Pattern Recognition ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Learning discriminative features for fast frame-based action recognition

Liang Wang [a,b,c], Yizhou Wang [a,*], Tingting Jiang [a], Debin Zhao [b], Wen Gao [a]

[a] National Engineering Lab for Video Technology & Key Laboratory of Machine Perception (MoE), School of EECS, Peking University, Beijing, China
[b] School of Computer Science and Technology, Harbin Institute of Technology, Heilongjiang Province, China
[c] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, China

ARTICLE INFO

Keywords:
Frame-based action recognition
Feature mining

ABSTRACT

In this paper we present an instant action recognition method, which is able to recognize an action in real-time from only two continuous video frames. For the sake of instantaneity, we employ two types of computationally efficient but perceptually important features – optical flow and edges – to capture motion and shape characteristics of actions. It is known that the two types of features can be unreliable or ambiguous due to noise and degradation of video quality. In order to endow them with strong discriminative power, we pursue combined features, of which the joint distributions are different in-between action classes. As the low-level visual features are usually densely distributed in video frames, to reduce computational expense and induce a compact structural representation, we propose to first group the learned discriminative joint features into feature groups according to their correlation, then adapt the efficient boosting method as the action recognition engine which take the grouped features as input. Experimental results show that the combination of the two types of features achieves superior performance in differentiating actions than that of using each single type of features alone. The whole model is computationally efficient, and the action recognition accuracy is comparable to the state-of-the-art approaches.

1. Introduction

With the growth of the amount of video data from a variety of sources (such as surveillance systems and broadcasting industry) [1], action recognition engines have become a highly demanded essential tool of video content analysis in applications such as video retrieval and anomaly detection.

In the literature, there has been a large body of work on action recognition (e.g. [2,3]), in which different types of features, representations and classification models are proposed to identify actions in videos. Readers are referred to [1,4,5] for thorough survey on existing methods. However, in most of local-feature-based methods the adopted features are generic, for example, variants of spatial temporal tensors such as spatio-temporal interest points (e.g. [6,7]). The discriminative structures embedded in different types of actions are not fully considered in the first place. Hence, it sometimes requires a sophisticated classification model in order to differentiate the distributions of the same set of generic features over different types of actions. However, it is known that complex models are usually computationally expensive and prone to overfitting.

In addition, although most of state-of-the-art methods reported impressive performance, the routine of action recognition requires to extract a large amount of features from a buffered video of a considerable length. Consequently, an action can only be recognized after an entire period or even several repetitions of the actions. This is inadequate to many online applications that require instantaneous responses such as human computer interaction (HCI) and video surveillance. In the literature, there is strong psychological evidence showing that human usually can instantly tell what happens in a scene only with a glance (e.g. [8]). Whereas, this *instantaneity* property has been rarely considered as a key criterion in evaluating action recognition engines for online purpose. As a result, even for state-of-the-art action recognition methods, if supplied with only a couple of frames, the recognition accuracy is barely above chance (see example results in Figs. 6 and 8).

In this paper, we attempt to address the two key issues in action recognition mentioned above: discriminative feature learning and instantaneity in response. Here, the "instantaneity" has two aspects of meanings, i.e. fast in speed and being capable of making decision with a couple of frames. Specifically, the proposed method is able to recognize actions using any two consecutive frames of an action video of resolution 160×120 with an average speed of 0.04 s using an Intel Core i5-2400 3.10 GHz, 4.0G RAM PC.

* Corresponding author. Tel.: +86 10 62758116.
E-mail address: Yizhou.Wang@pku.edu.cn (Y. Wang).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.08.016

Particularly, we propose to achieve the goal – instant action recognition – from the following three aspects.

(1) *Pursuing discriminative simple features.* We employ two types of computationally efficient (simple) but perceptually important features – the optical flows [9] and Canny edges [10] – to capture the motion and shape/structure information in video sequences, since recent psychophysical studies reveal that neurons in visual area of human brains have multidimensional functional organization in processing shape and motion information [11], and human beings recognize motion/actions from the "motion pathway" and the "form pathway" [12]. It is known that both features are fast to compute and have small memory demand, whereas can be unreliable under certain circumstances [13] (e.g. aperture problem and sensitive to different kinds of degradations), and motion or shape features alone may have weak discrimination power. However, the combination of the two cues can exhibit distinctive stable semantic characteristics as shown in Fig. 2. Thus, we strongly believe that there always exist some reliable and inexpensive features which can be *further exploited* to serve for certain challenging visual tasks. The key to the success is how to identify the "right" ones.

Simple features are usually densely distributed in the data (video sequences in our context), the number of their combination is even larger. In order to quickly identify those discriminative local structures from the large search space, we propose a discriminative feature pursuit scheme based on the FP-tree mining approach [14] and the Apriori algorithm [15]. The discriminative features are selected as those maximizing the Kullback–Leibler (KL) divergence between their distributions in the target action class and the negative ones.

(2) *Grouping features for compact representation.* Although only discriminate features are selected during the feature pursuit, they are chosen independently and the number of selected features is still large. To further reduce the dimensionality of the action representations, we propose to group these features within a local range according to their co-occurrence in observed action video frames. We adopt spectral clustering techniques to group the features, in which all pairs of features within a local range are connected into an undirected graph, and the Phi coefficient [16] is adopted to measure the association/co-occurrence strength of a pair of features. Through experiments, we observe that the resulted feature groups give insight into the dependence structure in the action data.

