Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 23: 1716 – 1730
© TÜBİTAK
doi:10.3906/elk-1910-71

Human activity recognition by using MHIs of frame sequences

Saeedeh ZEBHI, SMT AlMODARRESI[*], Vahid ABOOTALEBI
Electrical Engineering Department, Yazd University, Yazd, Iran

Received: 25.10.2019 • **Accepted/Published Online:** 02.03.2020 • **Final Version:** 08.05.2020

Abstract: A motion history image (MHI) is a temporal template that collapses temporal motion information into a single image in which intensity is a function of recency of motion. In recent years, the popularity of deep learning architectures for human activity recognition has encouraged us to explore the effectiveness of combining them and MHIs. Based on this, two new methods are introduced in this paper. In the first method, which is called the basic method, each video splits into N groups of consecutive frames, and the MHI is calculated for each group. Transfer learning with the fine-tuning technique is used for classifying these temporal templates. The experimental results show that some misclassification errors are created because of the similarities between these temporal templates; these errors can be corrected by detecting specific objects in the scenes. Thus, spatial information consisting of a single frame is also added to the second method, called the proposed method. By converting video classification problems into image classification problems in the proposed method, less memory is needed and the time complexity is greatly reduced. They are implemented and compared with state-of-the-art approaches on two data sets. The results show that the proposed method significantly outperforms the others. It achieves recognition accuracies of 92% and 92.4% for the UCF Sport and UCF-11 action data sets, respectively.

Key words: Motion history image, pretrained network, spatial stream, temporal stream

1. Introduction

Human action recognition is a great scene understanding ability for usages such as surveillance, health care, security, gaming, and intelligent environments. Different approaches have been used to solve the problem of action recognition [1–3]. They can be classified into vision-based and sensor-based approaches [4]. In vision-based approaches, a camera is used to capture information about human activities. Then computer vision techniques can be applied for recognizing the human activities. These approaches can give good results in proper light conditions but privacy is the main concern. In sensor-based approaches, human behavior is recorded by different sensors. Sensor-based approaches can be divided into three classes, namely wearable, object-tagged, and dense sensing [5]. These approaches have attracted much attention due to their low cost and the advancement in sensor technology in recent years. In the wearable approach, humans have to carry the sensors when they perform any activity, but wearing a tag is not feasible sometimes. In the object-tagged approach, sensors have to be attached to objects in daily use. In this case, users must use tagged-objects and so it forces the users to use specific objects. In the dense sensing approach, sensors are deployed in the environment (in which the activity is being performed) and the data will be collected through those sensors when a person performs any activity. The dense sensing approach is more practical and it does not have the limitations of the two previous

[*] Correspondence: Correspondence: smta@yazd.ac.ir

approaches. Nevertheless, vision-based approaches are focused on in this article because they can give good results and applying computer vision-based techniques is easy.

First, global representations extract global descriptors directly from main videos or images and encode them as features. Optical flow achieved by the Lucas–Kanade–Tomasi (LKT) feature tracker [6, 7], the motion energy image (MEI) and the motion history image (MHI) (which were presented by Bobick and Davis [8, 9]), 3D space–time volumes (STVs) [10], and discrete Fourier transform (DFT) have been introduced as these descriptors. Later, local descriptors such as the histogram of optical flow (HOF), speeded-up robust features (SURF) [11], 2D scale-invariant feature transform (SIFT) [12], 3D SIFT [13], the histogram of oriented gradients (HOG) [14], and the histogram of 3D-oriented gradients (HOG3D) [15] were presented. For constructing the last of these descriptors, local patches are calculated using interest point detectors or dense sampling [15]. Most local features are demonstrated to be robust against noise and partial obstructions compared to global features.

There have been many studies on human action recognition in videos that use these hand-crafted features [16] and input them into the support vector machine (SVM) classifier. In Kovashka and Grauman [17], each action was represented as an ensemble of local spatiotemporal feature vectors corresponding to sparse interest points extracted from videos, and human action recognition was done by analyzing these spatiotemporal feature vectors. Later, the improved dense trajectories (iDTs) [18] achieved the most outstanding performance as a hand-crafted feature. Cho et al. [19] used traditional multicore sparse representation to express video local motion features and global motion features. This method required learning the dictionary to express local motion features and was limited by the dictionary's expressive ability. In 2016, Abdulmunem et al. [20] introduced an approach considering saliency guided features. With saliency guidance, local and global features were extracted for encoding video information. Thus the saliency guided 3D SIFT-HOOF (SGSH) feature was used for feature representation. Simultaneously, Souly and Shah [21] used group lasso regularization to find the sparse representation for video saliency detection.

Recently, deep neural networks (DNNs) have attracted much attention in many areas such as object detection, object tracking, image segmentation, image classification, and action localization [22], because the feature construction process is automated. Image classification networks with high efficiency cause them to be used with little change for video. Therefore, by using them in Karpathy et al. [23], features are extracted independently from each frame and then their predictions are pooled across the entire video. In fact, this approach completely ignores temporal structure. Moreover, 2D convolutional neural networks (2D CNNs) are widely used in image analysis applications but they lose the ability to obtain motion information in terms of a 3D volume of video frames. Among the 2D CNN methods, it is preferable to add a recurrent layer such as an LSTM to the model [24, 25]. It can code state and acquire temporal ordering. By adding it to the last layers of the 2D CNN model, high-level changes are modeled, but small low-level motion that is important in many cases may not be detected. It is also expensive to train as it requires unrolling the network through several frames for back-propagation-through-time [26]. Sharma et al. [27] proposed a soft-attention LSTM model to selectively focus on parts of the video frames and classify videos after taking a few glimpses. Later, Zhou et al. [28] showed that a low dimension feature representation generated on the deep convolutional layers is more discriminative compared to traditional CNN features, which explore the outputs from the fully connected layers in a CNN. In addition, based on the convolutional feature maps, a multiscale pooling strategy was used to better handle the objects with different scales and deformations.

The 3D convolutional neural network (3D CNN) architecture [29–31] is one of the most usual deep models. By implementing 3D convolutions, features are extracted from both the spatial and the temporal dimensions, thereby acquiring the motion information coded in several neighbor frames. It works better than single frame baseline [31] since the motion features have been learned adequately. Because this architecture has many more parameters than a 2D CNN due to the additional kernel dimension, its training is more complex.

