Two-Stream Dictionary Learning Architecture for Action Recognition

Ke Xu, Xinghao Jiang, *Member, IEEE*, and Tanfeng Sun, *Member, IEEE*

***Abstract*— In this paper, a novel method based on the two-stream dictionary learning architecture for human action recognition is proposed. The architecture consists of interest patch (IP) detector and descriptor, two-stream dictionary models, and support vector machine (SVM) for classification. The novel IP detector combines a human detector and a contour detector to extract patches of interest on human contours. Then the IP descriptors are calculated in spatial stream and temporal stream separately. In each stream, a dictionary is trained for each action with the IP descriptors as an action model. In this way, measuring the similarity between an action sequence and an action model is transformed to reconstructing the IPs in this sequence with the model and computing the reconstruction error. For each action, an IP distribution histogram is constructed and the histogram is further used to train an SVM classifier in each stream. A score fusion method is applied to fuse the spatial and temporal SVM classification results to make a final decision. The proposed architecture is examined on four public data sets with different background complexities and camera motion conditions: Weizmann data set, KTH data set, Olympic sports data set, and HMDB51 data set. The results are further compared with state-of-the-art approaches in the experiment section to confirm the effectiveness of this architecture.**

***Index Terms*— Action recognition, dictionary learning, two streams.**

I. Introduction

ACTION recognition has been an attractive research topic for a few decades. It can be used in various applications such as abnormal event detection, autodriving system, and human–computer interaction. Due to the increasing number of surveillance cameras and the personal digital cameras, the demands for video content analysis and video analysis on big data are urgent in recent years.

Action recognition is one of the most challenging tasks in computer vision because of the diversity of video scenes and actions. According to the properties of video scenes, videos can be divided into three categories, including surveillance videos, hand-held camera videos, and recordings. A huge amount of videos are taken by surveillance cameras. Although surveillance videos are often captured by fixed cameras, it is still difficult to detect human body in a cluttered background or when occlusion happens. Therefore, researchers have proposed some methods to resist the background affection and occlusion. In traditional methods, local point detectors and descriptors are used as effective methods to represent actions. For example, Laptev *et al.* [1] proposed spatial temporal interest point (STIP) detector and applied histograms of gradient (HOGs) and histograms of flow descriptors for recognition. Bay *et al.* [2] proposed SURF detector and descriptor which perform well while being calculated in a relatively efficient way. These local point detectors and descriptors are robust to scale variance, optical variance, and occlusion. However, these local point detectors ignore the point locations information and point connections, so it may lead to some false recognitions. The performance of these detectors is also easy to be influenced by shadow, hair, and clothes texture. Therefore, human detectors and background subtraction methods should be considered with local point detectors together.

Tremendous amount of videos are also shot by hand-held cameras. Thousands of hand-held camera videos are uploaded to video websites every day. These videos have a characteristic of intense camera moving and cluttered background. The difficulty of action recognition in this kind of videos is that people conduct the same action with different gestures and viewpoints, which causes the intra-class diversity. The intra-class diversity sometimes confuses the classifiers and brings challenges to recognition.

To achieve a satisfactory recognition performance on this type of videos, descriptors have to be deeply researched to generalize the representation of actions. An appropriate classifier is also important when classifying actions with these descriptors. For example, Brun *et al.* [3] encoded action sequences as a string and used a string kernel framework for recognition. Bettadapura *et al.* [4] explored an augmenting bag of words (BOW) to discover temporal and structural information for activity recognition. Shukla *et al.* [5] used a temporal BOW model to capture the temporal domain feature. In these methods, the temporal information in the action sequences is well considered and the methods have good performances in recognition with classifiers such as support vector machine (SVM) and k-NN.

The other types of videos are recordings such as movies or sports recordings. Some actions in the movies are difficult to be recognized because of the people-object interactions. Sports recordings happen in particular sports scenes and the scenes information is helpful to recognition. Researches have proposed some approaches such as computing trajec-

Manuscript received December 16, 2015; revised March 31, 2016; accepted July 7, 2016. Date of publication February 7, 2017; date of current version March 3, 2017. This work was supported by the Nature Science Foundation of China under Grant 61572321, Grant 61272439, Grant 61572320, Grant 61272249, and Grant 61271319. This paper was recommended by Associate Editor T. Bouwmans. *(Corresponding author: Xinghao Jiang.)*

The authors are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. (e-mail: l13025816@sjtu.edu.cn; xhjiang@sjtu.edu.cn; tfsun@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2017.2665359

tories and tracking in dealing with this type of videos. Wang and Schmid [6] used dense trajectories along with local descriptors to track person movement in continuous 15 frames. Yu and Yuan [7] generated action proposals and formed action paths. These methods have the ability to model long duration actions and performed well in real life video data sets like HMDB51 [8]. But these methods have high computational cost and time cost due to the high resolution of frame and the pixel level tracking strategy.

The contributions provided by this paper consist of three parts: the novel interest patch (IP) detector and descriptor, the action representation method based on two-stream dictionaries, and the extensive experiments on four public data sets that demonstrate the effectiveness of our method. In [9], we proposed the idea of selecting points on human contours. However, the method in [9] is not conducted on videos with camera motion and cluttered background. And the computational cost and time cost of the method are not evaluated. Thus, in this paper, we extend the idea to a robust IP detector and use more representative descriptors to achieve a better recognition performance.

The IP detector is robust to camera motion and cluttered background. So it can provide stable patches and improve the performance of STIP detector. The IP descriptors are computed in both spatial stream and temporal stream. Then the two-stream dictionary learning architecture is proposed to train dictionaries with IP descriptors as action models. In this way, measuring the similarity between an action sequence and an action model is transformed to reconstructing the IP descriptors with the model and computing the reconstruction error. The IP distribution histograms are constructed by evaluating the least reconstruction errors and are used to train SVM classifiers. The spatial SVM probabilities for each class are denoted as spatial scores (SS), and the temporal SVM probabilities for each class are denoted as temporal scores (TS). A score fusion (SF) method is then applied on SS and TS to make a final recognition decision. This two-stream dictionary learning architecture is a robust and fast method for action representation and recognition. The experiments are conducted on four public data sets and the effectiveness of this method is proved with higher accuracy and less computational cost.

The rest of this paper is organized as follows. Section II elaborates the related works and the ideas that inspire our architecture. Section III explains the IP detector and descriptor proposed in this paper. Section IV presents the framework of our algorithm, including the spatial temporal dictionaries training, the IP distribution histograms construction, the spatial and temporal SVMs training, and the SF method. Section V presents the experimental analysis and shows the experimental results in comparison with some state-of-the-art approaches. Section VI draws the conclusion.