(3) *Learning efficient recognition engines.* Boosting is an ensemble learning method, which integrates simple weak learners into a strong one. It is computationally efficient and yet have comparable accuracy to kernel-based methods. In the proposed approach, we learn boosted decision trees [17] as the recognition engine to satisfy the instantaneity criterion and achieve competitive recognition performance. However, instead of selecting individual features, we use feature groups to learn a decision tree. Majority voting among the features in a group is adopted to train the split function on the tree nodes. To recognize actions, we learn a boosting classifier for each action class in a one-vs-all manner.

In the following, we first introduce related work in Section 2. Then, the method of learning discriminative features is presented in Section 3. Action classification model and corresponding action recognition experiments are shown in Section 4. We conclude the paper in Section 5.

2. Related work

In the literature, there is very limited number of work focused on recognizing actions in a limited number of frames. Here, we introduce some related ones. Fei-Fei et al. [8] recognized actions in single images by integrating scene and object level image interpretations without leveraging motion cues. However, it is known that to obtain such high level semantic information from an image is not only computationally expensive, but also can be unreliable in general. Wang et al. [18] proposed a hidden conditional random field (hCRF) model, which combines the global and local features of motion fields to distinguish actions. The local patch features are clustered into "parts" each of which corresponds to a hidden variable of hCRF. These parts and their interactions are learned by maximizing the conditional likelihood of the hCRF on the motion fields of individual video frames. Schindler and Van Gool also studied the problem of recognizing actions from a small number of frames ("snippets"), and achieved encouraging results [19]. However, both methods require to track the people in the videos using a bounding-box, which limits their applications to the constrained or simple environments, like the Weizmann dataset [20]. Carlsson and Sullivan [21] proposed to model the actions using the silhouettes of human poses in video frames containing the actions. However, their method requires to explicitly extract the boundary of the actors, which is a difficult problem itself especially for real-world data.

The proposed method differs from these methods in two important aspects. First, the real-time/online feature explored in the proposed method is not possessed by other state-of-the-art methods to our best knowledge. Second, compared to [19,18,21], in the proposed method, only the action labels of videos are needed in both the training and test stage without explicit annotating the bounding-boxes of the actors.

3. Learning discriminative simple features

In this section, we present a method to discover a set of discriminative simple features from the patches of video frames. In the following, we will first introduce the feature learning method, followed by a theoretical explanation of the method.

3.1. Discriminative simple feature pursuit

3.1.1. Feature representation

To reduce computation, we extract features within local patches of size $M \times N$ in pixels. The features in a patch are quantized into an index set, namely *patch index feature*. As illustrated in Fig. 1, the patch is equally divided into an $m \times n$ grid (Fig. 1(e) and (f) shows an example of $m = n = 3$.) In each cell, optical flow and Canny edge features are quantized into a two-digit index, respectively, each ranging from 1 to 4. The first digit encodes a cell location in the grid, the second accounts for the feature's orientation (see Fig. 1(a)). For example, the cell highlighted by a yellow dotted rectangle in Fig. 1(e) is located at the 7-th cell of the patch, and its optical flow feature is quantized to be 4. Hence, the optical flow index feature of the cell is 74. Similarly, its shape index feature is 72 (Fig. 1(f)). If considering both shape and motion, the joint feature index is 742. Then, the *patch index feature* is composed by the feature indices of all the cells within it. If the mean magnitude of the features in a cell is smaller than a threshold, the feature related to the cell is ignored, and denoted as "X" in Fig. 1(e)–(g).

It should be noted that the reason we adopt the mean orientation of the optical flow and Canny edge features rather than their magnitudes is because that the former is generally more robust than the latter.

3.1.2. The discriminative features

We consider a feature to be discriminative, if it satisfies the following property: their occurrence frequency is high in the target action class but low in other classes. Correspondingly, we

Please cite this article as: L. Wang, et al., Learning discriminative features for fast frame-based action recognition, Pattern Recognition (2012), http://dx.doi.org/10.1016/j.patcog.2012.08.016

