets used. Finally, the article is concluded in section five.

2 Related works

In computer vision, human action recognition can be considered a major field of research in which different methods are proposed for reducing the effect of challenges such as various action categories, different body appearances during playing one action by different persons, occlusion, etc. Early methods seek to describe body poses to represent human actions as global features. These features can be used to analyze the appearance or shape of the human body [18, 19]. There are different types of global features such as low-level mesh features [2, 20], extracted human-centered regions [21], and binary silhouettes [22, 23]. Lin et al. [21] sought to cluster the extracted silhouettes to create a sequence of action prototypes that can be used for different tasks such as action recognition or retrieval. Shao et al. [23] proposed a method for learning the statistical distribution of body poses through the extracted silhouettes. Shao and Chen [24] also created a histogram of body poses from the extracted silhouettes to represent human actions for different action analysis tasks.

Global feature-based methods have proper performance on clean datasets, but problems in background subtraction, tracking, and occlusion make them unsuitable for real applications [25].

Local features are introduced to represent local movements that occur in the stream as robust to noise and variation features [26]. In fact, each local feature is the description of a motion that is spatially and temporally enclosed in a 3D patch (i.e., cuboid) [27, 28]. The patches are considered around specific points that have the most important movements (i.e., spatiotemporal interest points or STIPs). The local feature-based methods include two main steps, namely STIP detection and patch description, respectively [24]. There are different STIP detection techniques, the most used of which are Dollar detector [29, 30], SIFT detector [31, 32], and 3D Harris [19]. Moreover, most descriptors used in various studies are gradient descriptor [33, 34], Histogram of Oriented Gradient 3D (HOG3D) [35, 36], 3D-Visual-Word [37], HOG/HOF (Histogram of Optical Flow) [38, 39], motion pattern descriptor [1], and Laplacian pyramid coding [19]. In order to achieve better results, some local and global features were recently used in parallel by Afza et al. [40] to recognize human actions. In this method, HOG, geometric, and silhouette features are extracted in parallel and fused to achieve a proper action representation. It should be noted that the total performance of this method is not better than that of recently introduced methods which extract features by convolutional neural networks. The main challenge of local feature-based methods lies in their limited performance on clean datasets and their inability to accurately model human body poses. Recent approaches have attempted to address this by combining local and global features to improve action recognition. However, these methods still lag behind those that utilize convolutional neural networks (CNNs) for feature extraction. CNNs have demonstrated superior performance by autonomously learning discriminative features from raw data.

CNNs are widely used in different recently developed methods for extracting the features of captured actions [13]. Deep networks seek to extract features, to represent action, and to classify actions in an integrated framework. In some methods, three-dimensional convolutional neural networks (3DCNNs) are directly used to extract the spatiotemporal model of the action and classify it [41]. In other methods, autoencoders are used to reduce the representation dimensionality or create the temporal model of the extracted spatial features [14, 42]. Utilizing 3DCNNs forces the methods to set many parameters and perform high computations. To tackle these challenges, Fan et al. [41] proposed a 3D convolutional neural network, namely 3D-1 bottleneck residual block, that performs properly with significantly less time needed compared with other 3DCNN-based methods. Singh et al. [16] also utilized 3D residual networks similar to 2D ResNet but with 3D convolutional kernels. In this study, 2D ResNets were used to extract the spatial information of actions, and their combination with the extracted 3D features forms the final action model. Methods that employ 3D networks for feature extraction exhibit high computational complexity and require a large number of parameters to be configured.

In still other methods, 2D CNNs are used to extract spatial features, and deep structures (e.g., AE) are used to create the final spatiotemporal model. Ullah et al. [14] proposed a framework for extracting spatial features using a deep network. Their method creates the final temporal model using autoencoders. In their study, VGG16 as a commonly used pre-trained CNN was incorporated for extracting the spatial model of the stream of keyframes for each video. Then, the extracted spatial models were concatenated and given to an AE to create the final representation. The vectors were finally given to a quadratic SVM (support vector machine) to classify the actions. Farrajota et al. [43] also tend to use the information of human body joints extracted by CNNs for human action recognition tasks. In this method, several networks are combined in a new network structure (namely the residual autoencoder network), including convolutional layers, residual blocks, max pooling layers, and AEs, to improve the overall performance. After spatial information is extracted using the proposed network, it is given to an LSTM network to discriminate the temporal model of actions. As another method, human actions are considered from different views; in it, some keyframes are initially detected and two different features are extracted [44]. The first feature relates to spatial information of the action extracted using the VGG19 network. The second one relates to multi-view features that are calculated by horizontal and vertical gradients. These features are then combined and used by the naïve Bayes classifier to perform the recognition task.

Muhammad et al. [15] used a CNN without pre-trained weights to represent human action. The proposed CNN has only one pooling layer instead of multiple layers of different sizes, which may lead to information loss. To learn features, this method uses residual blocks that each block includes dilated convolutional neural network. It ignores improper features to create the final model of the action by incorporating a deep convolutional neural network (DCNN). Finally, the temporal model of the action is created by a bidirectional LSTM network. In another study, Wang et al. [45] introduced a parameter-free spatial–temporal pooling block for modeling action to achieve more accurate recognition results. Javidani and Mahmoudi [46] proposed a light model to be trained with insufficient training data, in which three-dimensional video is divided into 1D in temporal axis on top of 2D in spatial ones. In this method, pre-trained networks are used to model spatial dimensions, and the time axis is classified based on the spatial representation. Pirri et al. [47] introduced an end-to-end network to anticipate and forecast actions with memory. Saif et al. [48] introduced a convolutional long short-term deep network (CLSTDN) for recognizing human action-based intentions. CLSTDN combines a convolutional neural network (CNN) and a recurrent neural network (RNN). The pre-trained Inception-ResNet-v2, with 164 deep layers, is used for CNN to extract spatial features. The recurrent neural network, specifically long short-term memory (LSTM), captures temporal features. This method cannot outperform the state-of-the-art methods significantly. The primary issue with CNN-based methods lies in their limitations to effectively capture the complete range of local or global changes in actions. While certain methods utilize 2D CNNs to extract spatial features and incorporate deep structures such as autoencoders (AE) to construct the spatiotemporal model, others leverage human body joint information or multi-view features obtained from CNNs. However, these approaches may encounter challenges such as information loss, insufficient training data, or limited ability to model complex temporal dynamics. To address these limitations, recent studies have introduced two-stream methods that combine deep features with traditional features to enhance the representation and modeling of actions.

Two-stream frameworks that are used for human action recognition seek to use two types of features to achieve proper representation for action [49, 50]. In some of these methods, one stream relates to traditional, whether global or local, features for extracting temporal features of action, and another one relates to deep features for extracting its spatial features. Dai et al. [51] utilized optical flow and CNN-based features in their introduced framework. Here, the deep spatial information of keyframes is extracted to be used in addition to the optical flow features. Then, two LSTM-based frameworks are used to extract temporal information of the actions. Zhao et al. [52] proposed an LSTM-based framework that uses optical flow features and deep spatial information of keyframes to represent an action. This method uses the VGG16 to extract the spatial information of keyframes, and these features are given to an LSTM. The LSTM network computes the final model of this stream. In the other stream, a DenseNet is used to create the temporal model based on the sequence of optical flow features. The outputs of these streams are then fused, and a SVM classifier recognizes the action based on the fused features.

In a recently developed method, Zong et al. [49] used three interactive streams (i.e., appearance, motion, motion saliency streams) in Multiplier-ResNets (residual networks). This method seeks to consider salient motion information to achieve accurate action representation rather than two-stream methods that only capture the spatial and temporal information of actions. Interestingly, this method seeks to ignore the fusion step and considers the connections between streams to predict the final label. Tu et al. [53] proposed a two-stream method that considered some regions in the video instead of the whole frame to represent human action. In one stream, a region corresponding to the appearance and body motions of the human is considered as the region of interest. Then, the optical flow features of this region are extracted. In the other stream, the most important body part with the major movements is considered in a region to be used for extracting motion saliency. The features extracted by these streams are used in a CNN-based framework, the output of which is the final spatiotemporal model.

Ma et al. [54] also used regions of interest and human poses to represent the action as a proper method. In this method, detected regions are given to a CNN, and the output of the pooling layer is used instead of the output of the fully connected layer as the spatial features of actions. Then, an encoder is used to extract temporal features and create the final model based on the sequence of extracted spatial features. Fang et al. [55] proposed an action recognition model, 3 s-STNet, with a three-stream spatial–temporal network architecture. This model captures spatial–temporal features by integrating information from a spatial–temporal graph, RGB appearance image, and a tree–structure–reference–joints image (TSRJI), all derived from human skeleton data. In the first stage, spatial–temporal features are extracted using a spatial–temporal graph convolutional network (ST-GCN) and two Res2Net-101 networks. In the second stage, the features from these streams are fine-tuned and fused to exploit their complementary nature and diversity, enhancing the model's ability to recognize actions effectively. However, the aforementioned multi-stream methods encounter several challenges, including fusion issues, information loss, inadequate representation, high complexity and computational cost, and neglect of salient motion information. The intricate fusion of features from different streams can detrimentally impact performance. Additionally, focusing solely on specific regions or body parts may result in information loss, incomplete representation, and an inability to capture crucial motion information.

Most of the current methods use some keyframes instead of all frames to reduce the volume of computations. Utilizing keyframes leads to some frames which may relate to important body movements being ignored, and ultimately, the created model cannot properly represent the whole action. This challenge leads to considerable overlap between human actions like diving and jumping which have similar motions and body movements during action execution. Moreover, the sequence of motions and movement (namely sub-actions) is ignored in almost all methods. It should be noted that an action can be defined as a sequence of motions. Practically, if a person sees a special motion, he/she can predict the subsequent motion that the actor will perform. Thus, the sequence of actions can be an important feature that must be considered in action modeling. Thus, segmenting action on a time axis (i.e., dividing action into sub-actions) can help model the sequence of sub-actions which leads to a semantical model. Moreover, detecting sub-actions and selecting keyframes from each one individually prevents any motion or body movement from being ignored, which decreases the overlap between similar action categories. To this end, some methods directly identify a discriminative subset of the most useful and redundancy-constrained features or novel dimensionality reduction and feature extraction techniques for different tasks such as face recognition and object-based nonparametric clustering [56, 57]. Li et al. [58] proposed a context-based tandem network (CTNet) based on exploring the spatial contextual information and the channel contextual information interactively for semantical spatial image segmentation. Khan et al. [59] also proposed a coarse-to-fine pupil localization method using multistage convolution to find its optimal horizontal and vertical coordinates. In another method, a convolution method is deployed for reading analogue aircraft instruments facilitated by the Flight Guardian [60]. The performance of these methods suggests that relying solely on a limited set of selected features is insufficient in effectively addressing the challenges of action analysis tasks. Moreover, these methods tend to overlook the sequential nature of motions within an action, and the absence of sub-action modeling diminishes the representation capacity while increasing the overlap between action categories.

Singh [61] proposed a novel weakly supervised method for recognizing and localizing spatiotemporal actions in untrimmed videos. The approach utilizes a graph representation with clip-level annotations, setting it apart from previous techniques. It identifies local actions within action clips and employs a deep multiple instances ranking framework to handle variations within and similarities between classes. But, the final result of this method is not better than the previously introduced methods. However, it can be considered as a graph-based research that can attract attention in the future. Furthermore, the methods that use single-modal data lack adequate information, and the spatiotemporal features of videos and skeleton sequences cannot be representative enough. To this end, Wu et al. [62] proposed a novel map for depth features using descriptors that indicate the spatiotemporal information of skeleton joints. Finally, the HOG is used to represent these features and then an SVM classifier is used for recognizing the action. Totally, current methods cannot be used for real applications such as human–robot interactions because of the lack of the ability to handle before-executed actions and concatenating the sequence of actions to define the semantics. Some methods like the method of Shen et al. [63] are introduced to this end by using continuous skeletal data and making a difference between operational and transitional actions.

3 Proposed method

In the current paper, a video stream is segmented on a time axis to detect the sub-actions for semantically modeling a human action based on the sequence of deep features of sub-actions. As sub-actions are simpler than the main action from the viewpoint of the execution manner, one sub-action would have similar features in different videos. Thus, modeling action based on the sequence of sub-actions would be a stable method. In fact, a semantical representation is used here to model the action. It should be noted that each action is a unique sequence of sub-actions, and this method would achieve significantly better results. Here, sub-actions are detected using an energy diagram that models the body movements in each video. After dividing videos into sub-actions, some keyframes from each sub-action are extracted so as to consider all important body movements during the action. Then, deep spatial features of keyframes of a sub-action are extracted using a CNN, and a bi-level deep structure is used to calculate the temporal features of each sub-action and finally model the sequence of sub-actions features. The sequence of sub-action features is the basis of the recognition task by a classifier in the final step of the framework.