II. Related Work

Classification structure based on local point detectors for action recognition is popular in recent years. This structure contains local points extraction and description, bag of visual words building, feature distribution histogram calculation, and SVM classifiers training. It is proved that local point detectors and descriptors have good resistance in optical changing, occlusion, scale variance, and rotation. Researchers have proposed many efficient and outstanding local detectors and descriptors. Mikolajczyk and Schmid [10] proposed a robust scale and affine invariant interest point detector. Dollár *et al.* [11] introduced spatiotemporal cube features descriptor. Ren and Ramanan [12] reported histograms of sparse codes which combines the advantages of HOG [13] and sparse coding. Scovanner *et al.* [14] used 3D-SIFT to extend 2D-SIFT descriptor from static images to video sequences. Klaser *et al.* [15] proposed a detector called HOG3D based on orientation histograms of 3D gradient orientations. Dalal *et al.* [16] used motion boundary histograms (MBHs) to compute derivatives of optical flow and achieved robustness to camera motion. Yeffet and Wolf [17] proposed local ternary patterns as an extension of classic local binary patterns feature.

These local descriptors capture both the spatial and temporal statistical features around local points. These methods have many advantages, such as the low feature dimension and the low computational cost. For example, 3D-SIFT has 128 dimension and MBH has only 96 dimension, so the model training phase and the classification phase have low computational cost and low time cost. However, these methods also have some disadvantages. For example, the statistical features ignore the point locations, which results in the missing of point connection information. Inspired by these methods, in this paper, all the pixels in an IP are described and the descriptions are concatenated. The IPs descriptor consist of pixel values and gradients in spatial domain, as well as the optical values and optical gradients in temporal domain. In this way, the spatial temporal features and the point connection information are both reserved.

Another disadvantage of local point descriptors is the utilization of short temporal sequence window, which can only capture motions in a few frames. So researchers have explored methods using longer frame sequences for better recognition performance. For example, trajectories are used by researchers to track the action movement. Wang *et al.* [18] have used dense trajectories together with local descriptors and achieved outstanding performance on variety of data sets. Jiang *et al.* [19] used local patch trajectories to model the motion relationships discarded by STIP detector. Brun *et al.* [3] encoded long frame sequences into aclets sequences to model global temporal information. Fernando *et al.* [20] captured video-wide temporal information as representation and used ranking machine to train models. Zhu *et al.* [21] proposed a two-layer structure for action recognition to automatically exploit a mid-level action representation.

These methods encode long action sequences to represent actions and achieved good recognition accuracies. However, tracking every points in a frame takes much time. So inspired by these ideas, spatial sequence window and the temporal sequence window are used along with IP detector in this paper. The window size is set to five frames without overlapping in order to catch continuous variations in both spatial domain and temporal domain.

After actions are represented by descriptors, a BOW model is often constructed with K-means cluster method in traditional framework. In recent years, some more complex BOW models are researched, such as temporal BOW proposed by Shukla *et al.* [5] and augmenting BOW proposed by Bettadapura *et al.* [4]. Besides, BOWs can be optionally replaced by some high dimension encoding methods like Fisher vector or some machine learning encoding methods like sparse coding, ranking machine encoding, and dictionary learning architecture. These methods are also reported to achieve good performances for action recognition. Qiu *et al.* [22] reported a sparse dictionary-based representation for action. Liu *et al.* [23] proposed a Hessian regularized sparse coding method for action recognition. Lu *et al.* [24] proposed the idea of slicing frame to patches in different scales and using patches to train dictionaries. Fanello *et al.* [25] introduced a real-time action recognition method with dictionary learning. The work of Fanello *et al.* [25] inspires the two-stream dictionary learning architecture in this paper, and their real time action recognition achievement has proven that the recognition time cost of dictionary learning method can be controlled in a reasonable scale.

The idea of using Fisher vector for classification is proposed in [26]. And researchers have applied the Fisher vector on action recognition. Jain *et al.* [27] used trajectories and VLAD coding technique and outperformed the reported result on challenging data sets. Wang *et al.* [18] explored Fisher vector as an alternative feature encoding approach to BOW histograms in action recognition. Experiments in [18] show that Fisher vector performs better to BOW in some extent. Peng *et al.* [28] proposed stacked Fisher vectors (SFV) and reported better results than one layer Fisher coding in action recognition. These high-dimension encoding methods bring a raise in recognition accuracy, but also bring more computational cost and time cost.

Deep learning method like convolutional neural network (CNN) has been widely used in object recognition and image classification. So researchers bring deep learning networks to action recognition area and achieve outstanding performance. Simonyan and Zisserman [29] used a two-stream convnet network, their separate using of spatial stream and temporal stream network inspired our two-stream architecture. Ji *et al.* [30] proposed a deep learning network for action recognition using 3D CNNs. Baccouche *et al.* [31] used sequential deep learning for human action recognition. Charalampous *et al.* [32] proposed an online deep learning method for action recognition. Xie *et al.* [33] introduced a pyramidal deep learning architecture for human action recognition. Deep learning networks have reported state-of-the-art recognition results on many action data sets. The architecture of dictionary learning stream proposed in this paper borrows the idea of deep learning, which is splitting the frame into basis units and reconstructing the frame for classification.

However, the methods in [29] and [30] also have some disadvantages. For example, the methods need many training samples, the model training phase takes much time, and the whole frame is taken as an input so the recognition results may be affected by background. The temporal window size is also limited to 2 in [29]. The architecture in this paper achieved a higher accuracy than Simonyan's work since more temporal information is used. Our method also spends less time in model training phase in contrast with Simonyan's and Ji's networks.

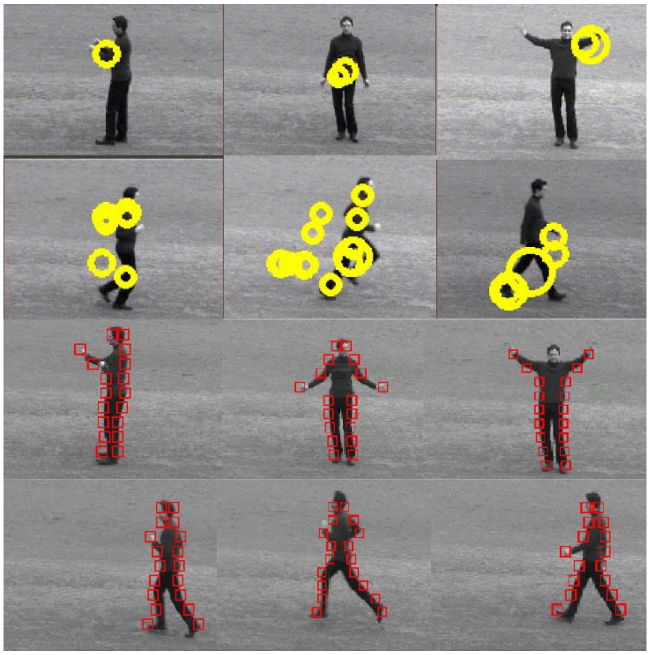


Fig. 1. Samples from KTH data set. Harris3D STIPs [1] are marked by circles. IPs proposed in this paper are marked by rectangles.