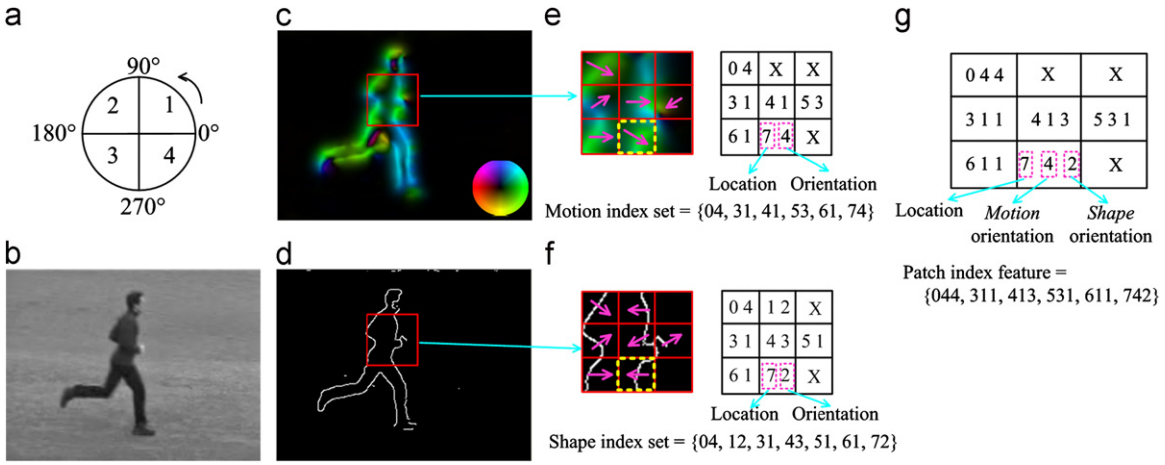

Fig. 1. Encoding patch features as indices. (a) Optical flows and Canny edges are quantized into four sections according to their mean orientation indexed by 1–4. (b) A frame of action "Running". (c) & (d) show its optical flows and Canny edges, respectively. The intensity indicates the magnitudes of the two features. The color in (c) encodes the optical flow orientation. (e)–(g) illustrate the encoding method of the cells and the patch. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

design two measurements for each index feature **a** for the two criterion, i.e. the *average positive occurrence (APO)* $\varrho_{D_c}(\mathbf{a})$ and the *positive-negative occurrence ratio (PNO)* $\upsilon_{D_c/D_{\bar{c}}}(\mathbf{a})$, as

$$\varrho_{D_c}(\mathbf{a}) = \frac{\#_{D_c}(\mathbf{a})}{|D_c|}, \quad \upsilon_{D_c/D_{\bar{c}}}(\mathbf{a}) = \frac{\varrho_{D_c}(\mathbf{a})}{\varrho_{D_{\bar{c}}}(\mathbf{a})} \tag{1}$$

where D_c and $D_{\bar{c}}$ denote the patches from the video frames of positive action class c and the patches from video frames of negative action classes, respectively. $\#_{D_c}(a)$ is the number of **a** appears in D_c. $\varrho_{D_c}(\mathbf{a})$ measures the popularity of **a** in D_c, and $\upsilon_{D_{\bar{c}}}(\mathbf{a})$ reflects **a**'s occurrence contrast between positive and negative classes, i.e. the discriminative power. **a** is identified as a *discriminative simple feature* if both $\varrho_{D_c}(\mathbf{a})$ and $\upsilon_{D_{\bar{c}}}(\mathbf{a})$ are above predefined thresholds.

3.1.3. Discriminative feature pursuit

To learn discriminative features between different video classes, frame patches are sampled from both the positive video frames and the negative ones, forming D_c and $D_{\bar{c}}$, respectively. On each frame, we uniformly extract overlapping image patches whose centers are 5 pixels away either vertically or horizontally. Then, *patch index features* are extracted from both D_c and $D_{\bar{c}}$.

Algorithm 1 describes the discriminative feature pursuit procedure, in which θ_ϱ and θ_υ denote the thresholds of the APO and PNO (refer to Eq. (1)), respectively. To learn discriminative features, we first employ the FP-growth frequent pattern mining technique in [14] to mine feature candidates whose occurrence frequency in D_c is larger than θ_ϱ. In Step 3, if a patch index feature **a** is a subset of another patch index feature **a**′, we say **a** is *contained* in **a**′. And the *maximal set* of a feature set is composed by the features which are not contained in any other features of the feature set. Then, in Step 4 we prune the candidates whose PNOs are smaller than θ_υ.

Algorithm 1. Discriminative simple feature pursuit algorithm.

Input: Positive patch set D_c, negative patch set $D_{\bar{c}}$, θ_ϱ, θ_υ.
Output: Learned discriminative feature set F_c for action class c.

1. $F_c = \emptyset$
2. Find all the patch index features whose occurrence frequency in D_c ($\varrho_{D_c}(\mathbf{a})$) are larger than θ_ϱ using the method in [14], denoted as set F_d.
3. Find the *maximal set* F_m from F_d.
4. For each patch index feature **a** in F_m,
   (a) Compute its occurrence frequency $\varrho_{D_{\bar{c}}}(\mathbf{a})$ in $D_{\bar{c}}$.
   (b) If $\upsilon_{D_c/D_{\bar{c}}}(\mathbf{a}) > \theta_\upsilon$, remove it from F_d and add it to F_c.
   (c) For each $\mathbf{a}' \in F_d$ also contained in **a**, if $\varrho_{D_c}(\mathbf{a}')/\varrho_{D_{\bar{c}}}(\mathbf{a}) < \theta_\upsilon$, remove it from F_d
5. If F_d is empty, return F_c; Otherwise, iterate from step 3.