Simonyan and Zisserman [32] introduced a different approach. They thought that video could be decomposed into spatial and temporal components. The spatial component carries information about scenes and objects that are presented in the video. The temporal component gives the moving information of the observer (the camera) and the objects. Motion features were clearly formed in the shape of stacked optical flow vectors and so their architecture has two separate networks: one for spatial content and the other for motion content. The input to the spatial net is a single frame of the video and the input to the temporal net was tested by authors. They concluded that bidirectional optical flow stacked across ten frames performed best. The two streams were trained separately and combined using an SVM. This method performed better than the single stream method by obviously capturing local temporal movement. Since then, many video classification approaches have followed this design and have made improvements in terms of new representation [33], different backbone architecture [34, 35], and exploiting richer temporal structures [24, 36].

Ravanbakhsh et al. [37] first computed CNN features for each frame of a video and then mapped them into a short binary code space. For selecting key frames, the changes in each bit of the binary codes were tracked across time. Video snippets between consecutive key frames were fed into a hierarchical decomposition. PCA was applied to each level of the hierarchy to reduce the dimension within each level. All levels were stacked as a vector representation for a video snippet. In the last step, a histogram of temporal words was built for all the videos and a classifier was trained to predict action labels.

Gammulle et al. [38] extracted features frame-wise from each video through a CNN and passed those features sequentially through an LSTM in order to classify the sequence. Based on this, four CNN-to-LSTM fusion models were introduced. The architectures varied when it was considered how the features from the CNN model were fed to the LSTM action classification model. Two direct mapping models, convolutional-to-LSTM (conv-L) and fully connected-to-LSTM (fc-L), and two merged models, conv-L+fc-L fusion method 1 (fu-1) and conv-L+fc-L fusion method 2 (fu-2), were considered. The direct mapping models had less accuracy compared to the merged models, and the model conv-L had the lowest accuracy due to the highly dense features from the convolutional layer that were difficult to discriminate. In the conv-L+fc-L fusion method 1 (fu-1), the outputs of the first two models were merged and passed through a soft-max layer to evaluate the final classification. Two types of LSTMs called sequence-to-one LSTM and sequence-to-sequence LSTM were used with their last model such that the resultant sequence of predictions was fed to the final LSTM, which worked in a sequence-to-one LSTM. Further performance improvement was gained by adding a third LSTM layer for combining the two streams.

Wang et al. [39] proposed a lightweight architecture for video action recognition, which consisted of a CNN, LSTM, and attention model. They used the convolution model to extract two kinds of features (semantic, spatial) for each frame, followed by FC-LSTM with their temporal-wise attention model. This architecture was designed for obtaining strong representational power for predicting just using RGB data, without optical flow data needing additional computations. The weakness of this architecture was that the spatial information was not encoded by the input-to-state or state-to-state transition in FC-LSTM.

As mentioned before, more 3D-CNN based approaches have high computational complexity. They concentrate on the problem of understanding the content of videos, which does not essentially require the modeling of their dynamics. In addition, the spatiotemporal filters of CNNs that maximize the recognition ability of the total system in an end-to-end manner should be learned. On the other hand, the high efficiency of pretrained networks in image classification problems encourages us to use them with little change for video.

The first contribution of the present paper is converting the video classification problem into an image classification one. To achieve this purpose, we just need to construct some simple and informative templates that can effectively display the temporal and spatial information of videos. In addition, these templates should not have high computational complexity. Thus, a global descriptor like MHI and one single frame of each video are used for saving temporal and spatial information of videos, respectively. Furthermore, finding the effective representation (combination of MHI and one single frame) for video classification is the second contribution. Accordingly, two new methods are introduced, which are called the basic and proposed methods. In the basic method, only the temporal stream based on MHI is used. The spatial stream is added to the proposed method, which increases efficiency significantly. Pretrained network and fine-tuning techniques are used for the classification problem. The rest of this paper is organized as follows. First, constructing the MHI is explained. Then the two new methods are presented and are implemented on two data sets. Finally, the results are compared with those of the state-of-the-art approaches.

2. Motion history image (MHI)

A MHI [9] is a static image in which the motion history of captured sequences at each point is represented by its pixel intensity. It is defined by the following equation:

$$H_\tau(x,y,t) = \begin{cases} \tau & if\ D(x,y,t)=1 \\ max(0, H_\tau(x,y,t-1)-1) & otherwise \end{cases}, D(x,y,t) = B(x,y,t+1) - B(x,y,t), \quad (1)$$

where H_τ is the MHI and $D(x,y,t)$ is the binary difference image between two consecutive preprocessed binary sequences $B(x,y,t)$ and $B(x,y,t+1)$. Variables x and y are image pixel coordinates and t is the temporal index. The variable τ is a threshold for extraction moving patterns in video image sequences. The equation $D(x,y,t)=1$ shows a motion occurrence in the t time on the coordinate point (x,y). Thus, the MHI is a scalar-valued image with this feature in which more recently moving pixels are brighter than others.

Based on MHI usages, various methods can be applied for constructing the difference image. One of them is the construction of the background image and subtraction of the frames from this background image [40]. In this usage, similar to Keskin et al. [41], the difference in consecutive frames is applied because there is no information about the background. Examples of MHIs constructed for 20 consecutive frames are shown in Figure 1. As is obvious from this figure, more recently moving pixels are brighter than other pixels. Indeed, MHIs show the direction of motion and this feature is used for action recognition in the following sections.

As seen, MHIs show temporal changes and the direction of motion. For example, Figure 1a shows that a person is lifting; also pixel intensities show the direction of motion such that more recently moving pixels are brighter than others. In other words, the MHI is a function of time and its pixel intensity on the coordinate point (x,y) is a function of the temporal history of motion at that point.

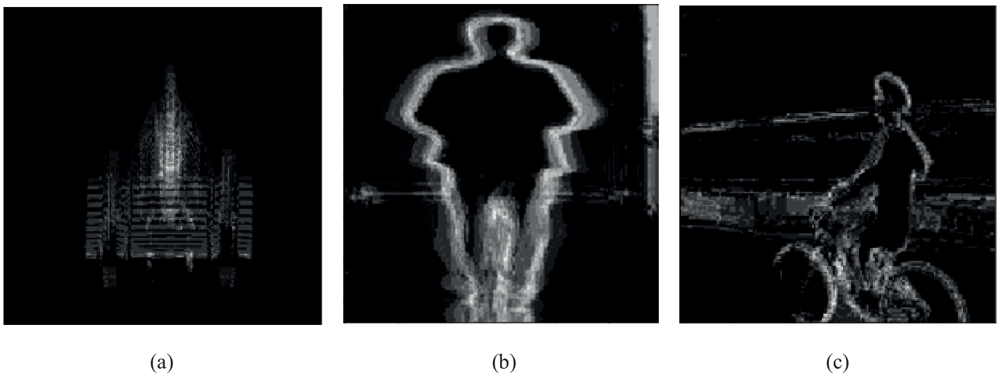


Figure 1. MHIs for 20 consecutive frames: (a) lifting, (b) walking-front, (c) biking.