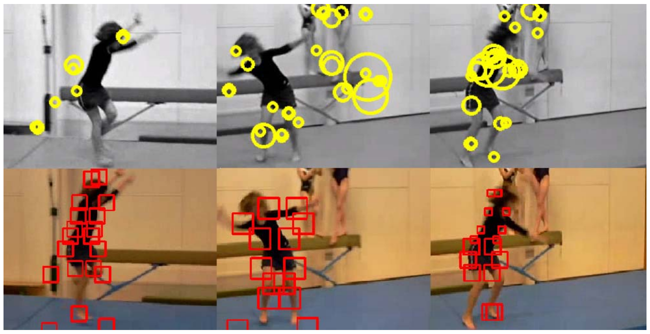


Fig. 2. Samples from HMDB51 data set. Harris3D STIPs [1] are marked by circles. IPs proposed in this paper are marked by rectangles.

III. Novel Interest Patch Detector and Descriptor

In recent years, local point detectors such as the STIP detector are widely used for action recognition. Traditional local point detectors find points which have high responses in both spatial region and temporal volumes. However, these detectors can be affected by background, shadow, clothes texture, hair, face, or camera motion. Meanwhile, the number of detected points for each frame is not stable (see the first and second rows in Fig. 1). Thus, it may cause a false recognition. At the same time, in some real world video samples, the detected points are always located on background due to the camera motion (see first row in Fig. 2). Therefore, a novel IP detector is proposed in this paper to improve the performance of traditional local point detectors.

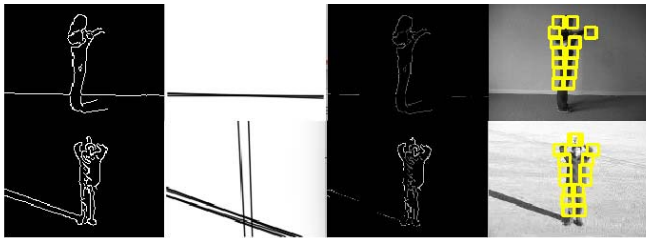


Fig. 3. Using Hough transform to eliminate shadow and background affection. Left to right: the first column is the original contour, second column is the lines detected, third column is contour after line subtraction, and fourth column is the detected IPs.

Inspired by the idea of [1] and [2], in frame t, HOG human detector [13] and ViBe background subtraction method [34] are applied to locate people with a body region. The background subtraction algorithm helps us to locate people more quickly because the human detector window size and sliding step can be estimated by the foreground area. Then canny contour detector is used in the body region and the contour C_t is stored. Hough transform is used to eliminate long horizontal lines and vertical lines in the frame caused by shadow and background (Fig. 3). Since the human contours are constructed by short lines, the long horizontal and vertical lines can be considered as background lines. In Fig. 3, the shadow and background lines are detected and eliminated. Although some human contours are subtracted too, the edges of human body are reserved and the IPs are not affected due to the level dividing strategy as follows.

The body region with height H_R is horizontally divided into N_L levels. For each level i from 1 to N_L, $l_i = H_R \times i/N_L$ refers to the lower bound of ith level. l_i is stored in vector $L = \{l_1, \ldots, l_{N_L}\}$ with $l_i < l_{i+1}$. As proposed in (1) and (2), in each level i, each point $P(x, y)$ in C_t is accessed with y from l_i to l_{i+1}. The points with minimum x and maximum x are selected as IP centers. Two IP centers are extracted in each level and the total IP number for each frame is $2 \times N_L$. Vector IP stores the IP centers ordinates

$$\text{IP[2i-1]} = \arg\min_x \{P(x, y) | P \in C_t, y \in (l_i, l_{i+1})\} \quad (1)$$

$$\text{IP[2i]} = \arg\min_x \{P(x, y) | P \in C_t, y \in (l_i, l_{i+1})\}. \quad (2)$$

Then a rectangle patch around an IP center is selected as an IP. The scale of the patches is decided by the area of body region. Three scales of IP are used in this paper, which are the $1/20$, $1/30$, and $1/50$ of body region area. With the three scales, IPs can capture the human contour information from global to local and IPs are all resized to 12×12 when training dictionaries. In spatial descriptor, for each pixel in an IP, pixel value $P(x, y)$ and the gradients in x-, y-, and t-directions P_x, P_y, and P_t are used as the spatial feature. Let k_n denote the number of pixels in an IP, the feature set is denoted by $F_s = \{f_1, \ldots, f_{k_n}\}$ and $\forall i \in [1, k_n]$, $f_i = \{P(x, y), P_x, P_y, P_t\}$. In temporal descriptor, the Farneback optical flow for frames is computed first. Then with exactly the same IP centers as spatial stream, the temporal description for each pixel includes optical flow values (v_x, v_y) and the gradients of optical flow in x, y, and t dimension $(v_{xx}, v_{yy}, v_{xt}, v_{yt})$. So the temporal feature $F_t = \{f_1, \ldots, f_{k_n}\}$ where $\forall i \in [1, k_n]$, $f_i = \{v_x, v_y, v_{xx}, v_{yy}, v_{xt}, v_{yt}\}$. Both the spatial sequence window length and the temporal sequence window length in this paper are set to five frames without overlapping in order to catch continuous changes in spatial and temporal domains.

In Figs. 1 and 2, Harris3D STIPs [1] are marked by circles. The IPs proposed in this paper are marked by rectangles. It is shown that the number of Harris3D STIPs in each frame is not stable and some points are located on background or on clothes, while our method has stable points number in each frame and most points located on the edge of action. Besides, the Harris3D STIP detector computes points responses on the whole video to find the global response threshold while IP detector processes one frame each time. So IP detector spends less time than STIP detector.

The IP descriptor is efficient in spatial representation because it reserves the human contour information. It is also efficient in temporal representation since it reserves the body edge motions which reflect the characteristics of an action. The IPs can also be described with tradition local point descriptors. In Section V, some traditional detectors and descriptors are compared with our IP detector and IP descriptor in recognition accuracy.

IV. Two-Stream Dictionary Architecture

A. Dictionary Learning

In this section, the two-stream dictionary learning architecture is built for training and classification. An overall pipeline is shown in Fig. 4. In training phase, first, the IPs are detected and described. Then the spatial dictionaries and temporal dictionaries are trained with the IP descriptors separately. For each action sample, the IP descriptors are reconstructed by all the dictionaries and each IP is assigned to a dictionary by evaluating the least reconstruction error. An IP distribution histogram is then constructed. The IP distribution histograms are taken as the samples for SVM training. The spatial SVM classifier and temporal SVM classifier are also trained separately.