The current framework is applied to individual videos which capture only one action executed by humans. There are three well-known video datasets used for evaluating action analysis methods: UCF-Sport, UCF-YouTube (UCFYT), and HMDB. The UCF-Sport dataset consists of 150 real videos from 9 action categories captured from sports scenes. There are different actors, backgrounds, viewpoints, and scenes in the videos of this dataset. UCF-YouTube is another used dataset with different actors, backgrounds, viewpoints, and scenes. This dataset contains 1600 videos in 11 action categories captured from real actions on YouTube videos. The action categories are basketball shooting, biking/cycling, diving, golf, swinging, horse riding, soccer juggling, tennis, diving, trampoline jumping, and volleyball spiking. As a large-scale dataset, HMDB with 6849 real clips in 51 action categories is used in this paper. Here, 2241 action-based videos from the HMDB dataset are gathered into 19 action categories (i.e., handstand, golf, jumping, flic flac, pull up, kick ball, clap hands, climb stairs, dive, fall on the floor, push up, run, sit down, sit up, somersault, stand up, turn, walk, and wave) to be used in evaluations. As another subset of HMDB, the JHMDB dataset is also used in this paper to evaluate the proposed method in comparison with state-of-the-art methods. The JHMDB dataset, including 923 videos in 21 action categories, is a challenging dataset because of the variety of categories and the limited number of training videos. In the rest of this section, the proposed method is described in detail (Table 1).

3.1 Segmenting video and extracting deep spatial features of sub-actions

As shown in Fig. 1, sub-actions of the video streams are detected in the first step (i.e., segmentation on the time axis). Action is divided into the sub-actions in which a body movement (motion) is executed by the actor. This segmentation helps identify all motions that may be executed in a short time. In continuation, these sub-actions are considered in keyframe selection, deep spatial feature computation, and in extracting the temporal model.

Unlike other methods which randomly select keyframes, our method seeks to select those keyframes which relate to all motions regardless of their duration to be used for modeling action properly. An action may be executed in different ways, which is referred to as different body poses or time of execution. These differences significantly affect all traditional and CNN-based methods, especially local feature-based ones. Human action, however, is in fact a sequence of sub-actions which are simpler and have fewer variations in duration. Thus, considering the sequence of sub-action models can lead to a more stable method against execution variations. Other methods may use fewer frames from the sub-actions that are shorter than others, because the sub-actions may have different execution times in different actions. Regardless of such variables, keyframes must be selected from all sub-actions/motions to prevent losing the information of short sub-actions. In fact, a motion, whether it is executed in a long or short time, must have its share (i.e., the number of keyframes) in creating the final model of the action. To this end, n keyframes is selected from each sub-action to be used in modeling sub-actions and the action itself. Thus, constituting sub-actions must be initially detected.

Here, some sub-actions would be detected, each of which would capture a motion of the action. In order to detect sub-actions, an important point with most of the movements is considered, and its response values are used to divide the action into the constituting sub-actions. The response function (i.e., R-value) is calculated for each point in the stream using the Dollar method [17] as follows:

$$R = (I * g * h_{\mathrm{ev}})^2 + (I * g * h_{\mathrm{od}})^2 \tag{1}$$

where I indicates the frame, and g indicates the kernel of Gaussian. Moreover, h_{ev} and h_{od} are Gabor filters that are calculated as follows:

$$h_{\mathrm{ev}}(t;\tau,\omega) = -cos(2\pi t\omega)e^{-t^2/\pi^2} \tag{2}$$

$$h_{\mathrm{od}}(t;\tau,\omega) = -\sin(2\pi t\omega)e^{-t^2/\pi^2} \tag{3}$$

The most important point with the highest R-value is selected and tracked during the video stream. The R-values of the point are gathered on a diagram known as the motion diagram which is mapped to the energy diagram by applying the continuous wavelet transform (CWT) for easier computations in the next step by representing the motion diagram in an over-complete form, considering the characteristics of the wavelet. The energy diagram is used to divide the action into sub-actions by a threshold value T. As shown in Fig. 2a, the value of the energy line for each video frame is changing, where it would be near zero the executing motion changes. This point indicates the frame between two segments as the point between two sub-actions. Here, the value corresponding to each frame in the energy line is considered to be C, and the number of all frames is shown by F. Thus, a separator value is calculated as follows (minimum values for this separator indicate the motion changing candidates):

$$\text{separator - value} = \left(\frac{\sum_{i=1}^{F}|C_i|}{F}\right)/\mathrm{T}. \tag{4}$$

where T is a threshold value and is empirically detected. This method segments actions on the time axis to achieve the sub-actions. Hereafter, keyframes are selected from all sub-actions equally. Keyframes are picked up in a distributed form in each sub-action (as shown in Fig. 2b). These keyframes contain all important human poses when the actor executes the sub-action. Then, deep features of

Table 1 Brief introduction of used datasets

	Dataset			
	UCF-Sport	UCFYT	JHMDB	HMDB
Number of videos	150	1600	923	6849
Number of action categories	10	11	21	51
Size of dataset	1.7G	997 M	380 M	2G

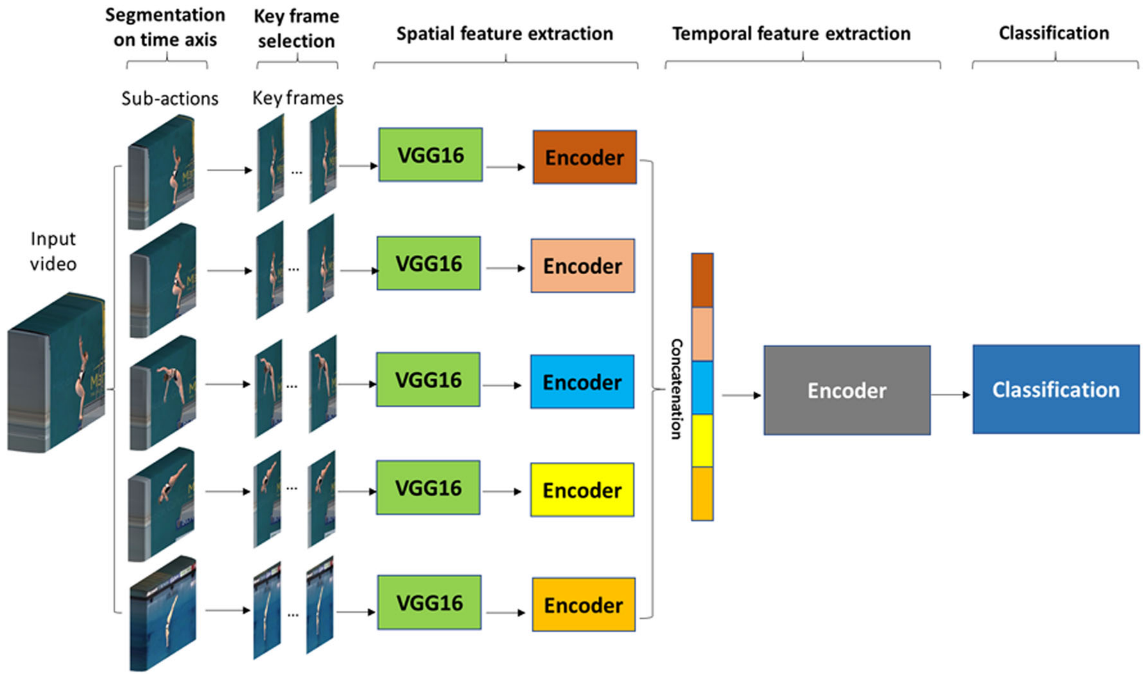
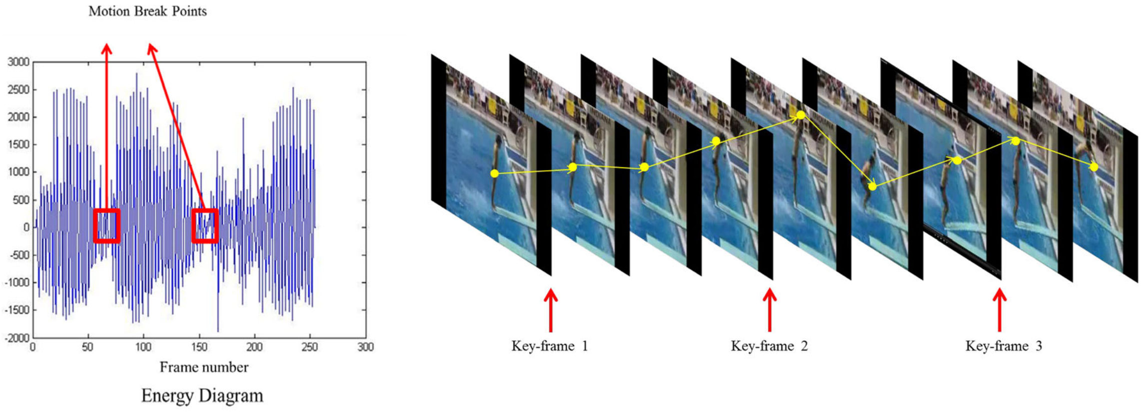


Fig. 1 Framework of the proposed semantic-based deep spatial feature extraction



(a)

(b)

Fig. 2 **a** Breaking an action into motions. **b** Selecting the keyframes of the *first* sub-action of the diving action

these keyframes can be extracted as deep spatial features of sub-actions.

After detecting keyframes, deep spatial features of the sub-actions are extracted using a pre-trained deep network. Here, VGG-16 is used to extract the deep spatial features of each keyframe. The output of the fully connected FC8 layer is the spatial feature of the input keyframe. For each keyframe, a 1 × 1000 vector is created by the VGG-16 as the spatial feature of the frame. As n keyframes are extracted from s sub-actions, s matrices with $n \times 1000$ dimensions are computed as the deep spatial features of an action. These matrices will be used to create the final model of each action.

3.2 Computing final models and performing classification

To create the action model, the temporal information of the extracted deep spatial features for each sub-action must initially be computed, and their sequence model is used as the action representation. To this end, two levels of autoencoders are considered to extract the temporal features of detected sub-actions and compute the action model based on the sequence of sub-action features (See Fig. 3). In fact, the latent layer of the second-level AE is the spatiotemporal model of the action in the stream video. This

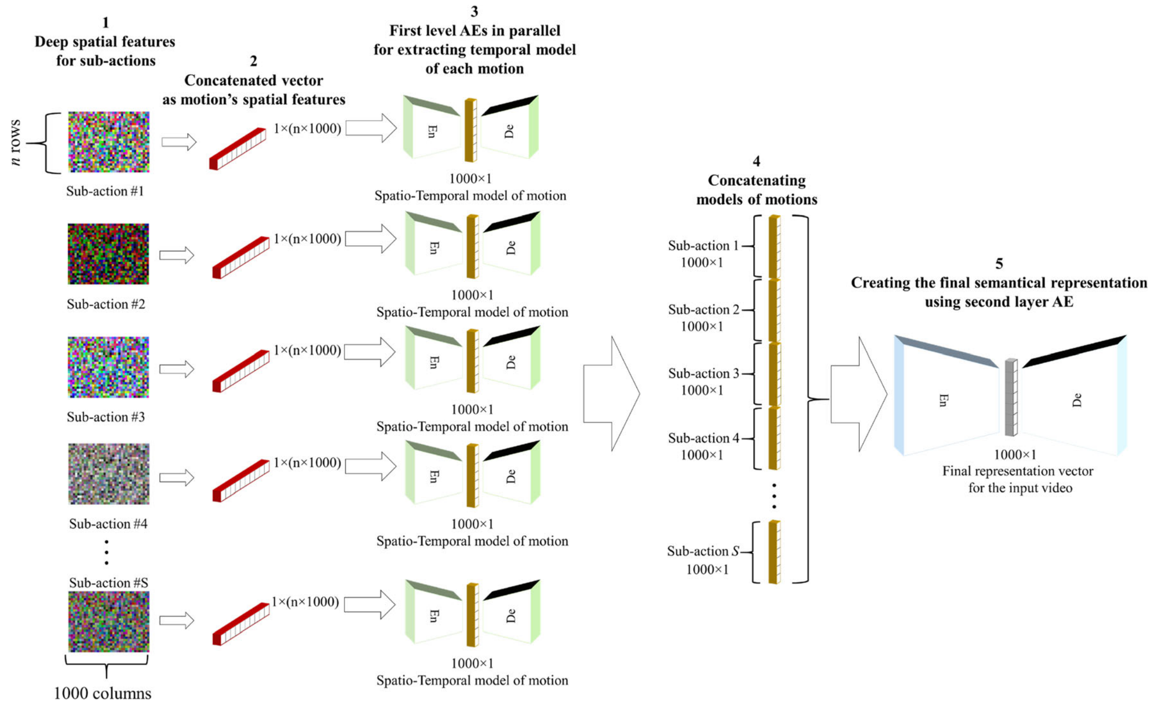


Fig. 3 Structure of creating the final spatiotemporal model for recognition tasks

model is then used by the classifier to perform the recognition task.