3. Methods

In this section, two new methods are introduced. In the first method, which is called the basic method, only the temporal stream (MHIs of frame sequences) is used. The spatial stream (a single frame) is added to the second method, called the proposed method.

3.1. Basic method: MHIs of frame sequences

First, each video splits into N groups of consecutive frames. MHIs are calculated for each group, creating MHIs of frame sequences. Then MHIs of frame sequences are fed to the classifier. These steps are shown in Figure 2.

By using this approach, motion information in N groups of consecutive frames is abstracted in MHIs of frame sequences for each video. It is clear that N has an important role. A small value of N causes the temporal differences of the first frames in actions to be ignored; on the other hand, by selecting a large value of N, fine temporal differences in each group are considered. Therefore, it is necessary to select the optimal value for N to maximize accuracy.

Designing a CNN from scratch is complex and time consuming, and training the weights from random values increases computational complexity. For classifying, transfer learning with the fine-tuning technique is used. Transfer learning is the reuse of a pretrained model on a new also relevant problem. VGG-16 is a convolutional neural network that is trained on more than one million images from the ImageNet database[1]. For the classifying problem with this pretrained network, two densely connected classifiers were added after the flatten layer on top of the convolutional base before compiling the model. The neuron numbers for these two fully connected layers are 100 and C neurons, respectively, where C is the number of classes. To prevent overfitting, a dropout layer is added after the first dense layer. Fine-tuning consists of unfreezing a few of the top layers of a frozen model base used for feature extraction and jointly training the newly added part of the model and these unfrozen layers. For this purpose, all Conv blocks are frozen except Conv block 5 and the newly added top layers. The rectified linear unit (ReLU) is used for the activation function. Classification is done with the softmax function and cross-entropy loss function. The details of the VGG-16 convolutional base architecture and its settings are shown in Figure 3.

It is worth noting that this pretrained network should have exactly three input channels, but MHIs of frame sequences are grayscale images. The images simply need to appear to be RGB. Therefore, the images are

[1] ImageNet (2018). ImageNet database [online]. Website https://www.image-net.org/ [accessed 21 10 2018]

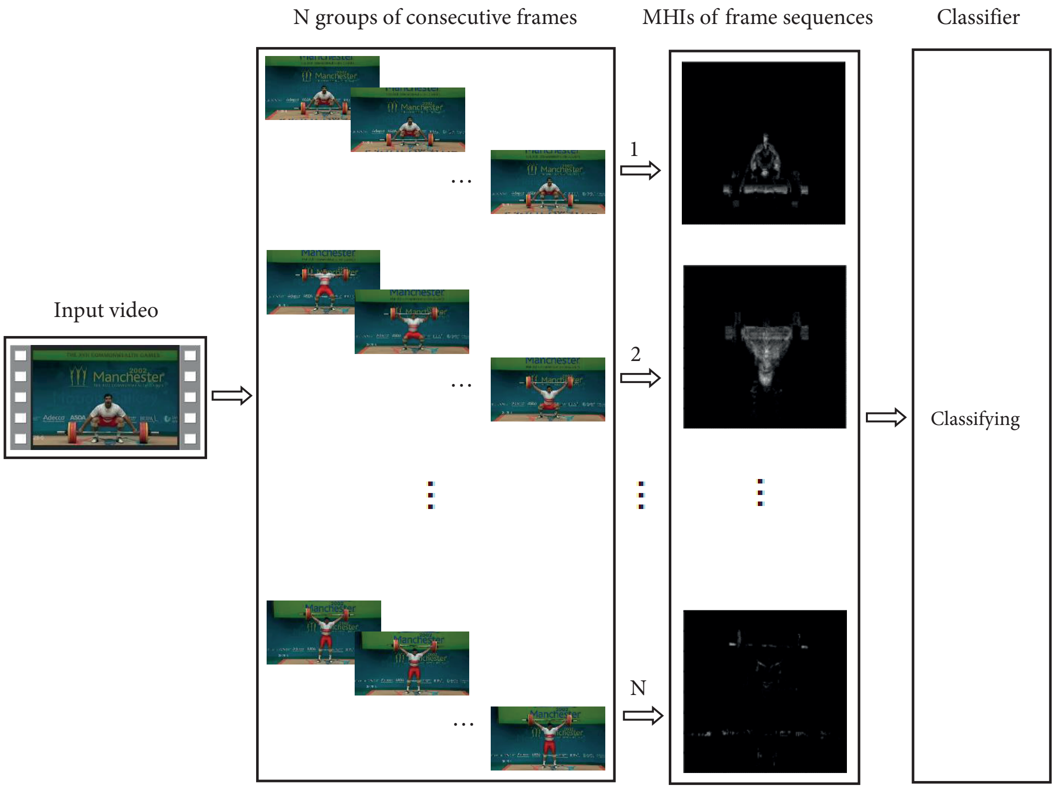


Figure 2. Flow graph of the basic method.