In testing phase, the IPs are extracted and described on spatial and temporal streams. The IP distribution histograms of test sample are constructed using the trained dictionaries. The IP distribution histograms are then classified with spatial SVM and temporal SVM to generate SS and TS for each action. Finally, the SF method is applied to fuse SS and TS to make the final recognition decision. The architecture details are demonstrated below.

Suppose a data set containing N actions, each action contains n video samples and each video sample contains k IPs. For each action in spatial domain, an IP descriptor is denoted by s_j, $j \in [1, n \times k]$. The action descriptor matrix is $S = [s_1, \ldots, s_{n\times k}]$. Let p denote the dictionary centers number and q denotes the descriptor length. A descriptor dictionary $D_i \in \mathbb{R}^{p\times q}, i \in [1, N]$ is learned by the following equation, which is solved by K-SVD algorithm [35]:

$$\min_{D,X} \|S - D_i X\|_2^2 \quad \text{s.t. } \|x_i\|_0 \le T_0 \quad (3)$$

where $X = [x_1, \ldots, x_i, \ldots, x_{n\times k}]$ is the sparse coefficients matrix, x_i is a coefficients column vector, and T_0 is the sparse

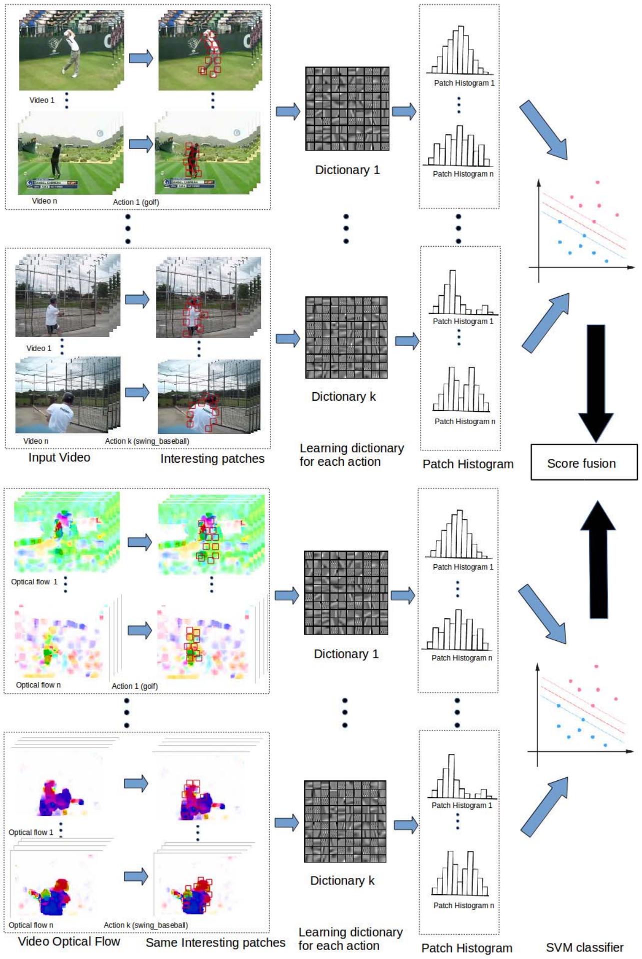


Fig. 4. Two-stream architecture for action recognition. IPs are selected and described on spatial and temporal streams. Then dictionaries for each action are trained. IPs for each action are reconstructed by all dictionaries to construct IP distribution histograms. Spatial and temporal SVM classifiers are trained separately with the histograms. Finally, a fusion method is applied on the SS and TS to make a final decision for classification.

constrain. N dictionaries are generated after training. For a video sample containing k IPs with the descriptor matrix $S' = [s'_1, \ldots, s'_k]$, s'_t denotes an IP descriptor, $t \in [1, k]$. Reconstructing an IP with dictionaries $D = \{D_1, \ldots, D_N\}$ is expressed as

$$\mathrm{Err}(t, i) = \min_{\beta} \left\| s'_t - D_i \beta_i \right\|_2^2 \quad \text{s.t. } \|\beta_i\|_0 \le T_0 \tag{4}$$

where $i \in [1, N]$ and $\mathrm{Err}(t, i)$ denotes the reconstruction error of the descriptor s'_t using the ith dictionary. $\beta \in \mathbb{R}^{q \times 1}$ contains sparse coefficients. $\|s'_t - D\beta\|_2^2$ is the data fitting term and $\|\beta\|_0$ is the sparsity regularization term. T_0 is the upper bound parameter to control sparsity. The equation is solved by OMP algorithm [36].

The temporal dictionaries $D = \{D_1, \ldots, D_N\}$ are trained with the same method as spatial dictionaries using (3). The reconstruction errors Err for each patch are also computed as (4).

The size of a dictionary is controlled by dictionary center number K and IP descriptor length. Similar to traditional BOW with K-means cluster, as K increases, the recognition accuracy rises and becomes stable. A K-Accuracy table is shown in Section V to reveal this relationship.

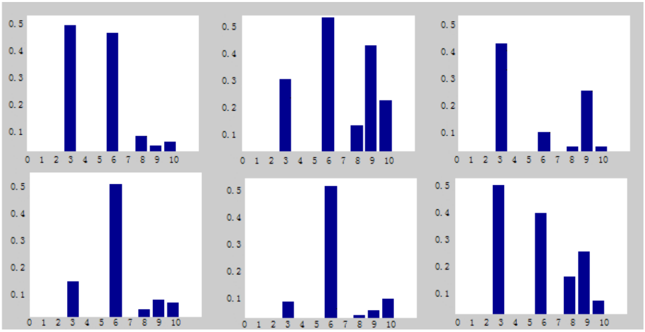


Fig. 5. Two similar actions IPs histograms in Weizmann data set. The first row is from "skipping" and the second row is from "running." The use of SVM can achieve a better result in classification than mean of Err.

B. IP Distribution Histogram

In this section, the IP distribution histogram is introduced as a representation for actions. The histograms are used to train SVM classifiers.

After spatial dictionaries and temporal dictionaries have been trained for each class, in each stream and in each sample, each IP is reconstructed by all the dictionaries and is assigned to a dictionary by evaluating the least reconstruction errors.