As shown in Fig. 3, raw information from each sub-action (i.e., a $n \times 1000$ matrix) is concatenated into a vector to be given to the first-level AE. This vector is then given to the first-level AE, and the latent layer of this AE is used as the spatiotemporal model of the sub-action (i.e., the model of the corresponding motion). The input and output layers of the AE have $n \times 1000$ neurons, and the latent layer of the AEs has 1000 neurons. In our method, temporal features of sub-actions are extracted in parallel form to make feature extraction faster.

Then, the latent layers of AEs in the first level (as spatiotemporal models of the sub-actions) are given to the second-level AE to compute the model of their sequence. The models created by the first-level AEs are sequentially concatenated to create a vector to be given to the second-level AE. The second-level AE has 1000 neurons in the latent layer, and the input and output layers have $s \times 1000$ neurons. This AE seeks to extract the model of the sequence of motions as the final spatiotemporal model of the input action. Considering the sequence of motions leads to a semantical model helping the recognition task perform better than other methods. As another advantage of this method, the sub-actions have representations with fewer variations, and modeling their sequence leads to a more stable action modeling method. It should be noted that if the number of sub-actions for an action category is less than s, zero-padding is used to compensate the size of the input vector of the second-level AE. Finally, the latent layer of the second AE is used for the classification. To perform classification, a simple fully connected neural network is used that contains two hidden layers with the Relu activation function and an output layer with the Softmax activation function. Here, the sparse categorical cross-entropy method is used to compute the predicted value as the basis of detecting the final label.

4 Experimental results

In this paper, a vector is computed for each stream video including the model, the sub-actions, and their sequence as the action representation. In the recognition task, the final representation is given to a classifier to detect the label (i.e., action category) of the input video. The experiments are run on a computer system with Intel Core i7 and 16 GB DDR3 memory working under Microsoft Windows 10 operating system. In these experiments, the values of σ and τ variables are 2.4 and 1.7 as the best values, respectively [1]. In fact, these parameters indicate the size of the Gaussian and the Gabor filters. Furthermore, the value of T empirically is set to 16. After finding sub-actions, 30 keyframes are selected for further calculations (i.e., the value of S is 30).

In this section, two types of experiment are run and their results are reported. The first one relates to evaluating the rate of predicted labels as a recognition task. In the recognition task, 80% of videos in each dataset are used in the training step, and the remaining 20% are considered as test data. Furthermore, fivefold cross-validation is used during the experiments to make results more reliable. Other experiments relate to evaluating segmented sub-actions and their deep spatial features in a retrieval framework. In this experiment type, the extracted deep spatial features of sub-actions are directly used to calculate the similarity between videos and select the ones most similar to the query video as the retrieval task, and the results are compared with the best state-of-the-art retrieval methods. It should be noted that improving the performance of the retrieval task as a challenging task relies on the quality of sub-actions and the computed features, because retrieval methods lack any training step during their procedure. The remainder of this section comprises two subsections related to evaluating the proposed recognition method (i.e., 4.1 subsection) and evaluating the sub-actions and their features in the retrieval task (i.e., 4.2 subsection), respectively.

4.1 Evaluating the performance of the proposed method

Here, a video is given to the trained system, and the system detects a label based on the included action. The proposed method is compared to optical flow + CNN-based features [49], two-stream region-based [51], deep multi-view representation [44], traditional multi-stream [40], dilated CNN + BiLSTM (bidirectional LSTM) [15], DAE (deep AE) with CNN [14], and region sequence-based multi-stream [52] and LRTF (light relation to trajectory features) [46] methods as recent ones. As shown in Fig. 4, our method divides each video into sub-actions. Sub-actions are spatially and temporally modeled, and their sequence is the action model. Table 2 shows the accuracy of these methods on UCF-Sport, UCFYT, and JHMDB datasets; the best values are shown in bold face. It is clear from this table that the proposed method has the best accuracy on all datasets. The considerable performance progress relates to the JHMDB dataset, with 21 action categories as a challenging dataset. Here, the proposed method is compared with different state-of-the-art methods. The second-best method on the UCF-Sport dataset is a multi-stream method that uses traditional features. On the UCFYT dataset, the second-best performance was achieved by deep multi-view representation, which uses the spatial features extracted by a VGG16 network to represent the action. The LRTF method also gave the second-best performance on the UCFYT dataset, because it computes the model in detail by considering spatial dimensions independently and calculating the temporal feature on the spatial model in another step. On the JHMDB dataset, dilated CNN + BiLSTM is the second-best method that uses CNNs with updated weights. Overall, the methods that use deep networks for feature extraction perform better than the methods that use traditional features.

Our method, as a single-stream framework, performs better because of semantical features extracted in the form of a sequence of sub-actions. The proposed method is based on modeling the sequence of sub-actions which are simpler, have representation with less variations, and consider sub-actions regardless of their execution time. Thus, the final action model includes a detailed action structure to be more accurate for categorization. After training a simple classifier, the method learns the sequence of sub-actions to achieve a good and stable performance on all datasets whether they have few or many action categories.

Table 3 shows that the proposed method gives the best performance on almost all action categories of the UCF-Sport dataset. Here, the proposed method is weaker than the traditional multi-stream method on kicking and riding horse action categories. The proposed method uses the information of the sequence of sub-actions (i.e., motions) to create the spatiotemporal representation of the action. Thus, those action categories with similar motions that are frequently repeated and dominant undermine the proposed method during classification. Clearly, the opposite may not be the case. For example, kicking action has a similar foot motion to horse riding, which results in a slight overlap between these action categories (see Fig. 5a). The foot motion in the horse riding action is frequently repeated, and this motion in kicking the ball action is dominant in the video. Table 4 shows that the proposed method has the best performance on all action categories except the trampoline one. Clearly, the trampoline action has a frequently repeated motion that is considerably similar to a major part of diving and volleyball spiking actions. Thus, because of the motions appearing in the final model, the proposed method achieves a little overlap between such categories. It should be noted that as the representation vectors contain the information of the sequence of sub-actions, the neural network classifier learns the differences among various action categories and achieves a proper performance. Overall, the proposed method has the best performance.

Because state-of-the-art methods perform weakly on the JHMDB dataset, only the second-best method in Table 1 was selected for use in comparing accuracy levels per category in Table 5. Table 5 also shows that the proposed method significantly outperforms the dilated CNN + BiLSTM method. It is clear that the proposed method performs more weakly only on run and pick action categories, which similarly relates to these actions existing inside other action categories as dominant sub-actions. Nevertheless, the proposed method has an overall better performance than other methods. The average accuracy

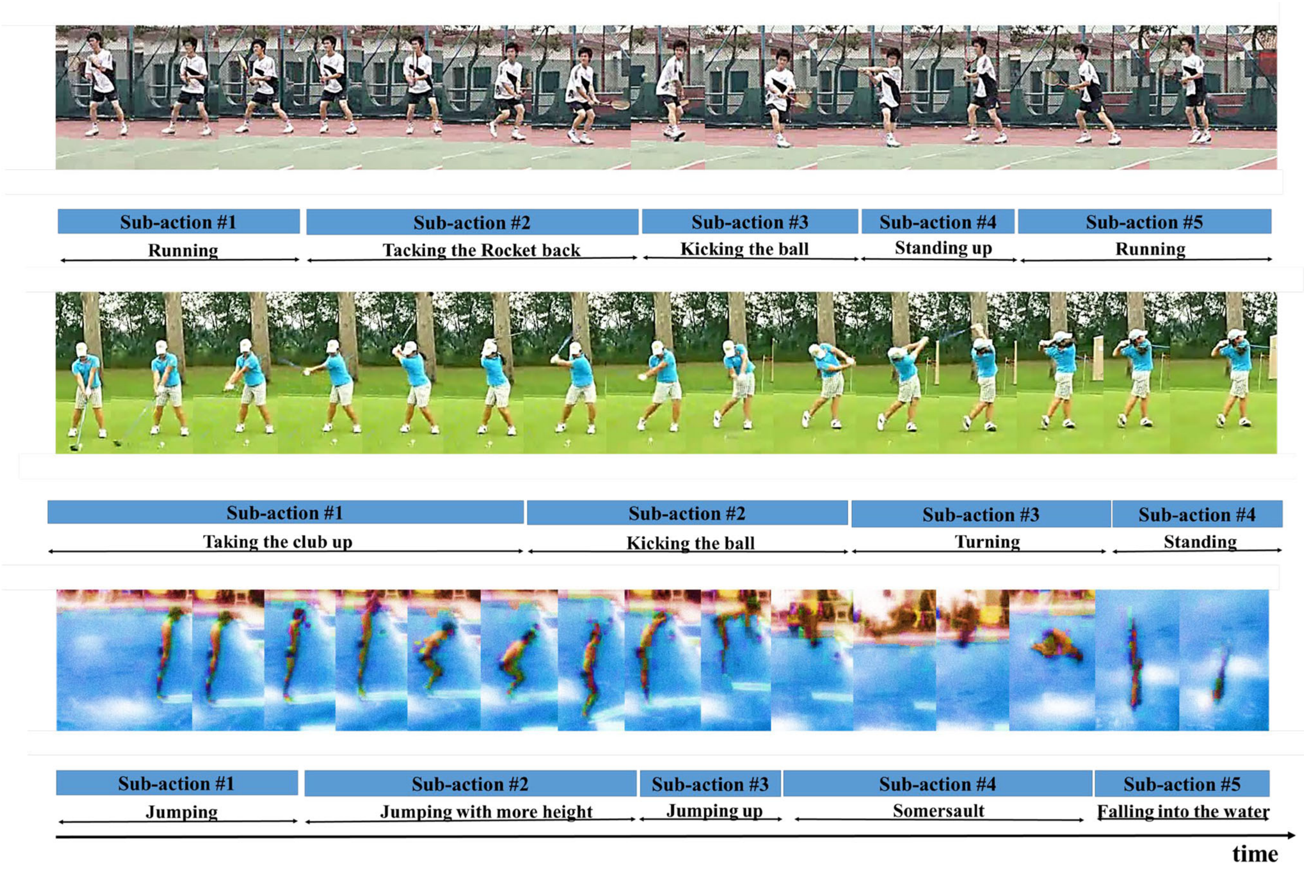


Fig. 4 Segmenting each stream video into sub-actions for tennis, golf, and diving actions

Table 2 Comparison of the performance of different action recognition methods

	Dataset		
	UCF-Sport	UCFYT	HMDB
Optical flow + CNN-based features [49]	0.9753	–	0.763
Deep multi-view representation [44]	0.98	0.994	–
Traditional multi-stream [40]	0.993	0.945	–
Dilated CNN + BiLSTM [15]	0.991	0.983	0.802
LRTF [46]	–	0.993	0.774
DAE + CNN [14]	–	0.9621	–
Two-stream region-based [51]	–	–	0.7117
Region sequence-based multi-stream [52]	–	–	0.769
Proposed method	**0.996**	**0.995**	**0.926**