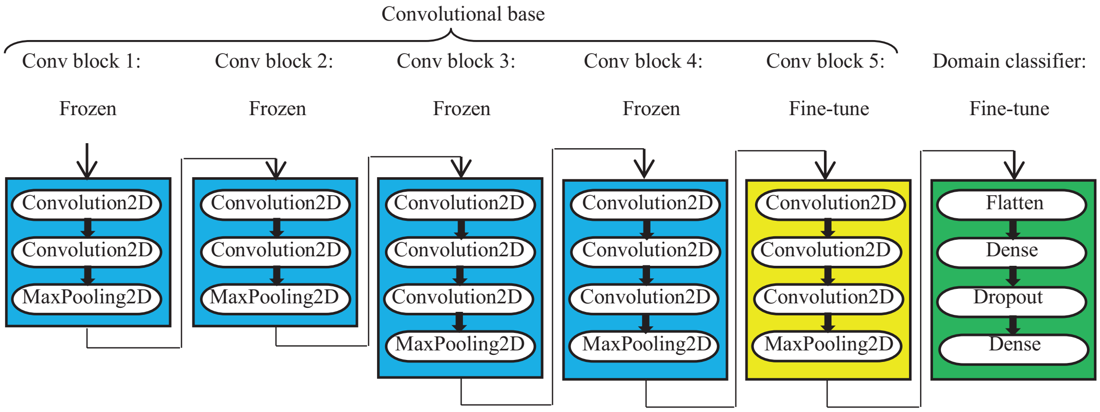


Figure 3. Pretrained network (VGG-16) and fine-tuning.

repeated three times on a new dimension. We will have them over all three channels, and the performance of this network should be the same as it was for RGB images. This technique is shown in Figure 4. Therefore, by using this technique the video classification problem is converted into an image classification one and training the network will be much easier and faster.

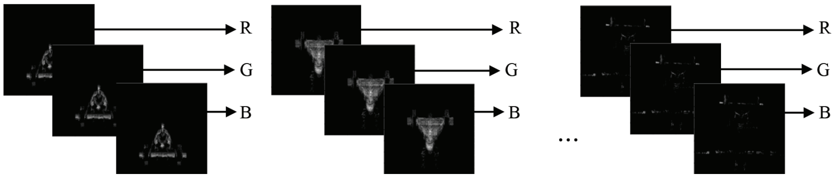


Figure 4. Expand grayscale images to RGB images.

3.2. Proposed method: MHIs of frame sequences + spatial

Besides the temporal information used in the basic method, some actions are strongly dependent on particular objects. In fact, some actions with similar motions (or subsequently similar MHIs of frame sequences) cannot be distinguished with only the temporal stream. One single frame of each video was also selected for the spatial stream. For selecting this frame, each video was split into three sections. Experiments showed that choosing a random sample frame from the first or last (third) section was not desirable because of low spatial accuracy. Presumably, specific objects have not entered the scenes yet, or they have left the scenes such that most of the spatial information (specific object) exists in the second section. By choosing random sample frames from the second section, the same accuracies will be obtained. A pretrained network (VGG-16) and fine-tuning are also used here for image classification architecture. An overview of the proposed method is shown in Figure 5.

As shown in Figure 5, two separate streams are used for classification. A single frame of each video and MHIs of frame sequences are used for spatial and temporal stream recognition, respectively. Both streams have the same architecture. Each stream is implemented using a pretrained network (VGG-16) and fine-tuning, the same as the basic method. For each video, a probability matrix of size N by C is achieved with a temporal stream, where C is the number of classes (columns) that exist in the data set, and N is the number of frame sequences (rows). The sum of each row of the matrix is 1. Similarly, a probability matrix of size 1 by C is acquired with a spatial stream for each video. By averaging the N+1 row matrices, a predictive row matrix of size 1 by C is generated. Finally, the column label that has the maximum value shows the predicted class. Thus late average fusion is used for combining softmax scores and acquiring the final decision.

4. Experiments

In this section, experiments were performed on two action data sets for comparing these two methods with state-of-the-art methods.

4.1. Data sets

Two benchmark data sets were considered, which include the UCF Sport [42] and UCF-11 [43]. Sample frames from these data sets are presented in Figure 6.

The UCF Sport data set includes 150 video sequences with a resolution of 720×480. It comprises 10 actions that include walking, running, kicking, lifting, diving, golf swing, riding horse, skate boarding, swinging-side, and swinging-bench. These actions are performed in different real environments that cover various viewpoints and include a lot of camera motion.

The UCF-11 action data set is a challenging data set because of large variations in camera motion, illumination, viewpoint, background clutter, etc. It contains 1600 videos, each labeled as one of the following 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog.

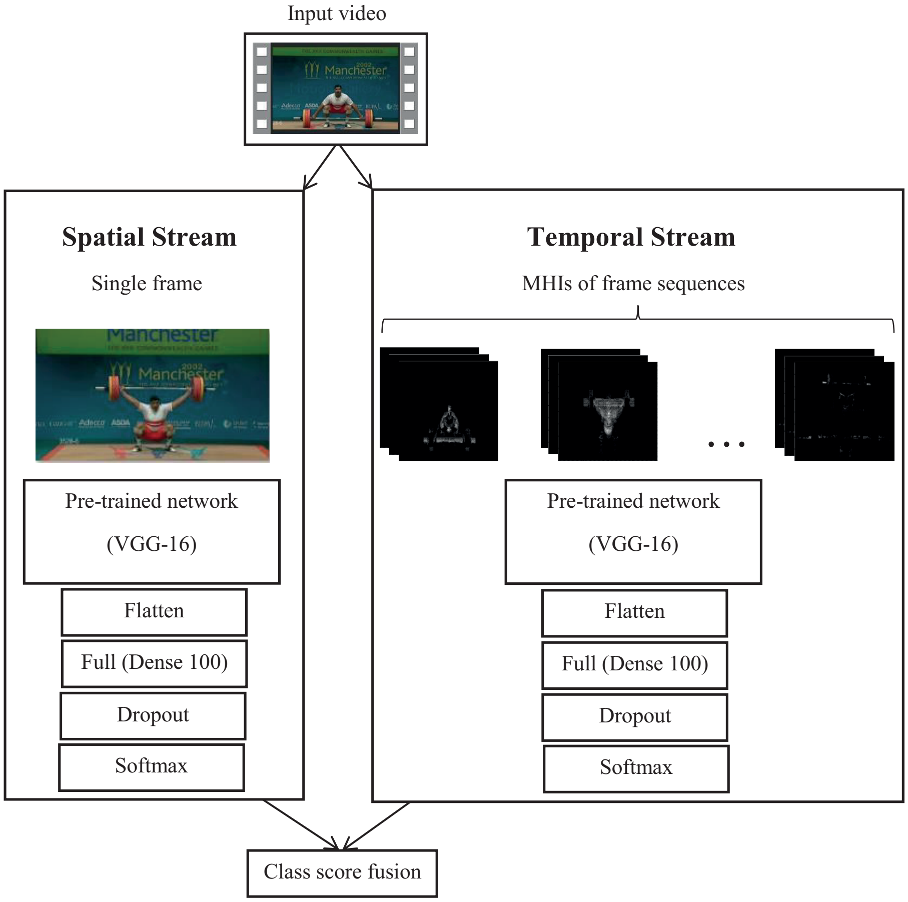


Figure 5. Flow graph of the proposed method.