Some learned discriminative simple features are illustrated in Fig. 2(a). In the figure, the discriminative simple features are learned to discriminate the two action pairs "Handwaving" vs "Running" and "Jogging" vs "Running". As can be observed, the mined discriminative features of each type capture the semantic structures of actions from different aspects. For instance, in Fig. 2(b), the optical flow features differentiate the "Waving" frame from the "Running" frame by the arm motion, whereas they distinguish the "Running" from "Waving" by the motion of leg and torso (which generally move horizontally). The Canny edge features discriminate the two action frames by the poses of leg contours. The combined features capture the two action frames' characteristic differences of motion and shape/pose on both the arm and leg simultaneously. For the confusing action frame pair "Jogging" vs "Running" (Fig. 2(c)), the motion feature alone cannot distinguish them well. However, the edge features identify the "Jogging" frame using the vertical lines along the torso, and pick up the slant lines along the leg as discriminative simple features of the "Running" frame. This may be due to the motion magnitude difference between the two actions. Compared to the shape feature, the combined features further include some new bits around the arms for the "Running" frame. These observations confirm the enhancement of the discriminative power brought by combining motion and shape features.

The detection of learned discriminative simple features in a video frame is also very efficient. Given a video frame, we first sequentially scan the $M \times N$ video patches in a fixed step length, then extract path index features using the method in Section 3.1.1. Feature detection is then achieved by checking whether the index set of a feature is contained in the feature index set of the patch.

3.2. Theoretical underpinning

This section presents theoretical underpinnings of the model and the algorithms presented in the previous section. Readers

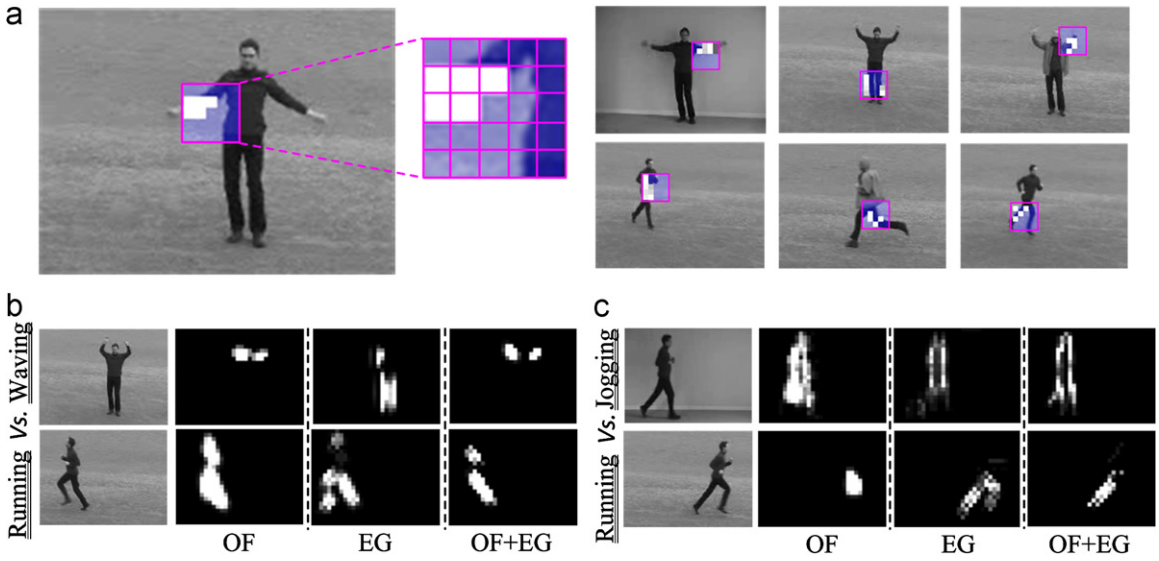

Fig. 2. Mined discriminative simple features (a) and their density maps (b), (c) for two action frame pairs from KTH dataset [22]. A discriminative simple feature corresponds to a mined subset (highlighted white feature bases) distributed in a frame patch within the purple sliding window. The discriminative simple features are learned from three types of features, i.e. optical flow (OF), Canny edge (EG) and their combination (OF+EG). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

who are more interested in algorithms and applications can jump to the next section.

Let A denote the set of all possible local feature configurations of cells in a patch (patch index features). Given a video frame I_c containing action instances of class c, it is composed by the instances of a set of patch index features denoted by $F_{c,I}=\{(\mathbf{a}_i,n_i^I)\}_{i=1}^{|A|}$, where n_i^I is the number of time $\mathbf{a}_i$ appears in I_c. Hence, a type of actions can be characterized by the set of patch index features and the distribution of the frequencies of these features appearing in the action frames. In words, given a set of patch index features, an action instance can be evaluated by the appearing frequencies of its patch index features. Similar to [23], the set of discriminative patch index features $A_c=\{\mathbf{a}_i\}_{i=1}^{|A_c|}$ for action class c can be found by those maximizing the average log-likelihood ratio

$$r=\frac{1}{N}\sum_{j=1}^{N}\log\frac{p(I_{c,j}|A_c)}{q(I_{c,j}|A_c)}=\frac{1}{N}\sum_{j=1}^{N}\log\frac{p((n_{\mathbf{a}_i}^{I_{c,j}})_{i=1}^{|A_c|})}{q((n_{\mathbf{a}_i}^{I_{c,j}})_{i=1}^{|A_c|})} \tag{2}$$

where $p(\cdot)$ is the probability density of the occurrence frequencies of discriminative patch index features in the video frames containing action c, and $q(\cdot)$ is the distribution of the same features in other type of actions. $n_{\mathbf{a}_i}^{I_{c,j}}$ is the number of $\mathbf{a}_i$ in $I_{c,j}$. N denotes the number of training video frames of action type c. When $N\to\infty$, the log-likelihood ratio converges to the Kullback–Leibler (KL) divergence between the distributions of the patch index features action type c and the other actions. Assuming the features are independent, we have

$$r=\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{|A_c|}\log\frac{p(n_{\mathbf{a}_i}^{I_{c,j}})}{q(n_{\mathbf{a}_i}^{I_{c,j}})} \tag{3}$$

As a result, using the method proposed in Section 3.1.3, we can find the discriminative features of action c.