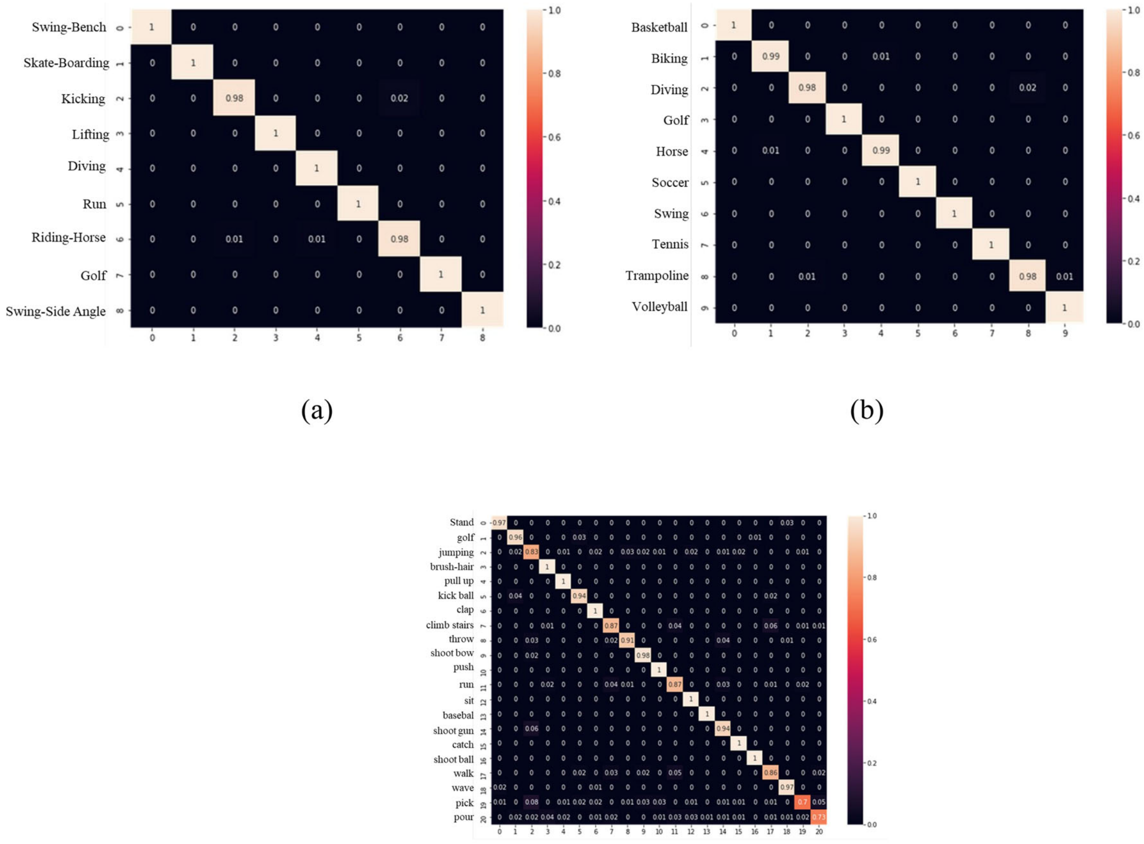
Bold values refer to the best values

rates of the proposed and dilated CNN + BiLSTM methods on run and pick action categories are 0.79 and 0.85, respectively, indicating a difference between them of about 6%. The average accuracy rates of these methods in other categories are 0.94 and 0.77, respectively, a difference between them of about 17%. Thus, the proposed method is much better than the dilated CNN + BiLSTM method.

Similar to the justification of the results in Tables 3, 4, and 5, Fig. 5 shows where the action categories have overlap. Figure 5 presents the confusion matrix of the performance of the proposed method on the three used datasets. In all these confusion matrices, the x and y axes indicate the predicted and the real labels, respectively. For easier comparisons, Tables 6, 7, and 8 represent the statistics of overlap of Fig. 5a, b, and c, respectively.

Table 3 Comparison of the performance of different action recognition methods per each action category of UCF-Sport dataset

	Categories								
	Swing Bench	Skateboarding	Kicking	Lifting	Diving	Run	Horse riding	Golf	Swing-Side Angle
Deep multi-view representation [44]	1	0.96	0.96	1	1	0.94	1	0.99	0.96
Traditional multi-stream [40]	0.99	1	0.99	1	1	0.99	1	0.99	0.99
Dilated CNN + BiLSTM [15]	0.99	0.97	0.98	1	1	0.98	0.99	0.99	0.98
Proposed method	1	1	0.98	1	1	1	0.98	1	1



(a)

(b)

(c)

Fig. 5 Confusion matrix of the proposed method on **a** UCF-Sport, **b** UCFYT, and **c** JHMDB datasets

Table 4 Comparison of the performance of different action recognition methods per each action category of UCFYT dataset

	Categories									
	Basketball	Biking	Diving	Golf	Horse	Soccer	Swing	Tennis	Trampoline	Volleyball
Deep multi-view representation [44]	0.99	0.99	0.98	0.98	0.99	0.98	0.99	0.98	**0.99**	0.98
Traditional multi-stream [40]	0.97	0.93	0.94	0.95	0.98	0.95	0.94	0.98	0.86	0.96
Dilated CNN + BiLSTM [15]	0.99	0.98	0.97	1	0.99	0.99	0.98	0.96	0.98	0.98
Proposed method	**1**	**0.99**	**0.98**	**1**	**0.99**	**1**	**1**	**1**	0.98	**1**

Bold values refer to the best values

Table 5 Comparison of the performance of different action recognition methods per each action category of JHMDB dataset

	Categories																				
	Stand	Golf	Jumping	Brush hair	Pull up	Kick ball	Clap	Climb stairs	Throw	Shoot bow	Push	Run	Sit	Baseball	Shoot gun	Catch	Shoot ball	Walk	Wave	Pick	Pour
Dilated CNN + BiLSTM [15]	0.6	0.64	0.72	0.92	0.73	0.74	0.67	0.78	0.69	0.75	0.72	**0.94**	0.93	0.92	0.7	0.97	0.78	0.72	0.82	**0.77**	0.7
Proposed method	**0.97**	**0.96**	**0.83**	**1**	**1**	**0.94**	**1**	**0.87**	**0.91**	**0.98**	**1**	0.87	**1**	**1**	**0.94**	**1**	**1**	**0.86**	**0.97**	0.7	**0.73**

Bold values refer to the best values

Figure 5a indicates that the proposed method detects about 2% of kicking ball videos as being in the horse riding category. Clearly, the actor who kicks the ball plays a similar foot motion to that of the actor riding the horse. Hence, the proposed method achieves a little overlap between these two action categories. It should be noted that other action categories have such similarities, but the used neural network classifier compensate for such effect by learning the sequence of sub-action models (i.e., the features and their sequences). Thus, the overall performance is better than that of state-of-the-art methods. Figure 5b also shows that about 1% of biking videos may be considered horse riding. Furthermore, about 1% of dive videos are labeled as trampoline. The diving action is composed of jumping, somersault, and falling into water sub-actions, with the jumping sub-action having more execution time than the other two parts. Thus, our semantical method considers a little overlap between these action categories.

Figure 5c relates to the JHMDB dataset which contains more overlaps than Fig. 5a and b, because it includes a large number of action categories. The overlaps are meaningful in this figure, e.g., 4% of the videos in the run category are considered as climb stair action. Clearly, these two actions have similar sub-actions that are frequently repeated. As another example, kick ball videos have 4% overlap with the golf action category, indicating the existence of similar motions (see Fig. 6). Moreover, 5% of walk videos are labeled as run, which indicates the similarity of their spatial features.

In Tables 6, 7, and 8, the information in confusion matrices is analyzed using statistical data. In these tables, the rate of the overlaps between each pair of categories is presented. Here, rows and columns show the actual and detected labels, respectively. Also, precision and recall are used for analyzing the results. To this end, TP_i (true positive) and FN_i (false negative) values for each row indicate the rate of correct labeling and the rate of mistakenly labeled videos of a category, respectively. On the other hand, the FP_i (false positive) value for each column indicates the rate of videos that are mistakenly assigned to an action category. For example, TP for the kicking category is 0.98 and shows the rate of correct labeling for this category in Table 6. The value of FN for this category is 0.02 which indicates that 2% of kicking videos are mistakenly attributed to other categories. The value of FP for this category is 0.01 and indicates that 1% of videos of other categories are mistakenly labeled as kicking. Clearly, the proposed method has proper performance in assigning videos to their own category, and the values of precision and recall confirm this claim.

Figure 6 shows some cases of incorrect labeling. Here, a video including horse riding action is labeled as biking. In fact, the horse riding action is mistakenly considered a

Table 6 Statistics of categories overlaps for UCF-Sport dataset

ID number of category	Categories	Categories									TPi (Rate of videos with correct labeling in each row)	FNi (Sum of underlined values in each row)
		Swing Bench	Skateboarding	Kicking	Lifting	Diving	Run	Horse riding	Golf	Swing-Side Angle		
1	Swing Bench	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1**	0
2	Skateboarding	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1**	0
3	Kicking	0.00	0.00	**0.98**	0.00	0.00	0.00	0.02	0.00	0.00	**0.98**	0.02
4	Lifting	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	**1**	0
5	Diving	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	**1**	0
6	Run	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	**1**	0
7	Horse riding	0.00	0.00	0.01	0.00	0.01	0.00	**0.98**	0.00	0.00	**0.98**	0.02
8	Golf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	**1**	0
9	Swing-Side Angle	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	**1**	0
FPi (Sum of underlined values in each column)		0	0	0.01	0	0.01	0	0.02	0	0		

$$\sum_{i=1}^{9} \mathrm{FP}_i = 0.04 \sum_{i=1}^{9} \mathrm{FN}_i = 0.04 \sum_{i=1}^{9} \mathrm{TP}_i = 8.96 \text{ Precision} = \frac{\sum_{i=1}^{9} TP_i}{\sum_{i=1}^{9} TP_i + \sum_{i=1}^{9} FP_i} = 99.55\text{Recall} = \frac{\sum_{i=1}^{9} TP_i}{\sum_{i=1}^{9} TP_i + \sum_{i=1}^{9} FN_i} = 99.55$$

Bold values refer to the True Positive rate

biking action. Additionally, Fig. 6 shows that diving and golf actions are mistakenly considered jumping and kick ball actions, respectively. Clearly, actions that are labeled interchangeably actually contain very similar movements. Overall, the results show that the initial idea appears in the performance of the proposed method. As a simple classifier is used here, these results indicate the proper model computed for actions based on the sequence of the features of sub-actions.

4.2 Evaluating raw information of sub-actions

In this section, the main goal is to evaluate the extracted deep spatial features of detected sub-actions using a retrieval framework. The retrieval task seeks to find most similar videos to a query one based on a comparison of their features. In this task, as a trained model is lacking, the accuracy values are fewer than those of the recognition task. The results of this task indicate the usefulness of the extracted sub-actions and their features. It is expected that the overlaps between action categories with similar motions will be more than the overlaps in the recognition task because of the lack of the training step and learning the sequence of features. Here, the extracted deep spatial features are used to compare the content of videos and identify those videos most similar to the query one. As shown in Fig. 7, a pooling technique is applied to each matrix corresponding to a sub-action in order to decrease the dimensionality of the features. This pooling technique includes pooling in horizontal and vertical directions instead of only one direction to avoid creating squares of spatial features. In fact, this pooling structure can create a robust vector as a high-level feature to represent the spatial features of a sub-action. The result of each pooling is gathered in a vector. Up to now, for each sub-action, a vector was calculated, and concatenating the vectors based on their time order resulted in the final vector. To perform another dimensionality reduction, 1D pooling is applied to this concatenated vector which is used to compare the contents of the videos. In our experiments, dynamic time wrapping was used to compare each pair of vectors. Here, each video of a dataset is eliminated from the dataset and considered as the query video. Then, the query's vector was compared to the vectors of all videos to find the top 20 videos most similar to the query one. The accuracy of the method indicates the rate of detected videos that are in the same category as the query one.