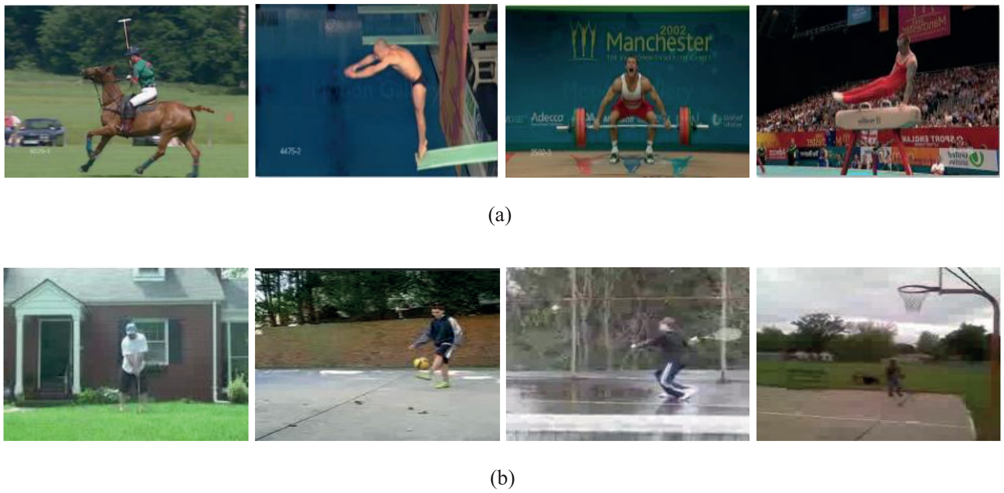
A summary of the explained data sets is presented in Table 1, in order to present the information more clearly.

Table 1. Summary of the explained data sets.

Data sets	Classes	Videos	Resolution	Year
UCF Sport	10	150	720 × 480	2008
UCF–11	11	1600	–	2009

4.2. Experimental setting

Videos contain various numbers of frames. They are divided into N groups of consecutive frames. MHIs of frame sequences are calculated for these groups of frames. Considering $N = 1$ does not seem sensible to achieve high efficiency, the assumption that temporal changes that occur in the initial frames are ignored. The neuron number for the last fully connected layer is equal to 10 and 11 for the UCF Sport and UCF-11 action data sets, respectively. RMSprop with a learning rate of 0.0001 is used for the optimization of the two methods. RMSprop (root mean square propagation) almost immediately heads off in the right direction, scales the learning rate,



(a)

(b)

Figure 6. Samples of experimental action data sets: (a) UCF Sport, (b) UCF-11 action data set.

converges fast, and performs well in practice. Moreover, it is recommended to leave the parameters of RMSprop at their default values (except the learning rate, which can be freely tuned). Due to these properties, RMSprop is chosen as an optimization method for classifier training. N starts from 2 and increases until the best result is achieved. The number of epochs is set to 100, the batch size is fixed at 10, and the dropout rate is set to 0.9 in dropout layers to prevent overfitting. Moreover, such as in Liu et al. [43], five-fold cross-validation is used for experimental setups. Experiments have been implemented on a Python-based deep learning library called Keras[2] with one NVIDIA GeForce GTX 1050 card and 8 GB RAM.

5. Results and discussion

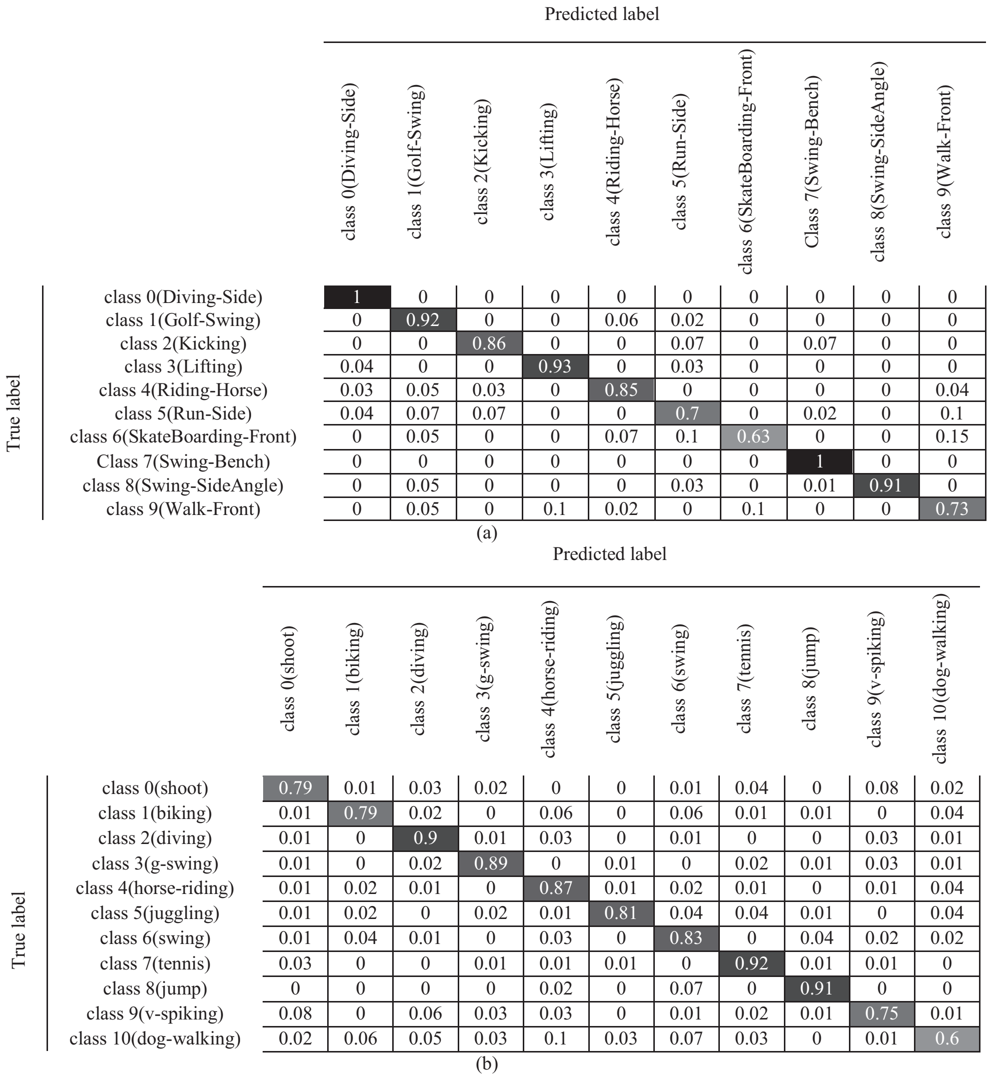
The basic method was initially applied to these data sets. The results for different N-values are listed in Table 2. As shown, the best efficiencies are achieved with $N = 3$ and $N = 4$ for UCF Sport and UCF-11, respectively. Confusion matrices for the best efficiencies are shown in Figures 7a and 7b. The time complexity of computing MHIs of frame sequences (N temporal templates) for each video is 1.05 s, and this time is independent of the value of N.