4. Action classification model

In this section, we present two efficient action recognition methods using the learned discriminative features. We first introduce a frame-based action recognition method, then extend it to classify actions in videos.

4.1. Frame-based action classification model

We adopt a bag of word (BoG) representation for modeling actions. The visual words are patch-index-features. Since the discriminative features are learned independently, there may exist strong correlation between these features. On the one hand, the correlated features do not bring extra information for recognizing actions; on the other hand, high-dimensional data cause overfitting problem [24]. Considering this, we propose to group the features according to their correlation. A boosting classifier with decision trees as weak learners is adapted to classify each action. The input of the decision trees are feature groups rather than individual features.

4.1.1. Feature grouping

We define correlation between features according to their co-occurrence, and group correlated features via clustering.

The clustering proceeds by first building a graph to connect each feature in A_c. The graph nodes are the features, and an edge between two features is weighted by their *co-occurrence score*, is computed as the exponential of Phi coefficient [16] on their co-occurrence statistics in the training dataset. Particularly, given a patch set of action class c (D_c), the Phi coefficient between two features $\mathbf{a}_1$ and $\mathbf{a}_2$ is computed as

$$\phi(\mathbf{a}_1,\mathbf{a}_2)=\frac{n_{11}n_{00}-n_{01}n_{10}}{\sqrt{(n_{10}+n_{11})(n_{01}+n_{00})(n_{10}+n_{00})(n_{11}+n_{01})}} \tag{4}$$

where n_{11} is the number of patches that contain both of $\mathbf{a}_1$ and $\mathbf{a}_2$, n_{00} is the number of patches containing neither of them, and n_{01} and n_{10} are the numbers of patches containing either of them.

The graph partition method in [25] is employed to segment the graph into a number of groups. In Fig. 3, we show some instances of feature groups. It can be observed that the features in the same group can be closely related to some semantics. For example, the two features in Fig. 3(a) both account for the head and torso shape of the people when stretching their hands; and the features in Fig. 3(e) are grouped together corresponding to the leg motion in the "walking" action.

Please cite this article as: L. Wang, et al., Learning discriminative features for fast frame-based action recognition, Pattern Recognition (2012), http://dx.doi.org/10.1016/j.patcog.2012.08.016

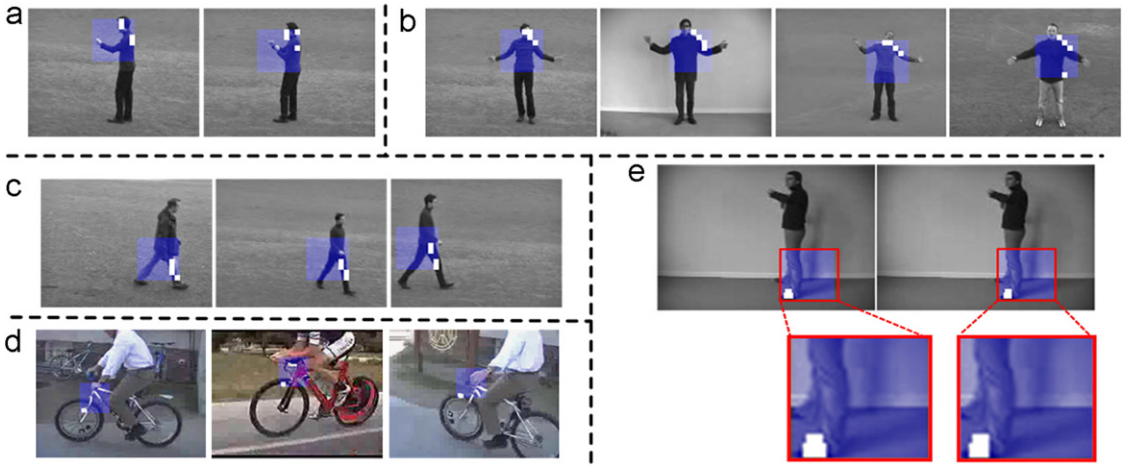

Fig. 3. Examples of feature groups. (a)–(e) each shows a number of features in one group.

4.1.2. Boosted decision trees with grouped features

The Boost framework with decision trees as weak learners are employed as the classification model due to its computational efficiency and competitive classification accuracy. However, instead of using individual features, we learn the split functions of tree nodes based on the feature groups. We call this model as *boosted trees with grouped features* (*BTwGF*).