Here, our method is compared with state-of-the-art methods, namely motion pattern [1], resultant vector [26], 4D resultant [29], Jones [32], STF (spatial–temporal feature) [5], VG (vocabulary-guided) pyramid [56], and ST (spatial–temporal) pyramid [56] methods. Table 9 compares the performance of these methods on the used datasets. Clearly, the proposed method performed best on all

Table 7 Statistics of categories overlaps for UCFYT dataset

ID number of category	Categories	Categories										TPi (Rate of videos with correct labeling in each row)	FNi (Sum of underlined values in each row)
		Basketball	Biking	Diving	Golf	Horse	Soccer	Swing	Tennis	Trampoline	Volleyball		
1	Basketball	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1**	0
2	Biking	0.00	**0.99**	0.00	0.00	$\underline{0.01}$	0.00	0.00	0.00	0.00	0.00	**0.99**	$\underline{0.01}$
3	Diving	0.00	0.00	**0.98**	0.00	0.00	0.00	0.00	0.00	$\underline{0.02}$	0.00	**0.98**	$\underline{0.02}$
4	Golf	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	**1**	0
5	Horse	0.00	$\underline{0.01}$	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	**0.99**	$\underline{0.01}$
6	Soccer	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	**1**	0
7	Swing	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	**1**	0
8	Tennis	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00	**1**	0
9	Trampoline	0.00	0.00	$\underline{0.01}$	0.00	0.00	0.00	0.00	0.00	**0.98**	$\underline{0.01}$	**0.98**	$\underline{0.02}$
10	Volleyball	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	**1**	0
FPi (Sum of underlined values in each column)		0	$\underline{0.01}$	$\underline{0.01}$	0	$\underline{0.01}$	0	0	0	$\underline{0.02}$	$\underline{0.01}$		

$$\sum_{i=1}^{10} \mathrm{FP}_i = 0.06 \sum_{i=1}^{10} \mathrm{FN}_i = 0.06 \sum_{i=1}^{10} \mathrm{TP}_i = 9.94 \text{ Precision} = \frac{\sum_{i=1}^{10} TP_i}{\sum_{i=1}^{10} TP_i + \sum_{i=1}^{10} FP_i} = 99.4 \text{Recall} = \frac{\sum_{i=1}^{10} TP_i}{\sum_{i=1}^{10} TP_i + \sum_{i=1}^{10} FN_i} = 99.4$$

Bold values refer to the True Positive rate

Table 8 Statistics of overlaps of some categories for JHMDB dataset

ID number of category	Categories	Categories											
		Stand	Golf	Jumping	Brush-hair	Pull up	Kick ball	Clap	Climb stairs	Throw	Shoot bow	Push	Run
1	Hand Stand	**0.97**	0	0	0	0	0	0	0	0	0	0	0
2	Golf	0	**0.96**	0	0	0	0.03	0	0	0	0	0	0
3	Jumping	0	0.02	**0.83**	0	0.01	0	0.02	0	0.03	0.02	0.01	0
4	Brush hair	0	0	0	**1**	0	0	0	0	0	0	0	0
5	Pull up	0	0	0	0	**1**	0	0	0	0	0	0	0
6	Kick ball	0	0.04	0	0	0	**0.94**	0	0	0	0	0	0
7	Clap	0	0	0	0	0	0	**1**	0	0	0	0	0
8	Climb stairs	0	0	0	0.01	0	0	0	**0.87**	0	0	0	0.04
9	Throw	0	0	0.03	0	0	0	0	0.02	**0.91**	0	0	0
10	Shoot bow	0	0	0.02	0	0	0	0	0	0	**0.98**	0	0
11	Push	0	0	0	0	0	0	0	0	0	0	**1**	0
12	Run	0	0	0	0.02	0	0	0	0.04	0.01	0	0	**0.87**
13	Sit	0	0	0	0	0	0	0	0	0	0	0	0
14	Baseball	0	0	0	0	0	0	0	0	0	0	0	0
15	Shoot gun	0	0	0.06		0	0	0	0	0	0	0	0
16	Catch	0	0	0	0	0	0	0	0	0	0	0	0
17	Shoot ball	0	0	0	0	0	0	0	0	0	0	0	0
18	Walk	0	0	0	0	0	0.02	0	0.03	0	0.02	0	0.05
19	Wave	0.02	0	0	0	0	0	0.01	0	0	0	0	0
20	Pick	0.01	0	0.08	0	0.01	0.02	0.02	0	0.01	0.03	0.03	0
21	Pour	0	0.02	0.02	0.04	0.02	0	0.01	0.02	0	0	0.01	0.03
FPi (Sum of underlined values in each column)		0.03	0.08	0.21	0.07	0.04	0.07	0.06	0.11	0.05	0.07	0.05	0.12

ID number of category	Categories	Categories										
		Sit	Baseball	Shoot gun	Catch	Shoot ball	Walk	Wave	Pick	Pour	TPi	FNi
1	Hand Stand	0	0	0	0	0	0	0.03	0	0	**0.97**	0.03
2	Golf	0	0	0	0	0.01	0	0	0	0	**0.96**	0.04
3	Jumping	0.02	0	0.01	0.02	0	0	0	0.01	0	**0.83**	0.17
4	Brush hair	0	0	0	0	0	0	0	0	0	**1**	0
5	Pull up	0	0	0	0	0	0	0	0	0	**1**	0
6	Kick ball	0	0	0	0	0	0.02	0	0	0	**0.94**	0.06
7	Clap	0	0	0	0	0	0	0	0	0	**1**	0
8	Climb stairs	0	0	0	0	0	0.06	0	0.01	0.01	**0.87**	0.13
9	Throw	0	0	0.03	0	0	0	0.01	0	0	**0.91**	0.09
10	Shoot bow	0	0	0	0	0	0	0	0	0	**0.98**	0.02
11	Push	0	0	0	0	0	0	0	0	0	**1**	0
12	Run	0	0	0.03	0	0	0.01	0	0.02	0	**0.87**	0.13
13	Sit	**1**	0	0	0	0	0	0	0	0	**1**	0
14	Baseball	0	**1**	0	0	0	0	0	0	0	**1**	0
15	Shoot gun	0	0	**0.94**	0	0	0	0	0	0	**0.94**	0.06
16	Catch	0	0	0	**1**	0	0	0	0	0	**1**	0
17	Shoot ball	0	0	0	0	**1**	0	0	0	0	**1**	0
18	Walk	0	0	0	0	0	**0.86**	0	0	0.02	**0.86**	0.14
19	Wave	0	0	0	0	0	0	**0.97**	0	0	**0.97**	0.03

Table 8 (continued)

ID number of category	Categories	Categories										
		Sit	Baseball	Shoot gun	Catch	Shoot ball	Walk	Wave	Pick	Pour	TPi	FNi
20	Pick	0.01	0	0.01	0.01	0	0.01	0	**0.7**	0.05	**0.7**	0.3
21	Pour	0.03	0.01	0.01	0.01	0	0.01	0.01	0.02	**0.73**	**0.73**	0.27
FPi (Sum of underlined values in each column)		0.06	0.01	0.09	0.04	0.01	0.11	0.05	0.06	0.08		

$$\sum_{i=1}^{21} \mathrm{FP}_i = 1.47 \sum_{i=1}^{21} \mathrm{FN}_i = 1.47 \sum_{i=1}^{21} \mathrm{TP}_i = 19.53 \text{ Precision} = \frac{\sum_{i=1}^{21} TP_i}{\sum_{i=1}^{21} TP_i + \sum_{i=1}^{21} FP_i} = 93 \text{Recall} = \frac{\sum_{i=1}^{21} TP_i}{\sum_{i=1}^{21} TP_i + \sum_{i=1}^{21} FN_i} = 93$$

The underline refers to considerable wrong overlap between categories

Bold values indicate the true positive (TP) rate

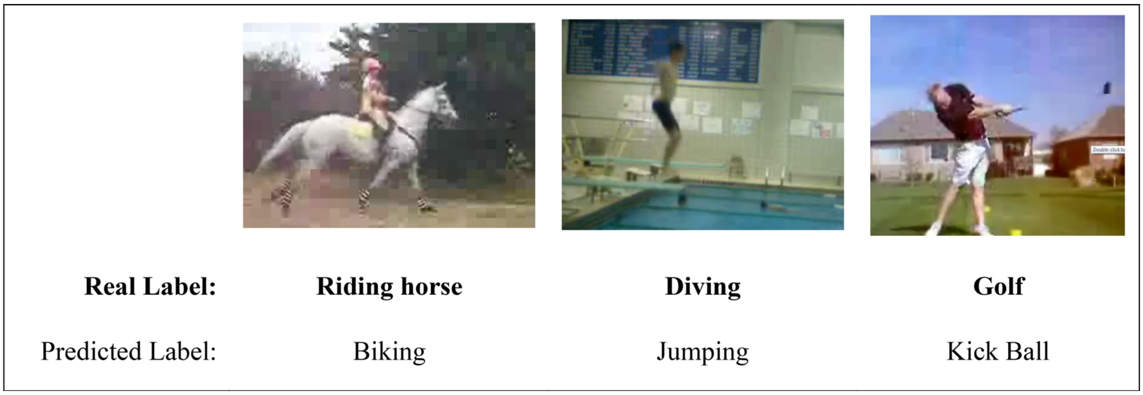


Fig. 6 Samples of incorrect labeling

datasets. The accuracy of the proposed method is about 4%, 1.5%, and 0.7% better than the accuracy of the second-best method (i.e., motion pattern method) on UCF-Sport, UCFYT, and HMDB datasets, respectively. These results indicate that the proposed method can extract proper sub-actions, for which the sequence of their spatial features can accurately represent the actions to be used in different action analysis tasks. The improvement on the HMDB dataset, which is a challenging one, seems to be less than the improvements on other datasets. It refers to several action categories existing in this dataset, most of which have similar sub-actions. In the retrieval task, the lack of a trained model for discriminating the categories resulted in weaker total accuracies than the recognition methods. However, the results indicated that the proposed framework performed better than the other modeling methods. In continuation, Fig. 9 is used to analyze the output results of the proposed method on UCF-Sport, UCFYT, and HMDB datasets.

Figure 8a presents the confusion matrix of the proposed model on the UCF-Sport dataset. Clearly, the most important overlaps achieved by the proposed method relate to golf-vs-kicking, lifting-vs-kicking, running-vs-kicking, skateboarding-vs-golf, and running-vs-swinging pairs in the UCF-Sport dataset. Most of these overlaps relate to the similar motions each pair of actions shares during their execution. For example, running and swinging actions have similar hand poses and motion speed in most videos, so their model would be partially similar. Moreover, some actors in swinging videos execute foot motions that are similar to the foot motions of running actors. Kicking and golf actions also have similar content, because the ball that is hit is mostly on the ground, and similar hand movements are executed in skateboarding and golf actions as well. Furthermore, a considerable part of an executed action such as kicking in some videos relates to other action/s such as running which results in a similar created model for these categories. Overall, the proposed model efficiently represents actions that can be used for human action analysis. Figure 9 indicates that the proposed method has the best performance in comparison to other methods.

The confusion matrix of the proposed method for the UCFYT dataset is shown in Fig. 8b. As can be seen, the main overlap relates to the trampoline golf categories. As another overlap, about 0.16% of the horse riding videos are retrieved for queries from the biking category.