Table 2. Accuracy of the basic method.

UCF Sport				
Training setting	$N = 2$	$N = 3$	$N = 4$	$N = 5$
Basic method	79.6 ± 3.7%	85.3 ± 3.9%	82.4 ± 5.2%	80.2 ± 5.7%
UCF–11				
Training setting	$N = 2$	$N = 3$	$N = 4$	$N = 5$
Basic method	54.4 ± 1.1%	82 ± 1.2%	82.4 ± 0.7%	80.7 ± 1.6%

The proposed method was applied to these data sets, and the results are summarized in Table 3. The results show that temporal information is not sufficient for acquiring high accuracy and so spatial information is added to the proposed method. As shown, by considering two streams and fusing them, the standard deviation

[2] Keras (2015). Chollet F [online]. Website https://github.com/fchollet/keras/ [accessed 04 07 2018]



Predicted label

True label		class 0(Diving-Side)	class 1(Golf-Swing)	class 2(Kicking)	class 3(Lifting)	class 4(Riding-Horse)	class 5(Run-Side)	class 6(SkateBoarding-Front)	Class 7(Swing-Bench)	class 8(Swing-SideAngle)	class 9(Walk-Front)
	class 0(Diving-Side)	1	0	0	0	0	0	0	0	0	0
	class 1(Golf-Swing)	0	0.92	0	0	0.06	0.02	0	0	0	0
	class 2(Kicking)	0	0	0.86	0	0	0.07	0	0.07	0	0
	class 3(Lifting)	0.04	0	0	0.93	0	0.03	0	0	0	0
	class 4(Riding-Horse)	0.03	0.05	0.03	0	0.85	0	0	0	0	0.04
	class 5(Run-Side)	0.04	0.07	0.07	0	0	0.7	0	0.02	0	0.1
	class 6(SkateBoarding-Front)	0	0.05	0	0	0.07	0.1	0.63	0	0	0.15
	Class 7(Swing-Bench)	0	0	0	0	0	0	0	1	0	0
	class 8(Swing-SideAngle)	0	0.05	0	0	0	0.03	0	0.01	0.91	0
	class 9(Walk-Front)	0	0.05	0	0.1	0.02	0	0.1	0	0	0.73

(a)

Predicted label

True label		class 0(shoot)	class 1(biking)	class 2(diving)	class 3(g-swing)	class 4(horse-riding)	class 5(juggling)	class 6(swing)	class 7(tennis)	class 8(jump)	class 9(v-spiking)	class 10(dog-walking)
	class 0(shoot)	0.79	0.01	0.03	0.02	0	0	0.01	0.04	0	0.08	0.02
	class 1(biking)	0.01	0.79	0.02	0	0.06	0	0.06	0.01	0.01	0	0.04
	class 2(diving)	0.01	0	0.9	0.01	0.03	0	0.01	0	0	0.03	0.01
	class 3(g-swing)	0.01	0	0.02	0.89	0	0.01	0	0.02	0.01	0.03	0.01
	class 4(horse-riding)	0.01	0.02	0.01	0	0.87	0.01	0.02	0.01	0	0.01	0.04
	class 5(juggling)	0.01	0.02	0	0.02	0.01	0.81	0.04	0.04	0.01	0	0.04
	class 6(swing)	0.01	0.04	0.01	0	0.03	0	0.83	0	0.04	0.02	0.02
	class 7(tennis)	0.03	0	0	0.01	0.01	0.01	0	0.92	0.01	0.01	0
	class 8(jump)	0	0	0	0	0.02	0	0.07	0	0.91	0	0
	class 9(v-spiking)	0.08	0	0.06	0.03	0.03	0	0.01	0.02	0.01	0.75	0.01
	class 10(dog-walking)	0.02	0.06	0.05	0.03	0.1	0.03	0.07	0.03	0	0.01	0.6

(b)

Figure 7. Confusion matrices resulting from the basic method for (a) UCF Sport, (b) UCF-11 data sets.

is decreased and the accuracy is effectively improved. This performance improvement is obvious on both data sets, as the proposed method performs better than the basic method with an improvement of 6.7% for UCF Sport and 10% for UCF-11. Confusion matrices for these improved results are shown in Figure 8. Furthermore, the time complexity of this method is similar to that of the basic method.

Table 3. Results of individual streams and fusing them.

UCF Sport		
Basic method (temporal stream)	Spatial stream	Proposed method (temporal stream + spatial stream)
85.3 ± 3.9%	62 ± 6.1%	92 ± 1.6%
UCF–11		
Basic method (temporal stream)	Spatial stream	Proposed method (temporal stream + spatial stream)
82.4 ± 0.7%	85.7 ± 2.9%	92.4 ± 0.4%

Predicted label

True label

	class 0(Diving-Side)	class 1(Golf-Swing)	class 2(Kicking)	class 3(Lifting)	class 4(Riding-Horse)	class 5(Run-Side)	class 6(SkateBoarding-Front)	Class 7(Swing-Bench)	class 8(Swing-SideAngle)	class 9(Walk-Front)
class 0(Diving-Side)	1	0	0	0	0	0	0	0	0	0
class 1(Golf-Swing)	0	0.87	0	0	0	0.08	0	0	0	0.05
class 2(Kicking)	0	0	0.96	0	0	0	0	0.04	0	0
class 3(Lifting)	0	0	0	0.9	0	0	0	0	0	0.1
class 4(Riding-Horse)	0	0	0	0	0.93	0	0	0	0	0.07
class 5(Run-Side)	0	0.04	0.05	0	0	0.81	0	0	0	0.1
class 6(SkateBoarding-Front)	0	0	0	0	0	0	0.86	0	0	0.14
Class 7(Swing-Bench)	0	0	0	0	0.06	0	0	0.94	0	0
class 8(Swing-SideAngle)	0	0	0	0	0	0	0	0	1	0
class 9(Walk-Front)	0	0	0	0.07	0	0	0	0	0	0.93