The proposed BTwGF are trained in a one-vs-all manner. Let $H_c(I)$ denote the classifier of action class c. It is composed of a set of weighted decision trees $\{h_{c,i}\}_{i=1}^{n_c}$

$$H_c(I) = \sum_i \alpha_{c,i} h_{c,i}(I) \tag{5}$$

where $\alpha_{c,i}$ is the learned weights of the i-th decision tree $h_{c,i}(\cdot)$. The action label c^* for an action video frame can be determined as

$$c^* = \arg\max_c H_c(I) \tag{6}$$

In training a decision tree $h_{c,i}$, on each tree node, we randomly select a subset of feature groups; for a feature group, we run a Boolean test on each feature of the group, and the corresponding split threshold is recorded as the value that maximizes the information gain. (A datum is assigned to a tree branch according to the majority voting of these Boolean tests of the features.) We compute the information gain for each feature group using the training data at the node. The group with the maximum sum of information gains is assigned to the tree node, and each Boolean test threshold of the features are employed to the split function of the tree node. This step can be considered as a *Max pooling* operation on the grouped feature variables. This max-like behavior is observed in cortical neurons during visual processing for object recognition, and it implies that this nonlinear neuronal function induces feature invariance while preserving feature specificity [26,27]. We stop growing a decision tree when the information gain is trivial or at a shallow depth (three layers in our implementation) to ensure its good generalization ability.

The learned trees serve not only as a computational engine, but also as a discriminative structural representation for actions. We call them *patch-based actionlets*. Some example actionlets are illustrated in Fig. 4. The actionlets can be seen as a type of discriminative template describing the shape and motion constraints for the action. For example, the second example in Fig. 4 is a snapshot of the hand waving action. It shows an actionlet which contains three nodes accounting for a vertical line in the leg and an upward motion in the arm. (It should be noted that there are many actionlets existing in one video frame, however, in Fig. 4 only shows one actionlet per frame, it is for illustration purpose.)

4.2. Video-based action classification model

We further extend the frame-based method to recognize actions using more frames or even whole videos. The assumption is, if we can get good predictions using some of the frames in a video, by accumulation, the prediction accuracy can be improved over the whole video sequence.

For training the video-based model, we use the same method as the frame-based model mentioned above. Action recognition/classification on a given video clip is then accomplished by the following steps. (i) Key frame sampling. Key-frames are sampled from the video clip every four frames to reduce computational cost. (ii) Key frame selection. Because the key frames are sampled without any preference – some snapshots of actions can be ambiguous, while some are very distinguishable – different frames provide diverse confidence in judging its action label. We propose to further select a subset of *confident key frames* to participate in recognizing actions as follows. We compute the confidence score of a frame I_t as

$$\text{conf}(I_t) = \max_c \frac{\sum_i \alpha_{c,i} h_{c,i}(I)}{\sum_i \alpha_{c,i}}$$

(refer to Eq. (5)). If $\text{conf}(I_t) > 0.6$, the key frame I_t is a *confident key frame*. The selected confident key frame set is denoted as $\mathcal{K}$. (iii) We recognize the action in these confident key frames using the method in Section 4.1. (iv) The action label c^* of the video is determined by majority voting

$$c^* = \arg\max_c \sum_{I_t \in \mathcal{K}} H_c(I_t) \tag{7}$$

4.3. Experimental evaluation

Two challenging action datasets are used to evaluate the proposed method, the YouTube dataset [3] and the UCF sport dataset [28]. The YouTube dataset is collected from YouTube web site containing 11 types of actions, such as basketball shooting and volleyball spiking. There are 1595 video sequences in total. For each action type, the videos are manually divided into 25 groups by the authors of [28] each of which contains videos of similar background or subclips of a same video. The UCF sport action dataset is collected from broadcast television channels such as the BBC and ESPN containing various

Please cite this article as: L. Wang, et al., Learning discriminative features for fast frame-based action recognition, Pattern Recognition (2012), http://dx.doi.org/10.1016/j.patcog.2012.08.016

Fig. 4. Examples of patch-based actionlets for the KTH dataset [22]. In the first two frames, each decision tree/actionlet has three nodes; the rest have two nodes each.

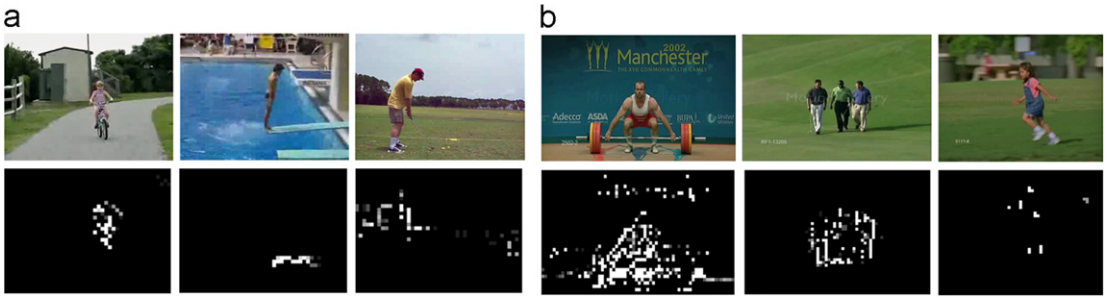

Fig. 5. Some example frames of (a) the YouTube dataset and (b) the UCF sport action dataset and their density maps of the learned discriminative simple features.