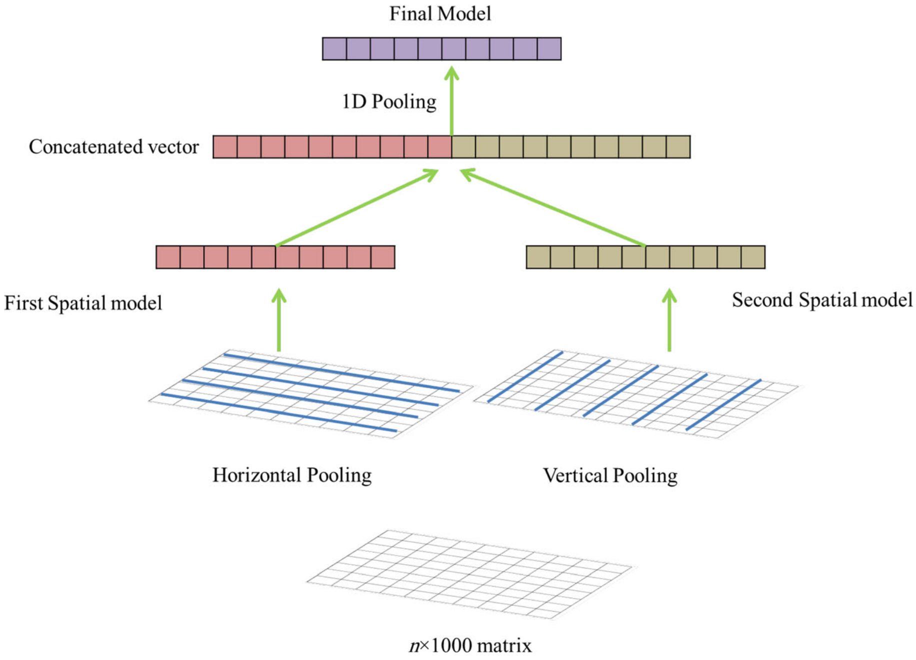


Fig. 7 Structure of the used pooling technique

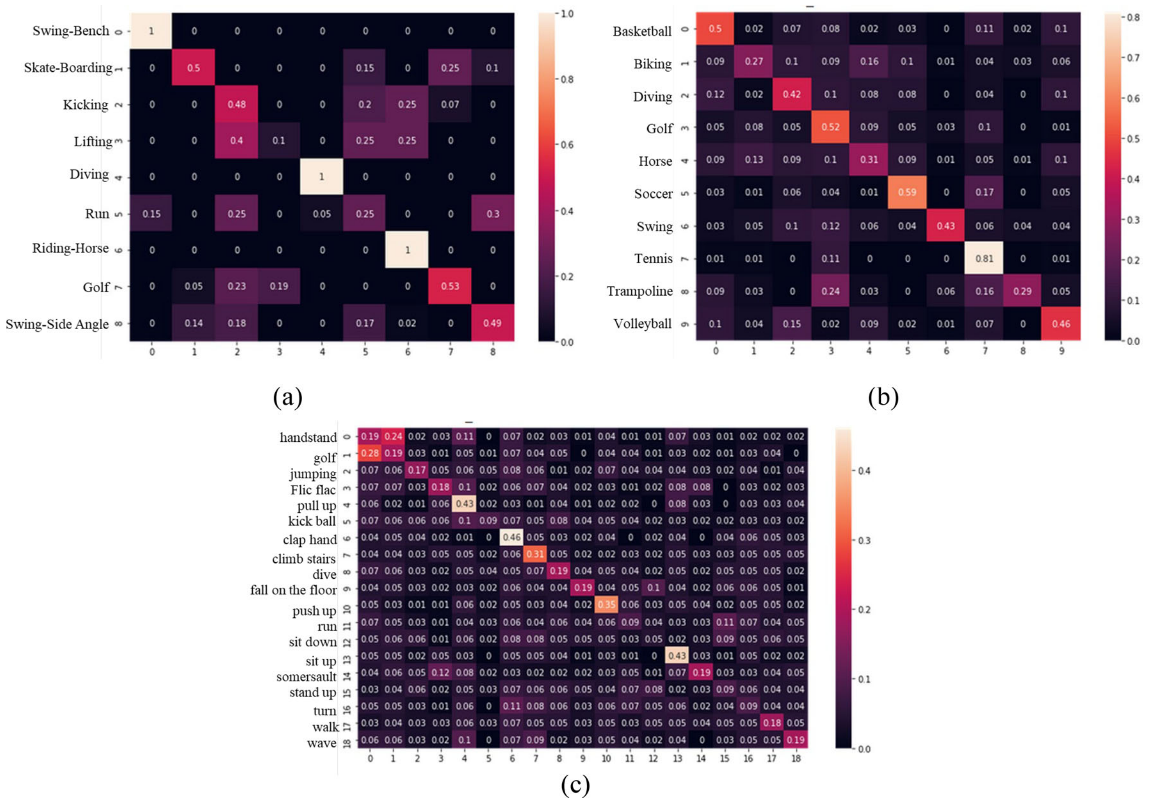
Table 9 Comparison of the accuracy of different retrieval methods

	Dataset		
	UCF-Sport	UCFYT	HMDB
Motion pattern [1]	0.55	0.45	0.21
Resultant vector [26]	0.545	0.356	0.127
4-D resultant [29]	0.546	0.4	0.143
Jones [32]	0.525	0.233	0.101
STF [5]	0.521	0.376	0.144
VG pyramid [56]	0.5	0.312	–
ST pyramid [56]	0.53	0.36	–
Proposed method	**0.59**	**0.465**	**0.217**

Bold values refer to the best values

Furthermore, about 0.17% of tennis videos are retrieved for queries from the soccer action. These overlaps relate to the similarity of motions that are executed during the actions, and matching the models more clearly highlights these similarities. Thus, it can be concluded that the proposed model can represent the sub-actions efficiently, and categories other than trampoline have no considerable overlap with other actions. Compared with other state-of-the-art methods, the proposed one has significantly better accuracy on four action categories and comparable results on other categories (see Fig. 10). The accuracy rate of the proposed method in these four categories (i.e., golf, soccer, swing, and tennis) is 0.52, 0.59, 0.44, and 0.81, respectively. These values for the motion pattern method, which is the second-best method, are 0.31, 0.38, 0.42, and 0.43, respectively. Thus, our method performs about 20% better on these categories than the motion pattern method. The principal overlap on the UCFYT that occurs with the proposed method relates to the biking, horse riding, and trampoline categories. It is clear that about 16% of the horse riding videos are retrieved for the queries in the biking category. In fact, these two categories have similar body poses and consequently, similar sub-actions executed by the actor on the horse and bicycle.

Another dataset used for our evaluations was HMDB, a challenging one. The existing scattering in the confusion matrix shown in Fig. 8c proves this claim. This dataset includes a considerable number of categories and videos. With more categories, the number of similar action categories will be increased. Table 9 indicates that the proposed method performed best on this challenging dataset, as analyzed in detail here. The major overlap between two categories refers to handstanding and golf actions. For example, about 24% of the golf videos are retrieved for queries of the handstand category, and about 28% of the handstand videos are retrieved for queries of the golf category. The main reason refers to the similar hand motions that are executed by the actors in these videos. Nevertheless, the proposed method has an overall better performance on these categories than the other methods (see Fig. 11). The proposed method's average accuracy on these categories is about 0.19, while this value for the motion pattern, resultant, and 4-D resultant methods is 0.16, 0.17, and 0.175, respectively. Thus, the proposed method has the best performance on these categories. As shown in Fig. 11, the proposed method performs significantly better on the pull up, clap hand, climb stairs, fall on the floor, push up,



(a)

(b)

(c)

Fig. 8 Confusion matrix of the proposed method's performance on **a** UCF-Sport, **b** UCFYT, and **c** HMDB datasets

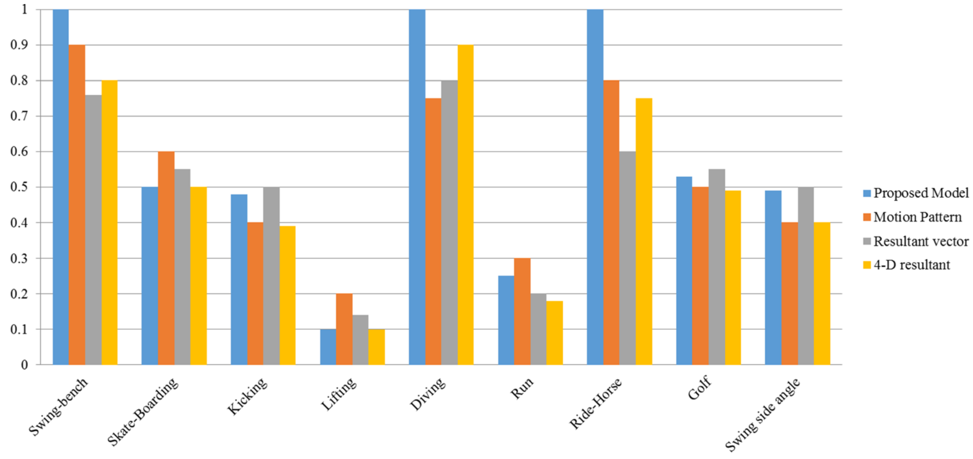


Fig. 9 Comparison of the performance of the proposed model to the state-of-the-art models on UCF-Sport dataset

sit up, turn, and walk categories. The average accuracy of the proposed method on these categories is 0.3056, and this value for the motion pattern, resultant, and 4-D resultant methods is 0.2137, 0.1378, and 1468, respectively.

For easy consideration and comparison, Tables 10, 11, and 12 numerically show the overlaps between the action categories of UCF-Sport, UCFYT, and HMDB datasets in the retrieval task, respectively. These tables can be considered instead of Fig. 8. In the retrieval task, a video is given to the system (i.e., query video) and the method must find 20 similar videos to the query video from the viewpoint of included action in them. In these tables, the "correct recommendation rate" is used to show the rate of the videos that are retrieved correctly. For example, this value for the "skateboarding" category in the UCF-Sport dataset equals 0.5. This value indicates that for all videos which include "skateboarding" action and are given to the system as a query, about 50% of retrieved videos relate to the skateboarding category. It is clear from Table 10 that the other 50% of videos that are found by the system contain run action(about 15%), golf action (about 25%), and swing-side angle action (about 10%).

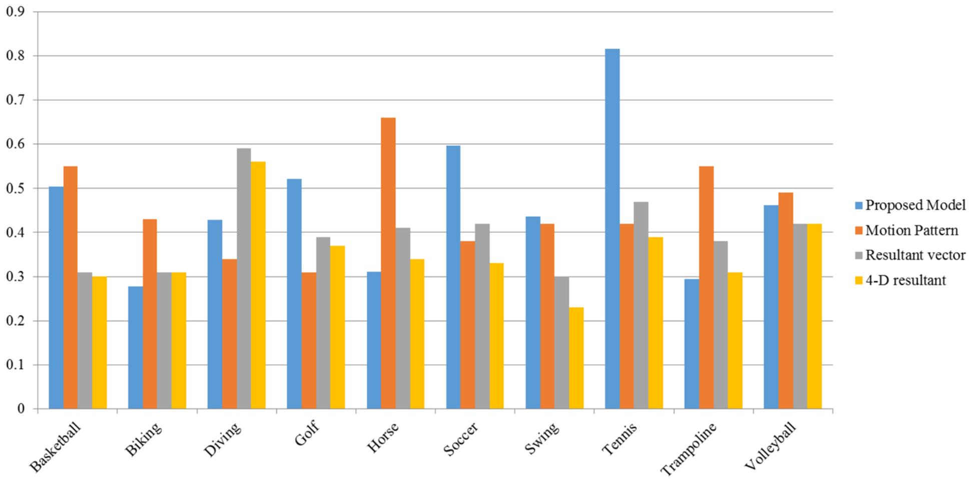


Fig. 10 Comparison of the performance of the proposed model with that of the state-of-the-art models on the UCFYT dataset

On the other hand, the "standard deviation of recommendation" is also calculated for each action category (i.e., each row) in all these tables. This criterion can show whether the distribution of mistakenly recommended videos is between all categories or only a few categories. Clearly, the larger values indicate the method performs better. In fact, it shows that the method can retrieve videos accurately from few categories whose actions include similar motions, and it avoids retrieving videos from all categories. The average value of "standard deviation of recommendation" for UCF-Sport, UCFYT, and HMDB datasets equals 0.2178, 0.1372, and 0.0499, respectively. In fact, the method has recommended videos from almost all categories for each video of the HMDB dataset. Of course, this performance is due to the numerous categories of this dataset. Moreover, the action categories in this dataset have similar motions. Anyway, this method has better performance than other action retrieval methods because of segmenting videos on the time axis, creating sub-actions, and modeling each video based on the sequence of sub-actions.

Overall, the proposed framework seeks to represent actions based on deep spatial and temporal information of executed sub-actions and their sequence. In fact, actions are modeled based on the sequence of all constituting sub-actions, regardless of their execution time, as an accurate semantical model. In total, these experiments indicated that considering deep spatial features of sub-actions and their sequence outperforms all state-of-the-art retrieval and recognition methods. This method also encourages researchers to define semantic in future works.