(a)

Predicted label

True label

	class 0(shoot)	class 1(biking)	class 2(diving)	class 3(g-swing)	class 4(horse-riding)	class 5(juggling)	class 6(swing)	class 7(tennis)	class 8(jump)	class 9(v-spiking)	class 10(dog-walking)
class 0(shoot)	0.88	0	0.03	0.01	0	0	0	0.02	0.01	0.05	0
class 1(biking)	0.01	0.9	0.01	0	0.04	0	0.01	0	0.01	0	0.02
class 2(diving)	0	0	0.98	0	0	0	0	0	0	0.01	0.01
class 3(g-swing)	0	0	0	0.95	0	0	0	0	0.01	0.02	0.02
class 4(horse-riding)	0	0	0.01	0	0.95	0.02	0.01	0	0.01	0	0
class 5(juggling)	0	0.02	0	0.02	0	0.93	0.01	0	0	0	0.02
class 6(swing)	0	0.02	0	0	0	0	0.95	0	0.02	0	0.01
class 7(tennis)	0.01	0	0	0	0	0	0	0.99	0	0	0
class 8(jump)	0	0	0	0.01	0	0	0.04	0	0.95	0	0
class 9(v-spiking)	0.01	0	0.05	0	0	0	0.01	0	0	0.93	0
class 10(dog-walking)	0.01	0.02	0.03	0	0.09	0.04	0.06	0	0	0	0.75

(b)

Figure 8. Confusion matrices resulting from proposed method for (a) UCF Sport, (b) UCF-11 data sets.

These matrices show that for UCF Sport in the basic method, most misclassification errors occur between Run-Side, SkateBoarding-Front (true labels), and Walk-Front (predicted label). It seems that by splitting each video into N groups of consecutive frames and changing the view angle of the camera throughout the videos, MHIs of frame sequences acquired from these actions will be similar. In addition, dynamic backgrounds increase this problem. This problem also exists in the proposed method, but some misclassification errors are resolved by detecting specific objects in the scenes. For example, by detecting skate, some errors that occurred between SkateBoarding-Front (true label) and Run-Side, Riding-Horse, and Golf-Swing (predicted labels) are removed.

For UCF-11, most misclassification errors occur between these classes (shoot and v-spiking, dog-walking, and horse-riding). Presumably, similar motions of the ball and persons in MHIs of frame sequences from shoot and v-spiking videos can create these errors. Similar MHIs of frame sequences can also be created with dog-walking and horse-riding because of the similarity of these activities. In the proposed method, this problem is partly fixed with the spatial stream by detecting the type of tour in shoot and v-spiking and distinguishing horse and dog shapes. The spatial stream can correct errors that are produced between two completely different classes (dog-walking (true label) and biking (predicted label), jump (true label), and swing (predicted label)) by detecting different objects (e.g., dog, bike, swing).

The performance of these two methods has also been compared with other methods on these data sets in Table 4. As seen in this table, the proposed method achieved better performance than the others. By using the proposed method, each video has abstracted to 1 and N templates for spatial and temporal streams, respectively, and there is no more need to save all frames of videos. Only 1.05 s is needed for computing the MHIs of frame sequences (N temporal templates) for each video, and this time is independent of the value of N. Thus less memory is needed and the time complexity is greatly reduced. Furthermore, by converting the video classification problem into an image classification one, network training is much easier and faster compared to other methods like 3D-CNN-based approaches.

Table 4. Comparison with other methods on UCF Sport and UCF-11.

UCF Sport	
Methods	Accuracy
(Souly and Shah, 2016) [21]	85.10%
(Wang et al., 2009) [16]	85.60%
(Le et al., 2011) [1]	86.50%
(Kovashka and Grauman, 2010) [17]	87.20%
(Wang et al., 2011) [18]	89.10%
(Weinzaepfel et al., 2015) [22]	90.50%
(Abdulmunem et al., 2016) [20]	90.90%
(Ravanbakhsh et al., 2015) [37]	88.10%
(Wang et al., 2018) [39]	91.89%
(Zhou et al., 2017) [28]	90.00%
Basic method	85.30%
Proposed method	92.00%
UCF-11	
Methods	Accuracy
(Hasan et al., 2014) [2]	54.50%
(Liu et al., 2009) [43]	71.20%
(Ikizler-Cinbis et al., 2010) [36]	75.20%
(Wang et al., 2011) [18]	84.20%
(Sharma et al., 2015) [27]	84.90%
(Cho et al., 2014) [19]	88.00%
(Ravanbakhsh et al., 2015) [37]	77.10%
(Wang et al., 2018) [39]	98.76%
(Gammulle et al., 2017) [38]	89.20%
(Gilbert et al., 2017) [3]	86.70%
Basic method	82.40%
Proposed method	92.40%

6. Conclusion

The main idea of this work is converting the video classification problem into an image classification one. Based on this idea, two new methods are introduced. In the basic method, each video splits into N groups of consecutive frames and MHIs are calculated for them. MHIs of frame sequences show temporal changes and the direction of motion. Results of the basic method show that some misclassification errors are created due to the similarity of these temporal templates and can be resolved by detecting specific objects in the scenes. In the proposed method, MHIs of frame sequences and one single frame are used to train the temporal and spatial streams, respectively. The final decision is achieved by applying the average fusion method. These two methods have been tested on two common data sets. The results show that the proposed method achieves better performance compared to the state-of-the-art methods. By using this proposed method, each video has been abstracted to N+1 images, and there is no more need to save all frames of videos. The time complexity is greatly reduced because of the simplicity of temporal template calculations. Moreover, network training is also much easier and faster compared to other methods like 3D-CNN-based approaches. In future work, we will attempt to amend these temporal templates for confronting interclass similarities such as SkateBoarding-Front and Walk-Front classes.