Suppose a data set with N actions and n samples, for each action, $H = \{h_1, \ldots, h_n\}$ denotes the histograms set. $\forall m \in [1, n]$, $h_m = \{g_1, \ldots g_N\}$ denotes a sample histogram. $\forall i \in [1, N]$, g_i is the patch number assigned to the ith dictionary. Suppose a video sample contains k IPs. As expressed in (5), for the tth IP where $t \in [1, k]$, a dictionary D_i is assigned to minimize the $Err(t, i)$, $i \in [1, N]$. The corresponding histogram $h_m(i)$ will plus one as expressed in (6) where m denotes the mth video. After each h_m in H are constructed, the histograms are normalized to demonstrate the frequency and distribution of IPs. An IP distribution histogram is used to represent a video sample because it reflects the characteristic of the action and reduces the affect of outliers

$$i = \arg\min_{i \in [1,N]} \ \mathrm{Err}(t, i), t \in [1, k] \tag{5}$$

$$h_m(i) = h_m(i) + 1. \tag{6}$$

C. SVMs Training and Score fusion

After IP distribution histograms are constructed, spatial, and temporal SVM classifiers are trained for recognition separately. The IP distribution histograms of each action are used as training samples. And an N-classes SVM structure is applied with χ^2 kernel. Then for a test video, IPs are extracted and described, and the IP distribution histograms for spatial and temporal streams are constructed with the trained dictionaries. The spatial histograms are classified with the spatial SVM and the SS is computed. In temporal stream, temporal IP distribution histograms are calculated and the TS is computed. As in Fig. 5, the histograms for "skipping" and "running" are difficult to be recognized because the false assigned IPs. The histograms can be well classified by a trained SVM classifier.

The sum of SS and TS is taken as the final score and the action corresponding to the highest score is selected as the

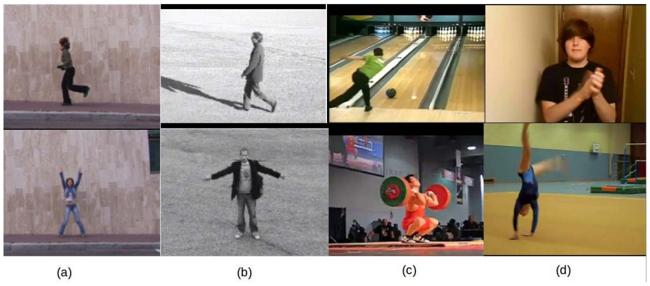


Fig. 6. (a) From Weizmann data set. (b) From KTH data set. (c) From Olympic sports data set. (d) From HMDB51 data set.

recognition result. In this paper, the SF method is simply made by summing the SSs and TSs because it is a simple and fast method to combine the two-stream scores. The principle is that a well-recognized action has a very high score in the corresponding class while a confusing action has almost the same scores in all classes. Therefore, the SF ensures that if an action is well-recognized in one stream, the result of the other stream will not affect the recognition performance.

V. Experiments and Analysis

A. Data Sets

In this section, four data sets with different scenes and camera movements are employed. Including Weizmann action data set, KTH data set, Olympic sports data set, and HMDB51 data set.

Weizmann data set contains 93 videos of nine people performing the following ten actions: running, walking, skipping, jumping-jacks, jumping forward on two legs, jumping in place on two legs, jumping sideways, waving with two hands, and waving with one hand. Each clip lasts about 2 s at 25 Hz with frame size of 180 × 144 pixels. It contains only one person in one frame and has no camera motion [see Fig. 6(a)].

The KTH human action data set is also evaluated in this paper. It contain six types of human actions (walking, jogging, running, boxing, hand-waving, and hand-clapping) performed by 25 actors with four different scenarios: S1 (outdoors), S2 (outdoors with scale variation), S3 (outdoors with different clothes), and S4 (indoors). There are totally 599 video clips with image frame size of 160 × 120 pixels. It contains one person in each frame. A few samples have camera motion of zooming in and zooming out. The data set contains optical variations and shadows on the background [see Fig. 6(b)].

The Olympic sports data set consists of athletes practising different sports. The samples are collected from YouTube and are annotated using Amazon Mechanical Turk. There are 16 sports actions (such as high-jump, pole-vault, basketball lay-up, and discus), represented by a total of 783 video sequences. These actions are taken in specific sports field [see Fig. 6(c)]. In the data set, 649 sequences are used for training and 134 sequences are used for testing as recommended by the authors.

The HMDB51 data set is collected from a variety of sources ranging from digitized movies to YouTube videos [see Fig. 6(d)]. In total, there are 51 action categories and 6766 video sequences. Using three train-test splits, there are 153 split files provide by the authors. For every class and split, 70 videos are used for training and 30 videos are used for testing.

TABLE I
Different Combinations in Our Architecture

Method	Weizmann	KTH
STIP + MRE	82.8%	81.7%
Dense + MRE	81.1%	84.2%
IP + MRE	91.2%	88.4%
IP + SDH	96.2%	93.7%
IP + TDH	97.9%	94.2%
IP + SF	**99.1%**	**95.8%**

B. Overall Experiment Settings

In this section, the overall experiment settings are explained. In the IP detector, the human region is divided into ten levels and the IPs are detected in three scales. The scales includes $1/20$, $1/30$, and $1/50$ of human region area. So $20 \times 3 = 60$ patches are detected on each frame. The patches are all resized to 12×12, and the sequence window is set to five frames without overlapping as described in Section III. The spatial descriptor length for each pixel is 4, which makes a spatial IP descriptor equals to $12 \times 12 \times 5 \times 4 = 2880$ dimension. The temporal descriptor length for each pixel is 6, which make a temporal IP descriptor $12 \times 12 \times 5 \times 6 = 4320$. Spatial and temporal dictionaries for each action are trained with dictionary size 256 and the sparsity upper bound is set to 10.

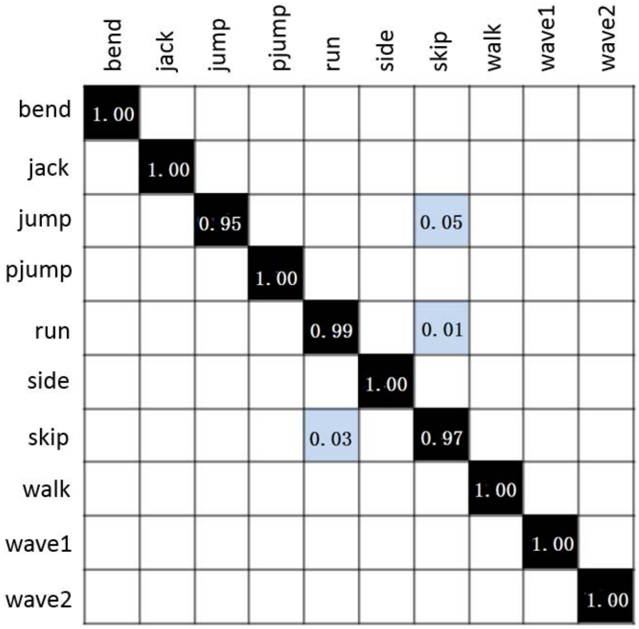
For classification, an N-class SVM classifier is used with a χ^2 kernel. The parameters of SVMs are $C = 50$ and $g = 0.003$.

The experiments are all conducted on a laptop with an Intel i5-4570 CPU and an 8-GB memory.