The major limitation of this method is that it may confuse the videos of two action categories that include similar human body movements. As an outstanding example, horse riding and biking videos were interchangeably labeled several times. Moreover, a sub-action of some videos may be executed faster, allowing other sub-actions to dominate in the final model, which results in incorrect labeling. For example, diving videos, which have three important sub-

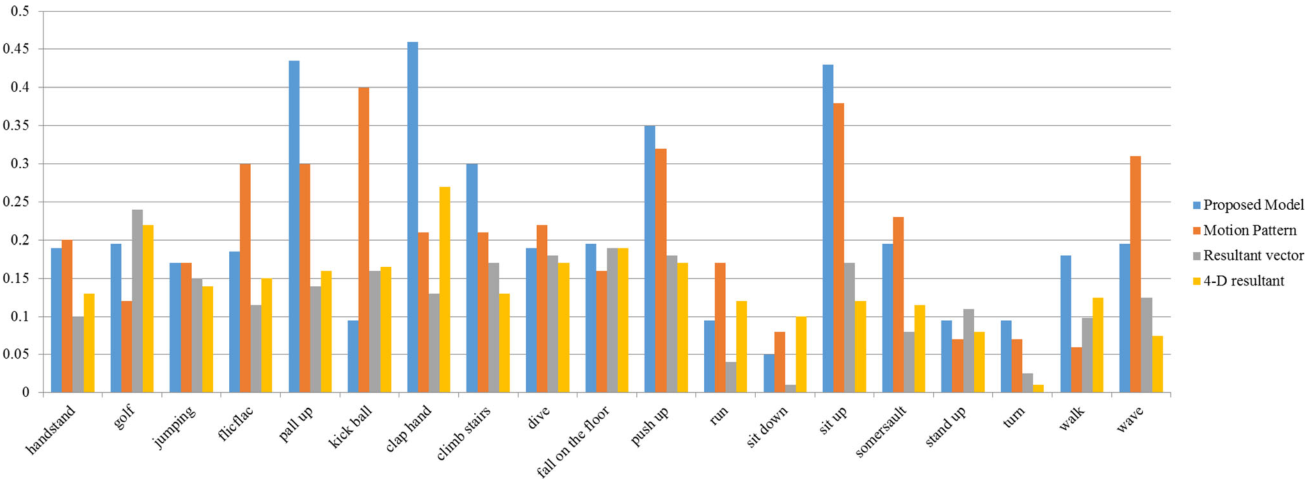


Fig. 11 Comparison of the performance of the proposed model with state-of-the-art models on the HMDB dataset

Table 10 Statistics of categories overlaps for UCF-Sport dataset- Retrieval task

ID number of category	Categories	Categories									Correct recommendation rate	Standard deviation of recommendation	The most similar category
		Swing Bench	Skateboarding	Kicking	Lifting	Diving	Run	Horse riding	Golf	Swing-Side Angle			
1	Swing Bench	**1**	0	0	0	0	0	0	0	0	**1**	*0.3333*	–
2	Skateboarding	0	**0.5**	0	0	0	0.15	0	0.25	0.1	**0.5**	*0.1709*	Golf
3	Kicking	0	0	**0.48**	0	0	0.2	0.25	0.07	0	**0.48**	*0.1683*	Horse riding
4	Lifting	0	0	0.4	**0.1**	0	0.25	0.25	0	0	**0.1**	*0.1516*	Horse riding
5	Diving	0	0	0	0	**1**	0	0	0	0	**1**	*0.3333*	–
6	Run	0.15	0	0.25	0	0.05	**0.25**	0	0	0.3	**0.25**	*0.1269*	Swing-Side
7	Horse riding	0	0	0	0	0	0	**1**	0	0	**1**	*0.3333*	–
8	Golf	0	0.05	0.23	0.19	0	0	0	**0.53**	0	**0.53**	*0.1807*	Kicking
9	Swing-Side Angle	0	0.14	0.18	0	0	0.17	0.02	0	**0.49**	**0.49**	*0.1621*	Kicking
Average											0.5944	0.2178	

Italics values refer to the standard deviation value

Bold values indicate the correct recommendation rate

Table 11 Statistics of categories overlaps for UCFYT dataset—retrieval task

ID number of category	Categories	Categories										Correct recommendation rate	Standard deviation of recommendation	The most similar category
		Basketball	Biking	Diving	Golf	Horse	Soccer	Swing	Tennis	Trampoline	Volleyball			
1	Basketball	**0.5**	0.02	0.07	0.08	0.02	0.03	0	0.11	0.02	0.1	**0.5**	*0.1472*	Tennis
2	Biking	0.09	**0.27**	0.1	0.09	0.16	0.1	0.01	0.04	0.03	0.06	**0.27**	*0.075*	Horse riding
3	Diving	0.12	0.02	**0.42**	0.1	0.08	0.08	0	0.04	0	0.1	**0.42**	*0.1217*	Basketball
4	Golf	0.05	0.08	0.05	**0.52**	0.09	0.05	0.03	0.1	0	0.01	**0.52**	*0.1517*	Tennis
5	Horse	0.09	0.13	0.09	0.1	**0.31**	0.09	0.01	0.05	0.01	0.1	**0.31**	*0.0843*	Biking
6	Soccer	0.03	0.01	0.06	0.04	0.01	**0.59**	0	0.17	0	0.05	**0.59**	*0.1806*	Tennis
7	Swing	0.03	0.05	0.1	0.12	0.06	0.04	**0.43**	0.06	0.04	0.04	**0.43**	*0.1204*	Golf
8	Tennis	0.01	0.01	0	0.11	0	0	0	**0.81**	0	0.01	**0.81**	*0.2534*	Golf
9	Trampoline	0.09	0.03	0	0.24	0.03	0	0.06	0.16	**0.29**	0.05	**0.29**	*0.1016*	Golf
10	Volleyball	0.1	0.04	0.15	0.02	0.09	0.02	0.01	0.07	0	**0.46**	**0.46**	*0.1363*	Diving
Average												**0.46**	0.1372	

Italics values refer to the standard deviation value

Bold values indicate the correct recommendation rate

Table 12 Statistics of overlaps for HMDB dataset—retrieval task

Categories	Categories										
	Handstand	Golf	Jumping	Flic flac	Pull up	Kick ball	Clap hand	Climb stairs	Dive	Fall on the floor	Push up
Handstand	**0.19**	0.24	0.02	0.03	0.11	0	0.07	0.02	0.03	0.01	0.04
Golf	0.28	**0.19**	0.03	0.01	0.05	0.01	0.07	0.04	0.05	0	0.04
Jumping	0.07	0.06	**0.17**	0.05	0.06	0.05	0.08	0.06	0.01	0.02	0.07
Flic flac	0.07	0.07	0.03	**0.18**	0.1	0.02	0.06	0.07	0.04	0.02	0.03
Pull up	0.06	0.02	0.01	0.06	**0.43**	0.02	0.03	0.01	0.04	0.01	0.02
Kick ball	0.07	0.06	0.06	0.06	0.1	**0.09**	0.07	0.05	0.08	0.04	0.05
Clap hand	0.04	0.05	0.04	0.02	0.01	0	**0.46**	0.05	0.03	0.02	0.04
Climb stairs	0.04	0.04	0.03	0.05	0.05	0.02	0.06	**0.31**	0.05	0.02	0.02
Dive	0.07	0.06	0.03	0.02	0.05	0.04	0.05	0.07	**0.19**	0.04	0.05
Fall on the floor	0.04	0.05	0.03	0.02	0.03	0.02	0.06	0.04	0.04	**0.19**	0.04
Push up	0.05	0.03	0.01	0.01	0.06	0.02	0.05	0.03	0.04	0.02	**0.35**
Run	0.07	0.05	0.03	0.01	0.04	0.03	0.06	0.04	0.06	0.04	0.06
Sit down	0.05	0.06	0.06	0.01	0.06	0.02	0.08	0.08	0.05	0.05	0.05
Sit up	0.05	0.05	0.02	0.05	0.03	0	0.05	0.05	0.04	0.01	0.03
Somersault	0.04	0.06	0.05	0.12	0.08	0.02	0.03	0.02	0.02	0.02	0.03
Stand up	0.03	0.04	0.06	0.02	0.05	0.03	0.07	0.06	0.06	0.05	0.04
Turn	0.05	0.05	0.03	0.01	0.06	0	0.11	0.08	0.06	0.03	0.06
Walk	0.03	0.04	0.03	0.03	0.06	0.03	0.07	0.05	0.05	0.03	0.05
Wave	0.06	0.06	0.03	0.02	0.1	0	0.07	0.09	0.02	0.03	0.05
Average											

Categories	Categories										
	Run	Sit down	Sit up	Somersault	Stand up	Turn	Walk	Wave	Correct recommendation rate	Standard deviation of recommendation	The most similar category
Handstand	0.01	0.01	0.07	0.03	0.01	0.02	0.02	0.02	**0.19**	*0.0644*	Golf
Golf	0.04	0.01	0.03	0.02	0.01	0.03	0.04	0	**0.19**	*0.0694*	Handstand
Jumping	0.04	0.04	0.04	0.03	0.02	0.04	0.01	0.04	**0.17**	*0.0351*	Claphand
Flic flac	0.01	0.02	0.08	0.08	0	0.03	0.02	0.03	**0.18**	*0.042*	Pullup
Pull up	0.02	0	0.08	0.03	0	0.03	0.03	0.04	**0.43**	*0.0944*	Sit up
Kick ball	0.04	0.02	0.03	0.02	0.02	0.03	0.03	0.02	**0.09**	*0.0248*	Dive
Clap hand	0	0.02	0.04	0	0.04	0.06	0.05	0.03	**0.46**	*0.1003*	Turn
Climb stairs	0.03	0.02	0.05	0.03	0.03	0.05	0.05	0.05	**0.31**	*0.0636*	Claphand
Dive	0.04	0.02	0.05	0.02	0.05	0.05	0.05	0.02	**0.19**	*0.0371*	Handstand
Fall on the floor	0.05	0.1	0.04	0.02	0.06	0.06	0.05	0.01	**0.19**	*0.0394*	Claphand
Push up	0.06	0.03	0.05	0.04	0.02	0.05	0.05	0.02	**0.35**	*0.0739*	Pullup
Run	**0.09**	0.04	0.03	0.03	0.11	0.07	0.04	0.05	**0.09**	*0.0235*	Standup
Sit down	0.03	**0.05**	0.02	0.03	0.09	0.05	0.06	0.05	**0.05**	*0.021*	Somersault
Sit up	0.01	0	**0.43**	0.03	0.01	0.05	0.02	0.02	**0.43**	*0.0937*	Climbstair
Somersault	0.05	0.01	0.07	**0.19**	0.03	0.03	0.04	0.05	**0.19**	*0.0427*	Flicflac
Stand up	0.07	0.08	0.02	0.03	**0.09**	0.06	0.04	0.04	**0.09**	*0.0198*	Sitdown
Turn	0.07	0.05	0.06	0.02	0.04	**0.09**	0.04	0.04	**0.09**	*0.0268*	Claphand
Walk	0.03	0.05	0.05	0.04	0.05	0.05	**0.18**	0.05	**0.18**	*0.0333*	Claphand
Wave	0.04	0.02	0.04	0	0.03	0.05	0.05	**0.19**	**0.19**	*0.043*	Climbstairs
Average									0.21	0.0499	

Italics values refer to the standard deviation value

Bold values indicate the correct recommendation rate actions (jumping, somersault, and falling into the water), would have a similar model to the jumping video model, if the somersault sub-action were executed faster in the video. To overcome these challenges, a CNN-based classifier is utilized in the recognition task, which leads to the proper performance compared to state-of-the-art and recent methods, but it cannot learn the chain of sub-actions.

5 Conclusion

In this paper, an action modeling method for the recognition task is introduced. This method seeks to model the sequence of deep features of motions of actions as the representation. Action is, in fact, considered a unique sequence of sub-actions, and sub-actions are simpler than actions. Thus, extracting sub-action features and modeling their sequence for each action results in a stable method. After computing the action model captured in a video stream, a simple learning algorithm is used to predict the action label of the video stream. In this method, all sub-actions ignoring their execution time have their own effect on the action model as an important advantage. The video stream is segmented based on an energy diagram computed using the value of response function for video points. Then, spatial and temporal features of sub-actions are extracted by a deep network, and the sequence of sub-action features is used to create the final action model (i.e., representation). Our method performs about 4.26% better than the second-best methods on UCF-Sport, UCFYT, and JHMDB datasets. In addition, detected sub-actions and their spatial features are evaluated by a retrieval framework as a hard task. Results of this experiment demonstrate the quality of segmenting action to sub-actions and extracting features of actions.

But, the main challenge of the proposed method is the potential confusion between videos of two action categories that include similar human body movements. For example, horse riding and biking videos have been interchangeably labeled several times due to the similarity in the body movements involved. This indicates that the method may struggle to accurately distinguish between these similar actions based on deep spatial and temporal information alone. Another challenge relates to the speed of executing sub-actions in a video that can affect the final model and lead to incorrect labeling. For instance, in diving videos where multiple important sub-actions (jumping, somersault, falling into the water) are involved, if the somersault sub-action is executed faster than the others, jumping sub-action may dominate the final model and result in mislabeling the video as a jumping action. While CNN-based model creator and classifier help improve performance compared to state-of-the-art methods, it cannot explicitly learn the sequence of sub-actions, which limits the method's ability to accurately represent and differentiate actions based on their temporal dynamics. Thus, as a future work, we are considering designing a Bayesian network for modeling the sequence of sub-actions efficiently. In other words, we seek to assign a virtual label to each sub-action, and action is modeled based on the sequence of labels. Furthermore, we aim to analyze sub-actions to eliminate repeated motions from our calculations so as to reduce the volume of computations.