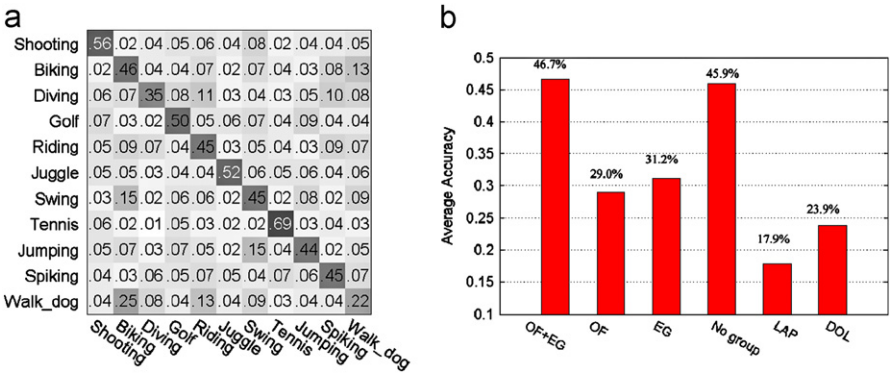

Fig. 6. Results of frame-based action recognition method on YouTube dataset. (a) Confusion matrix of the action recognition accuracy. (b) Recognition accuracy comparison among the proposed method using joint features (OF+EG), using motion feature (OF), using shape feature (EG), the method in [2] without the feature grouping (No group) and the methods in [6] (LAP) and [7] (DOL).

sports videos. The dataset contains only 146 video sequences but covers ten types of actions including "diving", "golf swinging", etc. We use it to test the robustness of the proposed method given limited amount of training data of challenging scenes.

In all the experiments, we set the average positive occurrence (APO) Eq. (1) threshold to 2; the positive–negative occurrence ratio (PNO) Eq. (1) threshold of each action is chosen such that there are K discriminative features mined ($K=700$ for YouTube dataset and 1200 for UCF dataset).

To evaluate the frame-based action classification method, we divide the videos in a dataset into a training set and a testing set. Then, video frames are uniformly sampled from the videos in a fixed step length (e.g. 6) for training and testing. When evaluating the video-based action classification method, we randomly sample subclips from the training and testing videos to build the training and testing datasets.

4.3.1. Evaluation on YouTube dataset

Cross-validation is used to evaluate the proposed method. In our implementation, the sampled patch size is $M=N=56$ and each patch is divided into a 8×8 grid with $m=n=7$ in pixels (refer to Section 3.1.1 for notation description). We learn 700 discriminative patch index features for each action class and cluster them into 300 feature groups using the method in Section 4.1.1. The distributions of the learned discriminative simple features are illustrated in Fig. 5(a). It is interesting to note that (i) most of the features are distributed on the acting subjects; (ii) some features are located on semantic meaningful context of the actions. For example, the springboard of the "Diving" action. The observation confirms that the learned features capture semantically meaningful parts of the action videos.

Frame-based action recognition performance. Fig. 6 compares the action recognition accuracy between the proposed method and two popular methods [22,7] which are based on the spatio-temporal interest point (STIP) detectors. To test the performance of the STIP based method, we use the same set of training/testing frames as ours. The STIPs are clustered into 2000 clusters by the K-means algorithm. As can be seen, the proposed method outperforms the STIP based methods. (Extracting STIPs requires usually more than six frames.)

Video-based action recognition performance. The confusion matrix of the action recognition accuracy is shown in Fig. 7(a). Fig. 7(b) shows the recognition accuracy changes when testing the proposed method on video clips of different lengths. For comparison, the same measure of the methods in [22,7] is plotted. It can be seen that the recognition accuracy of the proposed method increases when more video frames are provided. This demonstrates the accumulation effect of the frame-based recognition improves the recognition accuracy of videos.

Table 1 shows the comparison result between our method and a method proposed in [3]. The method in [3] recognizes actions using a combination of static and motion features, which achieves