C. Experiments on Weizmann and KTH

In this section, the recognition results and the analyses of these results are given. In Weizmann data set, following the leave-one-out split method, 92 videos are used as training data and 1 video is used as test data for each time. So each video is tested once with the other 92 videos as training samples. The training and testing phases are repeated ten times to compute a mean accuracy and generate a confusion matrix. The mean accuracy of Weizmann data set is 99.1%, which is shown in Fig. 7. Confusions exist among "jump," "skip," and "run" due to the similarities in gesture and moving. So the result may be improved by evaluating the width of legs when people moving.

In the experiments, STIP detector and descriptor [1] are used with the mean reconstruction error (MRE) as a baseline for comparison. When computing MRE, the IPs of an action are constructed by all the dictionaries. So each dictionary corresponds to an IP reconstruction error set. By evaluating the mean of this set, an MRE is counted. Therefore, the dictionary corresponding to the least MRE is picked as the recognition result. As expressed in Table I, in the experiments, STIP detector, densely sampled point detector, and IP detector



	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.00									
jack		1.00								
jump			0.95				0.05			
pjump				1.00						
run					0.99		0.01			
side						1.00				
skip					0.03		0.97			
walk								1.00		
wave1									1.00	
wave2										1.00

Fig. 7. Weizmann data set confusion matrix, the reported mean average precision is 99.1%.

TABLE II
COMPARISON RESULTS

Method	Weizmann	KTH
Niebles et al.[38]	-	91.30%
Zhang et al.[39]	92.89%	91.33%
Lu et al.[40]	93.50%	91.50%
Bregonzio et al.[41]	96.66%	93.17%
Ballas et al.[42]	-	94.60%
Kim et al. [43]	-	95.00%
Gorelick et al.[37]	97.54%	-
Our method(IPs + FS)	**99.10%**	**95.80%**

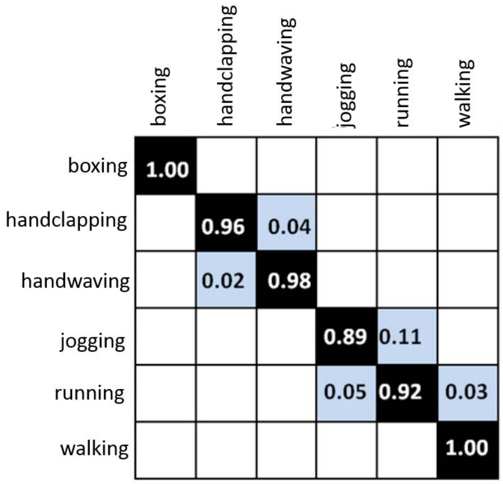
are compared in the first three rows with MRE. The IP detector achieves the best performance among the three methods. This proves that the IP detector can improve the performances of STIP detector and the dense sampled point detector. Meanwhile, the use of spatial IP distribution histogram (SDH) and temporal IP distribution histogram (TDH) have outperformed the results only using MRE. And the result becomes better with a spatial temporal SVM SF method. Table I shows the effectiveness of our IP detector, two-stream dictionaries learning method, and SF method. The method proposed in this paper is compared with other methods in Table II.

In Table II, Gorelick *et al.* [37] used space-time volumes and reported an accuracy of 97.54%. The confusions are mainly between "skip" and "jump." Our performance achieves a better result in accuracy mainly because the IPs are extracted on human body and described in both spatial and temporal streams. So the differences between skip and jump on leg contours and leg motions are well recognized.

The relationship between dictionary centers number K and the accuracy is also evaluated on data sets. A K-Accuracy table is shown in Table III as a sensitivity experiment of K. It is shown that as K increases, the accuracy rises and becomes stable. It is because a large value of K makes every dictionary containing enough basis units. So reconstructing any IP in the data sets will return a small reconstruction error and the IPs may be false assigned. A very large value of K also brings more computational cost. So in this paper, K is set to 256 for balancing the computational cost and performance.

TABLE III
K-ACCURACY RELATIONSHIP ON WEIZMANN AND KTH

K	100	150	200	250	300	500	1000
Weizmann	0.957	0.966	0.982	0.991	0.991	0.990	0.988
KTH	0.930	0.942	0.951	0.958	0.958	0.958	0.951



	boxing	handclapping	handwaving	jogging	running	walking
boxing	1.00					
handclapping		0.96	0.04			
handwaving		0.02	0.98			
jogging				0.89	0.11	
running				0.05	0.92	0.03
walking						1.00

Fig. 8. KTH data set confusion matrix, the reported accuracy is 95.8%.

In KTH data set, using leave-one-out split method, 24 actors are used as training data and 1 actor is used as test data for each time. The leave-one-out training and testing phases are repeated ten times to compute a confusion matrix. The accuracy of our method on KTH data set is 95.8% in Fig. 8.

It is shown that confusions occurred between "clapping" and "waving." Because of the similarity in gestures, the spatial classifier is confused by these two actions. The confusions between "jogging" and "running" are caused by viewpoint variation. Different viewpoints may cause different motion features. So some "running" videos maybe similar to "jogging" in temporal stream. The comparisons of our recognition performance and accuracies of other methods are also shown in Table II. Ballas *et al.* [42] used saliency to locate interest region and applied a content driven pooling method to model actions. The accuracy of this method is 94.6%. Kim *et al.* [43] used a tensor canonical correlation analysis to find joint space-time features and reported an accuracy of 95%. The accuracy of our architecture is 95.8%, which achieves the best recognition performance among the prior works.

D. Experiments on Olympic Sports

In Olympic sports data set, 649 sequences are used for training and 134 sequences are used for testing as recommended by the authors. The recognition accuracy for each action is shown in Table V. Low accuracies on "diving plat from 10 m," "high jump," and "triple jump" are observed.

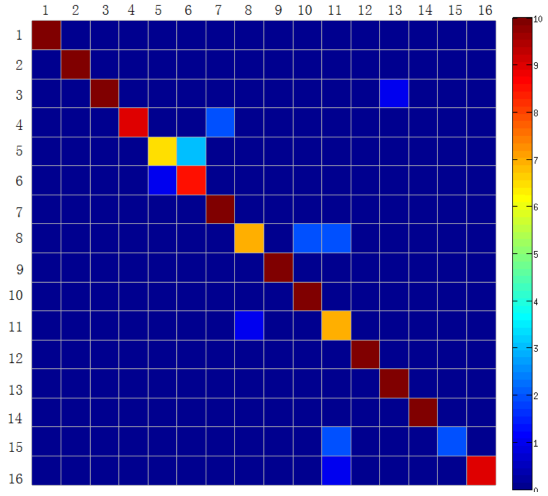


Fig. 9. Olympic sports data set confusion matrix, the reported mean average precision is 88.81%.