References

[1] Le QV, Zou WY, Yeung SY, Ng AY. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR 2011; Colorado, USA; 2011. pp. 3361-3368.

[2] Hasan M, Roy-Chowdhury AK. Incremental activity modeling and recognition in streaming videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014. pp. 796-803.

[3] Gilbert A, Bowden R. Image and video mining through online learning. Computer Vision and Image Understanding 2017; 158: 72-84.

[4] Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z. Sensor-based activity recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part C Applications and Reviews 2012; 42 (6): 790-808.

[5] Hussain Z, Sheng M, Zhang WE. Different approaches for human activity recognition: a survey. arXiv preprint arXiv:1906.05074. 2019 Jun 11.

[6] Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial intelligence - Volume 2 (IJCAI'81). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA; 1981. pp. 674-679.

[7] Shi J, Tomasi C. Good Features to Track. Ithaca, NY, USA: Cornell University, 1993.

[8] Bobick A, Davis J. An appearance-based representation of action. In: Proceedings of 13th International Conference on Pattern Recognition. Vol. 1. IEEE; 1996. pp. 307-312

[9] Bobick AF, Davis JW. The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2001; 1 (3): 257-267.

[10] Blank M, Gorelick L, Shechtman E, Irani M, Basri R. Actions as space-time shapes. In: Tenth IEEE International Conference on Computer Vision (ICCV'05). IEEE; 2005. pp. 1395-1402

[11] Bay H, Ess A, Tuytelaars T, Van Gool L. Speeded-up robust features (SURF). Computer Vision and Image Understanding 2008; 110 (3): 346-359.

[12] Lowe DG. Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision (ICCV '99). IEEE Computer Society, USA; 1999. pp. 1150-1157.

[13] Scovanner P, Ali S, Shah M. A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia. ACM; 2007. pp. 357-360.

[14] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: International Conference on Computer Vision & Pattern Recognition (CVPR'05). IEEE Computer Society; 2005. pp. 886-893.

[15] Klaser A, Marszałek M, Schmid C. A spatiotemporal descriptor based on 3d-gradients. In: BMVC 2008-19th British Machine Vision Conference. British Machine Vision Association; 2008. pp. 275-281.

[16] Wang H, Ullah MM, Klaser A, Laptev I, Schmid C. Evaluation of local spatiotemporal features for action recognition. In: BMVC 2009-British Machine Vision Conference. BMVA Press; 2009. pp. 124-131.

[17] Kovashka A, Grauman K. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE; 2010. pp. 2046-2053.

[18] Wang H, Kläser A, Schmid C, Cheng-Lin L. Action recognition by dense trajectories. In: CVPR 2011-IEEE Conference on Computer Vision & Pattern Recognition. IEEE; 2011. pp. 3169-3176.

[19] Cho J, Lee M, Chang HJ, Oh S. Robust action recognition using local motion and group sparsity. Pattern Recognition 2014; 47 (5): 1813-1825.

[20] Abdulmunem A, Lai YK, Sun X. Saliency guided local and global descriptors for effective action recognition. Computational Visual Media 2016; 2 (1): 97-106.

[21] Souly N, Shah M. Visual saliency detection using group lasso regularization in videos of natural scenes. International Journal of Computer Vision 2016; 117 (1): 93-110.

[22] Weinzaepfel P, Harchaoui Z, Schmid C. Learning to track for spatiotemporal action localization. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE; 2015. pp. 3164-3172.

[23] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R et al. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2014. pp. 1725-1732.

[24] Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S et al. Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2015. pp. 2625-2634.

[25] Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R et al. Beyond short snippets: deep networks for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2015. pp. 4694-4702.

[26] Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2017. pp. 6299-6308.

[27] Sharma S, Kiros R, Salakhutdinov R. Action recognition using visual attention. arXiv preprint arXiv:1511.04119. 2015 Nov 12.

[28] Zhou Y, Pu N, Qian L, Wu S, Xiao G. Human action recognition in videos of realistic scenes based on multi-scale CNN feature. In: Pacific Rim Conference on Multimedia. Springer; 2017. pp. 316-326.

[29] Ji S, Xu W, Yang M, Yu K. 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2012; 35 (1): 221-231.

[30] Taylor GW, Fergus R, LeCun Y, Bregler C. Convolutional learning of spatiotemporal features. In: European Conference on Computer Vision. Springer; 2010. pp. 140-153.

[31] Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision; Santiago, Chile; 2015. pp. 4489-4497.

[32] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems. 2014. pp. 568-576.

[33] Bilen H, Fernando B, Gavves E, Vedaldi A. Action recognition with dynamic image networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017; 40 (12): 2799-2813.

[34] Feichtenhofer C, Pinz A, Wildes RP. Temporal residual networks for dynamic scene recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2017. pp. 4728-4737.

[35] Wang L, Xiong Y, Wang Z, Qiao Y. Towards good practices for very deep two-stream convnets. arXiv preprint arXiv:1507.02159. 2015 Jul 8.

[36] Ikizler-Cinbis N, Sclaroff S. Object, scene and actions: combining multiple features for human action recognition. In: European Conference on Computer Vision. Springer; 2010. pp. 494-507.

[37] Ravanbakhsh M, Mousavi H, Rastegari M, Murino V, Davis LS. Action recognition with image based CNN features. arXiv preprint arXiv:1512.03980. 2015 Dec 13.

[38] Gammulle H, Denman S, Sridharan S, Fookes C. Two stream LSTM: a deep fusion framework for human action recognition. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE; 2017. pp. 177-186.

[39] Wang L, Xu Y, Cheng J, Xia H, Yin J et al. Human action recognition by learning spatiotemporal features with deep neural networks. IEEE Access 2018; 6: 17913-17922.

[40] Davis JW. Hierarchical motion history images for recognizing human motion. In: Proceedings IEEE Workshop on Detection and Recognition of Events in Video. IEEE; 2001. pp. 39-46.

[41] Keskin C, Erkan A, Akarun L. Real time hand tracking and 3D gesture recognition for interactive interfaces using HMM. ICANN/ICONIPP 2003; 2003: 26-29.

[42] Soomro K, Zamir AR. Action recognition in realistic sports videos. In: Computer Vision in Sports. Springer; 2014. pp. 181-208.

[43] Liu J, Luo J, Shah M. Recognizing realistic actions from videos in the wild. 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, USA; 2009. pp. 1996-2003.