Furthermore, this framework is feasible for both recognition and retrieval tasks in real-time applications. In the retrieval task, the framework performs modeling of the action within a video by applying two pooling procedures: horizontal–vertical pooling and 1D pooling. These pooling procedures are executed after segmenting the video and detecting keyframes. In the recognition task, the action model is created using two consecutive AEs. The complexity of the method is directly influenced by the combined complexities of its constituent AEs. This modular design ensures that the overall complexity of the model is comparable to other deep learning models. Therefore, deploying this model in real-time applications is equally feasible. It is important to note that advanced hardware accelerators like GPUs and TPUs can significantly enhance the feasibility of real-time deep learning applications by accelerating the model's computations. Therefore, due to the appropriate volume of AE computations, the simplicity of action modeling using pooling techniques, and the utilization of model optimization, hardware acceleration, and model parallelism (distributing the computational workload across multiple devices or machines), this method is suitable for real-time applications.

Data availability They also affirm that the submitted material is theirs and is original. Furthermore, no dataset was generated during the current study; the datasets used herein, i.e., UCFYT, UCF-Sport, and HMDB, are available on https://www.crcv.ucf.edu/data and https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database.

Declarations

Conflict of interest The current authors declare they have no conflicts of interest and received no funding in relation to this article.

References

1. Ramezani M, Yaghmaee F (2018) Motion pattern based representation for improving human action retrieval. Multimed Tools Appl 77(19):26009–26032

2. Veinidis C, Pratikakis I, Theoharis T (2019) Unsupervised human action retrieval using salient points in 3D mesh sequences. Multimed Tools Appl 78(3):2789–2814
3. Qin J, Liu L, Yu M, Wang Y, Shao L (2017) Fast action retrieval from videos via feature disaggregation. Comput Vision Image Underst 156:104–116
4. Ding S, Li G, Li Y, Li X, Zhai Q, Champion AC, Zhu J, Xuan D, Zheng YF (2017) Survsurf: human retrieval on large surveillance video data. Multimed Tools Appl 76(5):6521–6549
5. Zhang L, Wang Z, Yao T, Mei T, Feng DD (2018) Exploiting spatial-temporal context for trajectory based action video retrieval. Multimed Tools Appl 77(2):2057–2081
6. Ciptadi A, Goodwin MS, Rehg JM (2014) Movement pattern histogram for action recognition and retrieval. In: European conference on computer vision. Springer, Cham, p 695-710
7. Ramezani M, Yaghmaee F (2016) A review on human action analysis in videos for retrieval applications. Artif Intell Rev 46(4):485–514
8. Zhao S, Chen L, Yao H, Zhang Y, Sun X (2015) Strategy for dynamic 3D depth data matching towards robust action retrieval. Neurocomputing 151:533–543
9. Ramezani M, Yaghmaee F (2014) Content-based human actions retrieval by a novel low complex action representation. In: 2014 4th International conference on computer and knowledge engineering (ICCKE). IEEE, p 204–208
10. Jiang X, Zhong F, Peng Q, Qin X (2016) Action recognition based on global optimal similarity measuring. Multimed Tools Appl 75(18):11019–11036
11. Liu X, Li Y (2014) Research on human action recognition based on global and local mixed features. In: 2014 International conference on mechatronics, control and electronic engineering (MCE-14), Atlantis Press p 778–782
12. Jones S, Shao L, Du K (2014) Active learning for human action retrieval using query pool selection. Neurocomputing 124:89–96
13. Ji S, Xu W, Yang M, Yu K (2012) 3D convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
14. Ullah A, Muhammad K, Haq IU, Baik SW (2019) Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments. Futur Gener Comput Syst 96:386–397
15. Muhammad K, Ullah A, Imran AS, Sajjad M, Kiran MS, Sannino G, de Albuquerque VH (2021) Human action recognition using attention based LSTM network with dilated CNN features. Futur Gener Comput Syst 125:820–830
16. Singh R, Khurana R, Kushwaha AK, Srivastava R (2021) A dual stream model for activity recognition: exploiting residual-cnn with transfer learning. Comput Methods Biomech Biomed Eng: Imaging Vis 9(1):28–38
17. Dollár P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: 2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance. IEEE, p 65–72
18. Junejo IN, Dexter E, Laptev I, Perez P (2010) View-independent action recognition from temporal self-similarities. IEEE Trans Pattern Anal Mach Intell 33(1):172–185
19. Shao L, Zhen X, Tao D, Li X (2013) Spatio-temporal Laplacian pyramid coding for action recognition. IEEE Trans Cybern 44(6):817–827
20. Veinidis C, Pratikakis I, Theoharis T (2014) Querying 3D mesh sequences for human action retrieval. In: 2014 2nd International conference on 3D vision. vol 2. IEEE, p 33–40
21. Lin Z, Jiang Z, Davis LS (2009) Recognizing actions by shape-motion prototype trees. In: 2009 IEEE 12th international conference on computer vision. IEEE, p 444–451
22. Zhu F, Shao L, Lin M (2013) Multi-view action recognition using local similarity random forests and sensor fusion. Pattern Recognit Lett 34(1):20–24
23. Shao L, Wu D, Chen X (2011) Action recognition using correlogram of body poses and spectral regression. In: 2011 18th IEEE international conference on image processing. IEEE, p 209–212
24. Shao L, Chen X (2010) Histogram of body poses and spectral regression discriminant analysis for human action categorization. In: BMVC, p 1–11
25. Shao L, Liu L, Yu M (2016) Kernelized multiview projection for robust action recognition. Int J Comput Vis 118(2):115–129
26. Ramezani M, Yaghmaee F (2018) Retrieving human action by fusing the motion information of interest points. Int J Artif Intell Tools 27(03):1850008
27. Sharif M, Khan MA, Zahid F, Shah JH, Akram T (2020) Human action recognition: a framework of statistical weighted segmentation and rank correlation-based selection. Pattern Anal Appl 23(1):281–294
28. Sahoo SP, Ari S (2019) On an algorithm for human action recognition. Expert Syst Appl 115:524–534
29. Ramezani M, Yaghmaee F (2016) A novel video recommendation system based on efficient retrieval of human actions. Phys A: Stat Mech Appl 457:607–623
30. Chen S, Sun Z, Zhang Y, Li Q (2016) Relevance feedback for human motion retrieval using a boosting approach. Multimed Tools Appl 75(2):787–817
31. Shao L, Jones S, Li X (2013) Efficient search and localization of human actions in video databases. IEEE Trans Circuits Syst Video Technol 24(3):504–512
32. Jones S, Shao L (2011) Action retrieval with relevance feedback on YouTube videos. In: Proceedings of the third international conference on internet multimedia computing and service, p 42–45
33. Jiang YG, Li Z, Chang SF (2011) Modeling scene and object contexts for human action retrieval with few examples. IEEE Trans Circuits Syst Video Technol 21(5):674–681
34. Jones S, Shao L (2014) Unsupervised spectral dual assignment clustering of human actions in context. In: Proceedings of the IEEE conference on computer vision and pattern recognition, p 604–611
35. Jones S, Shao L (2013) Content-based retrieval of human actions from realistic video databases. Inf Sci 236:56–65
36. Zhen X, Shao L, Tao D, Li X (2013) Embedding motion and structure features for action recognition. IEEE Trans Circuits Syst Video Technol 23(7):1182–1190
37. Ji R, Yao H, Sun X (2011) Actor-independent action search using spatiotemporal vocabulary with appearance hashing. Pattern Recognit 44(3):624–638
38. Yu G, Yuan J, Liu Z (2015) Unsupervised trees for human action search. In: Human action analysis with randomized trees. Springer, Singapore, p 29–56
39. Páez F, Vanegas JA, González FA (2014) Online multimodal matrix factorization for human action video indexing. In: 2014 12th international workshop on content-based multimedia indexing (CBMI). IEEE, p 1–6
40. Afza F, Khan MA, Sharif M, Kadry S, Manogaran G, Saba T, Ashraf I, Damaševičius R (2021) A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection. Image Vis Comput 106:104090
41. Fan H, Luo C, Zeng C, Ferianc M, Que Z, Liu S, Niu X, Luk W (2019) F-E3D: FPGA-based acceleration of an efficient 3D convolutional neural network for human action recognition. In: 2019 IEEE 30th international conference on application-specific systems, architectures and processors (ASAP) vol 2160. IEEE, p1–8

42. Naeem HB, Murtaza F, Yousaf MH, Velastin SA (2021) T-VLAD: Temporal vector of locally aggregated descriptor for multiview human action recognition. Pattern Recogn Lett 148:22–28
43. Farrajota M, Rodrigues JM, du Buf JH (2019) Human action recognition in videos with articulated pose information by deep networks. Pattern Anal Appl 22(4):1307–1318
44. Khan MA, Javed K, Khan SA, Saba T, Habib U, Khan JA, Abbasi AA (2020) Human action recognition using fusion of multiview and deep features: an application to video surveillance. Multimed Tools Appl 14:1–27
45. Wang J, Shao Z, Huang X, Lu T, Zhang R, Lv X (2021) Spatial–temporal pooling for action recognition in videos. Neurocomputing 451:265–278
46. Javidani A, Mahmoudi-Aznaveh A (2022) Learning representative temporal features for action recognition. Multimed Tools Appl 81(3):3145–3163
47. Pirri F, Mauro L, Alati E, Ntouskos V, Izadpanahkakhk M, Omrani E. ime tn.
48. Saifuddin Saif AFM, Wollega ED, Kalevela SA (2023) Spatio-temporal features based human action recognition using convolutional long short-term deep neural network. Int J Adv Comput Sci Appl. https://doi.org/10.14569/IJACSA.2023.0140501
49. Zong M, Wang R, Chen X, Chen Z, Gong Y (2021) Motion saliency based multi-stream multiplier ResNets for action recognition. Image Vis Comput 107:104108
50. Abdelbaky A, Aly S (2020) Two-stream spatiotemporal feature fusion for human action recognition. Vis Comput 9:1–5
51. Dai C, Liu X, Lai J (2020) Human action recognition using two-stream attention based LSTM networks. Appl Soft Comput 86:105820
52. Zhao Y, Man KL, Smith J, Siddique K, Guan SU (2020) Improved two-stream model for human action recognition. EURASIP J Image Video Process 2020(1):1–9
53. Tu Z, Xie W, Qin Q, Poppe R, Veltkamp RC, Li B, Yuan J (2018) Multi-stream CNN: Learning representations based on human-related regions for action recognition. Pattern Recogn 79:32–43
54. Ma M, Marturi N, Li Y, Leonardis A, Stolkin R (2018) Region-sequence based six-stream CNN features for general and fine-grained human action recognition in videos. Pattern Recogn 76:506–521
55. Fang M, Peng S, Zhao Y, Yuan H, Hung CC, Liu S (2023) 3 s-STNet: three-stream spatial–temporal network with appearance and skeleton information learning for action recognition. Neural Comput Appl 35(2):1835–1848
56. Choi J, Jeon WJ, Lee SC (2008) Spatio-temporal pyramid matching for sports videos. In: Proceedings of the 1st ACM international conference on multimedia information retrieval, p 291–297
57. Li Z, Tang J (2015) Unsupervised feature selection via nonnegative spectral analysis and redundancy control. IEEE Trans Image Process 24:5343–5355
58. Li Z, Sun Y, Zhang L, Tang J (2021) CTNet: Context-based tandem network for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 44(12):9904–9917
59. Khan W, Hussain A, Kuru K, Al-Askar H (2020) Pupil localisation and eye centre estimation using machine learning and computer vision. Sensors 20(13):3785
60. Khan W, Ansell D, Kuru K, Amina M (2016) Automated aircraft instrument reading using real time video analysis. In: 2016 IEEE 8th international conference on intelligent systems (IS). IEEE, p 416–420
61. Singh D (2023) Graph representation for weakly-supervised spatio-temporal action detection. In: 2023 International joint conference on neural networks (IJCNN). IEEE, p 1–9
62. Wu Q, Huang Q, Li X (2023) Multimodal human action recognition based on spatio-temporal action representation recognition model. Multimed Tools Appl 82(11):16409–16430
63. Shen N, Feng Z, Li J, You H, Xia C (2023) Action fusion recognition model based on GAT-GRU binary classification networks for human-robot collaborative assembly. Multimed Tools Appl 82(12):18867–18885

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