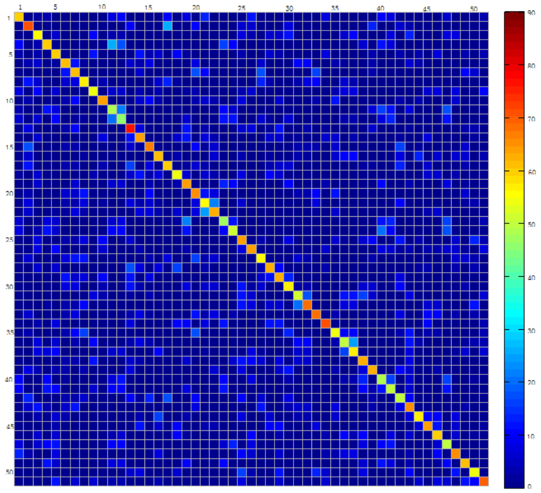


Fig. 10. HMDB51 data set confusion matrix, the reported mean average precision is 59.47%.

TABLE IV

COMPARISON ON OLYMPIC SPORTS

Method	accuracy
Jain et al.[27]	83.20%
Li et al.[44]	84.50%
Wang et al.[6]	84.90%
Gaidon et al.[45]	85.00%
Wang et al.[18]	90.40%
Our work.	**88.81%**

TABLE V

OLYMPIC SPORTS ACCURACY FOR EACH ACTION

Action	Accuracy	Action	Accuracy
1. basketball layup	100.00%	9. javelin throw	100.00%
2. bowling	100.00%	10. long jump	100.00%
3. clean and jerk	100.00%	11. pole vault	87.50%
4. discus throw	81.80%	12. shot put	100.00%
5. diving plat from 10m	66.67%	13. snatch	100.00%
6. diving spring board 3m	87.50%	14. tennis serve	100.00%
7. hammer throw	100.00%	15. triple jump	50.00%
8. high jump	63.60%	16. vault	90.00%

“Diving plat from 10 m” is confused with “diving spring board 3 m” because of the similar human gesture and movement. The “triple jump” and “high jump” are often confused with “long jump” due to the similar behavior of athletes. The confusion matrix is shown in Fig. 9, and the final reported recognition accuracy is 88.81%.

The comparison results of Olympic sports data set are shown in Table IV. Our architecture is compared with some recent accomplishments. Jain *et al.* [27] used trajectories and VLAD coding technique to achieve an accuracy of 83.2%. Li *et l.* [44] used dynamic pooling method for event detection and reported an accuracy of 84.5%. Gaidon *et al.* [45] learned hierarchical representations of activity videos in an unsupervised manner and the accuracy is 85.0%. Wang *et al.* [18] reported a baseline accuracy of 84.9% and the best accuracy in their paper is 90.4%. The proposed accuracy in our paper is 88.81% and it is closed to the performance in [18].

In Table IV, Wang *et al.* [18] achieved the accuracy of 90.4% by using human detector and densely sampled feature points on spatial temporal pyramid (STP). The descriptor length for each point is 426 and it reduced to 64 by principal component analysis. The Gaussian number K for Fisher coding is 256, which make the spatial Fisher vector (SFV) to $2 \times 64 \times 256 = 32\,768$ dimension. It takes about 25 ms to encode the IP descriptors in an action with Fisher coding on our computer. In our architecture, the descriptor length is 7200 dimension and the descriptors are directly used to train dictionaries and to generate IP distribution histograms. Meanwhile, on our computer, reconstructing IP descriptors of an action takes only about 10 ms. So the computational cost and time cost are less than Wang’s coding method.

E. Experiments on HMDB51

In HMDB 51 data set, following the original protocol, three train-test splits are used. There are 153 split files provided by the authors. For each class, there are 70 videos for training and 30 videos for testing in each split file. The confusion matrix is given in Fig. 10. The accuracy of each action is reported in Table VII and the comparisons with other methods are given in Table VI. From Fig. 10, it is reported that some actions are confused because the body takes up most part of the frame such as “eating,” “drinking,” “smoking,” and “talking.” In these actions, human detector cannot cover the whole body of people so the movements of body parts like hands or arms are not detected. Besides, the facial expressions like “smile” and “laugh” are difficult to be represented and the actions with similar movement like “cartwheel” and “somersault” also bring challenges to our recognition architecture. It can be inferred that our work perform better than some local description works from Table VI. Jiang *et al.* [19] used local patch trajectories to model motion relationships discarded by STIPs and reported an accuracy of 40.7%. Ballas *et al.* [42] used saliency of image and structural learning method to improve the recognition performance and the accuracy is

TABLE VI

COMPARISON ON HMDB51

Method	accuracy
Jiang et al.[19]	40.7%
Ballas et al.[42]	51.8%
Jain et al.[27]	52.1%
Zhu et al.[21]	54.0%
Simonyan et al.[29]	59.4%
Wang et al.[18] baseline	55.9%
Wang et al.[18] with HD	60.1%
Our work.	**59.5%**

TABLE VII

HMDB51 ACCURACY FOR EACH ACTION

Action	Accuracy	Action	Accuracy
1. brush hair	60.00%	27. pullup	55.56%
2. cartwheel	67.78%	28. punch	63.33%
3. catch	54.44%	29. push	64.44%
4. chew	56.67%	30. pushup	57.78%
5. clap	60.00%	31. ride bike	52.22%
6. climb	63.33%	32. ride horse	62.22%
7. climb stairs	62.22%	33. run	68.89%
8. dive	57.78%	34. shake hands	70.00%
9. draw sword	55.56%	35. shoot ball	54.44%
10. dribble	64.44%	36. shoot bow	51.11%
11. drink	48.89%	37. shoot gun	58.89%
12. eat	44.44%	38. sit	63.33%
13. fall floor	75.56%	39. situp	66.67%
14. fencing	63.33%	40. smile	48.89%
15. flic flac	65.56%	41. smoke	50.00%
16. golf	62.22%	42. somersault	51.11%
17. handstand	60.00%	43. stand	65.56%
18. hit	55.56%	44. swing baseball	58.89%
19. hug	64.44%	45. sword	64.44%
20. jump	65.56%	46. sword exercise	60.00%
21. kick	55.56%	47. talk	51.11%
22. kick ball	63.33%	48. throw	65.56%
23. kiss	48.89%	49. turn	61.11%
24. laugh	51.11%	50. walk	54.44%
25. pick	53.33%	51. wave	70.00%
26. pour	64.44%		

51.8%. Jain *et al.* [27] applied trajectories and VLAD coding technique and achieved an accuracy of 52.1%. Zhu *et al.* [21] proposed a two-layer structure for action recognition to automatically exploit a mid-level acton representation and the accuracy is 54.0%. Simonyan and Zisserman [29] used the two-stream convnets framework and reported an accuracy of 59.4%. Wang *et al.* [18] used improved trajectories and Fisher vector to achieve a baseline accuracy of 55.9%. They increase the accuracy to 60.1% by using human detector to remove the background trajectories.

The method in this paper reports an accuracy of 59.47% and it is closed to the performance by Simonyan and Zisserman [29] and Wang *et al.* [18]. The convnet used in [29] needs large training data and it takes a long time to train the convnet model. In [18], the use of STP and the SFV with centers number $K = 256$ make the Fisher coding features reach 32 768 dimension and it takes about 25 ms to encode IP descriptors of an action with Fisher coding on our computer. In our method, the length of descriptor is 7200 and it takes nearly 10 ms to reconstruct IP descriptors of an action on our computer. So the computational cost of this paper is less than Wang's encoding method.

VI. CONCLUSION

In this paper, a two-stream dictionary learning architecture is proposed for action recognition. The architecture contains the following parts. First, IPs detector based on human detector, background subtraction, and contour detector are used to extract IPs on human contours. Then the IP descriptors based on pixel values, gradients, and optical flow are computed in spatial and temporal streams. Dictionaries in the two streams are trained for each action and the IPs are assigned to the dictionaries based on the least reconstruction error. Then the IP distribution histograms are built. Spatial SVM and temporal SVM are trained based on the IP distribution histograms and the scores of these two SVMs are summed to make a final decision.

This architecture is evaluated on four widely accepted data sets: Weizmann data sets, KTH data sets, Olympic data sets, and HMDB51 data sets. These data sets are different in camera motion, background and resolution. The architecture achieves an accuracy of 99.1% on the Weizmann data set, 95.8% on the KTH data set, 88.81% on the Olympic sports data set, and 59.47% on the HMDB51 data set. The results show the effectiveness of our algorithm compared with the related methods.

REFERENCES

[1] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis. (ECCV)*, May 2006, pp. 404–417.

[3] L. Brun, G. Percannella, A. Saggese, and M. Vento, "Action recognition by using kernels on aclets sequences," *Comput. Vis. Image Understand.*, vol. 144, pp. 3–13, Mar. 2015.

[4] V. Bettadapura, G. Schindler, T. Ploetz, and I. Essa, "Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2619–2626.

[5] P. Shukla, K. K. Biswas, and P. K. Kalra, "Action recognition using temporal bag-of-words from depth maps," in *Proc. Citeseer*, 2013, pp. 41–44.

[6] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3551–3558.

[7] G. Yu and J. Yuan, "Fast action proposals for human action detection and search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1302–1311.

[8] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2556–2563.

[9] K. Xu, X. Jiang, and T. Sun, "Human activity recognition based on pose points selection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2834–2930.

[10] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[11] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. 2nd Joint IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking Surveill.*, Oct. 2005, pp. 65–72.

[12] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3246–3253.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, no. 1, pp. 886–893.

[14] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th ACM Int. Conf. Multimedia*, Sep. 2007, pp. 357–360.

[15] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proc. 19th Brit. Mach. Vis. Conf. (BMVC)*, 2008, pp. 1–275.

[16] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. 9th Eur. Conf. Comput. Vis. (ECCV)*,, May 2006, pp. 428–441.

[17] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 492–497.

[18] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 219–238, Sep. 2015.

[19] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo, "Trajectory-based modeling of human actions with motion reference points," in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2012, pp. 425–438.

[20] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, vol. 2. no. 7, pp. 5378–5387

[21] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu, "Action recognition with actons," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3559–3566.

[22] Q. Qiu, Z. Jiang, and R. Chellappa, "Sparse dictionary-based representation and recognition of action attributes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 707–714.

[23] W. Liu, Z. Wang, D. Tao, and J. Yu, "Hessian regularized sparse coding for human action recognition," in *Proc. 21st Int. Conf. MultiMedia Modeling*, Jan. 2015, pp. 502–511.

[24] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2720–2727.

[25] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: Real-time action recognition," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2617–2640, 2013.

[26] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, Dec. 2013.

[27] M. Jain, H. Jégou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2555–2562.

[28] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *Proc. 13th Eur. Conf. Comput. Vis. ECCV)*, Sep. 2014, pp. 581–595.

[29] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[30] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[31] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Human Behavior Understanding*. Springer, 2011, pp. 29–39.

[32] K. Charalampous and A. Gasteratos, "On-line deep learning method for action recognition," *Pattern Anal. Appl.*, vol. 19, no. 2, pp. 337–354, May 2014.

[33] L. Xie, W. Pan, C. Tang, and H. Hu, "A pyramidal deep learning architecture for human action recognition," *Int. J. Model., Identificat. Control*, vol. 21, no. 2, pp. 139–146, 2014.

[34] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[35] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[36] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 689–696.

[37] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[38] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, Crete, Greece, Sep. 2010, pp. 392–405.

[39] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion context: A new representation for human action recognition," in *Computer Vision—ECCV*. Springer, 2008, pp. 817–829.

[40] M. Lu and L. Zhang, "Action recognition by fusing spatial-temporal appearance and the local distribution of interest points," in *Proc. Int. Conf. Future Comput. Commun. Eng. (ICFCCE)*, 2014, pp. 75–78.

[41] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1948–1955.

[42] N. Ballas, Y. Yang, Z.-Z. Lan, B. Delezoide, F. Preteux, and A. Hauptmann, "Space-time robust representation for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2704–2711.

[43] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

[44] W. Li, Q. Yu, A. Divakaran, and N. Vasconcelos, "Dynamic pooling for complex event recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2728–2735.

[45] A. Gaidon, Z. Harchaoui, and C. Schmid, "Activity representation with motion hierarchies," *Int. J. Comput. Vis.*, vol. 107, no. 3, pp. 219–238, May 2014.

Ke Xu received the B.S. degree in electronic information and electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2013, where he is currently working toward the Ph.D. degree with the School of Electronic Information and Electrical Engineering.

His research interests include action recognition and abnormal event detection.

Xinghao Jiang (M'17) received the Ph.D. degree in electronic science and technology from Zhejiang University, Hangzhou, China, in 2003.

He was a Visiting Scholar with New Jersey Institute of Technology, Newark, NJ, USA, from 2011 to 2012. He is currently a Professor with the School of Information Security Engineering at Shanghai Jiao Tong University, Shanghai, China. His research interests include multimedia security and image retrieval, intelligent information processing, cyber information security, information hiding, and watermarking.

Tanfeng Sun (M'17) received the Ph.D. degree in information and communication system from Jilin University, Jilin, China, in 2003.

He was a Visiting Scholar with New Jersey Institute of Technology, Newark, NJ, USA, from 2012 to 2013. He is currently an Associate Professor with the School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include digital forensics on video forgery, videos content recognition, and understanding.

Dr. Sun is an SPS Member.