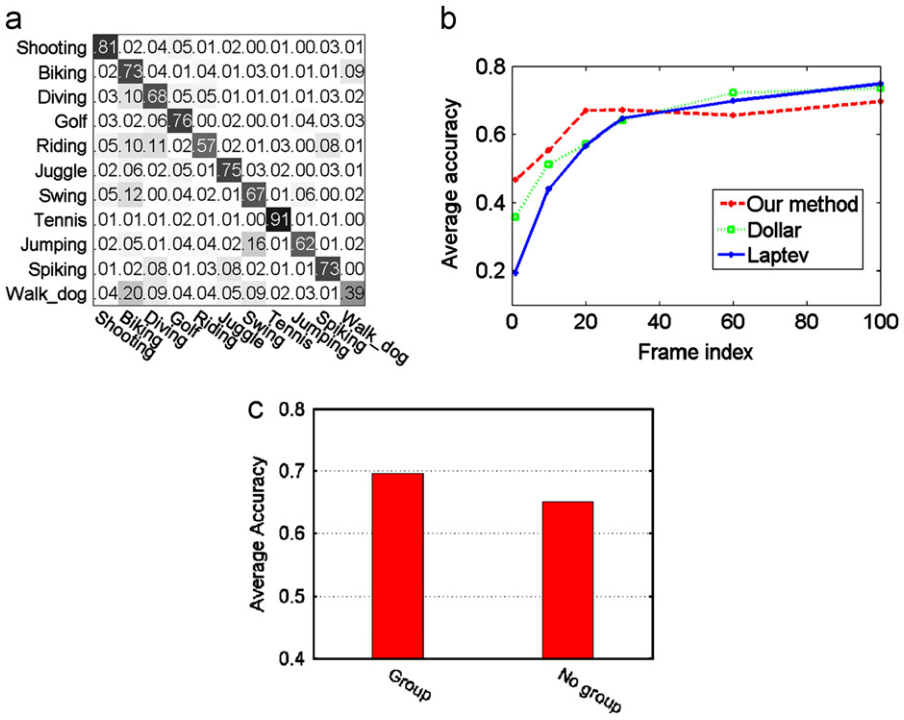

Fig. 7. Results of video-based action recognition method on YouTube dataset. (a) Confusion matrix of the action recognition accuracy. (b) Recognition accuracy curves over different frame length. (c) Comparison of average recognition accuracy with the method in [2].

Table 1
Comparison results on video-based action recognition on the YouTube dataset. The method in [3] used three types of features: motion, static and hybrid. The feature number is the average number of detected features in each training video.

Method	Ours	Method in [3]		
		Motion	Static	Hybrid
Accuracy	69.7%	65.1%	63.0%	71.2%
Feature number	3000	400	8000	8400

Table 2
Computational complexity on the YouTube dataset (1st and 2nd rows) and the UCF dataset (3rd and 4th rows).

Feature number	Resolution	Optical flow	Canny edge	Boosting	Disc. feature detection	Total
3300	320×240	0.0464	0.0094	0.0001	0.0764	0.1323
7700					0.1756	0.2315
2700	$\sim 400 \times 300$	0.0624	0.0121	0.0001	0.0735	0.1481
6300					0.1913	0.2659

the best performance on the YouTube dataset to our knowledge. The comparison shows that the proposed method achieves a comparable performance even using fewer number of features.

Computational complexity. Table 2 shows the average computational time on the YouTube dataset when using different number of discriminative features. It can be seen that the main time consumption is spent on feature detection.

4.3.2. Evaluation on UCF sport dataset

Since the number of videos in the dataset is very limited (about 10 sequences for each action type), for each action class, we randomly select Round(min(0.9∗N,N−1)) video sequences for training the action classification model, and the rest ones for testing. In the feature grouping, we cluster the 1200 discriminative patch index features of each action class into 300 groups. The size of the sampled patches and the grid size are the same as used in the YouTube dataset. Some example of the discriminative features are shown in Fig. 5(b).

Frame-based action recognition performance. The confusion matrix of the action recognition accuracy on UCF dataset is shown in Fig. 8(a). Although the dataset is complex and of small number of training data, we still achieve a reasonably good results (an average recognition accuracy of 60.9%). From the results in Fig. 8(a), we can see that the action "Golf Swing" and "Kick" are likely to be confused with "walking". This is because there are many frames of the two action videos also contain the walking actions. The three actions, "Diving", "Lifting" and "Swingbar", possess the highest classification accuracy. The comparison of the proposed method with the methods in [2,22,7] are shown in Fig. 8(b). As can be seen, our method is more robust and accurate when training set is small.

Video-based action recognition performance. The results are shown in Fig. 9. The recognition accuracy of our method using 60 frame length clips is 75.0%. It increases when more video frames are used but begins to drop when over 60 frames. This is because some actions in this dataset only last for about 40–60 frames. When more frames are provided, some action-irrelevant video frames also come to vote for the action types, which deteriorates the action recognition accuracy. (For example, the golf swing action contains walking or running in the video. The same case in the soccer videos.) The proposed method outperforms the method in [28], which uses a template based method with a recognition accuracy of 69.1%. The method in [28] needs to annotate a circle of actions in the videos so as to train the MACH model. Whereas, the proposed method achieves a better result using less supervision in training.

5. Conclusion

In this paper, we took the "instantaneity" criteria into consideration when building action recognition models and proposed

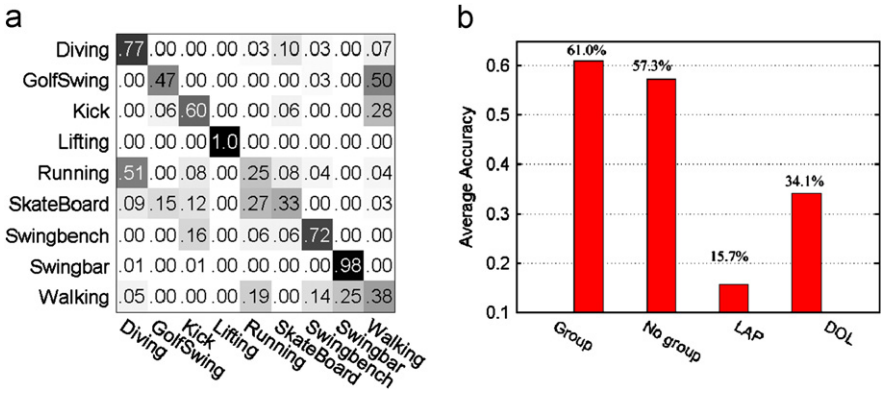

Fig. 8. Results of frame-based action recognition method on UCF dataset. (a) Confusion matrix of the action recognition accuracy. (b) Recognition accuracy comparison among the proposed method, the methods in [2,6,7].

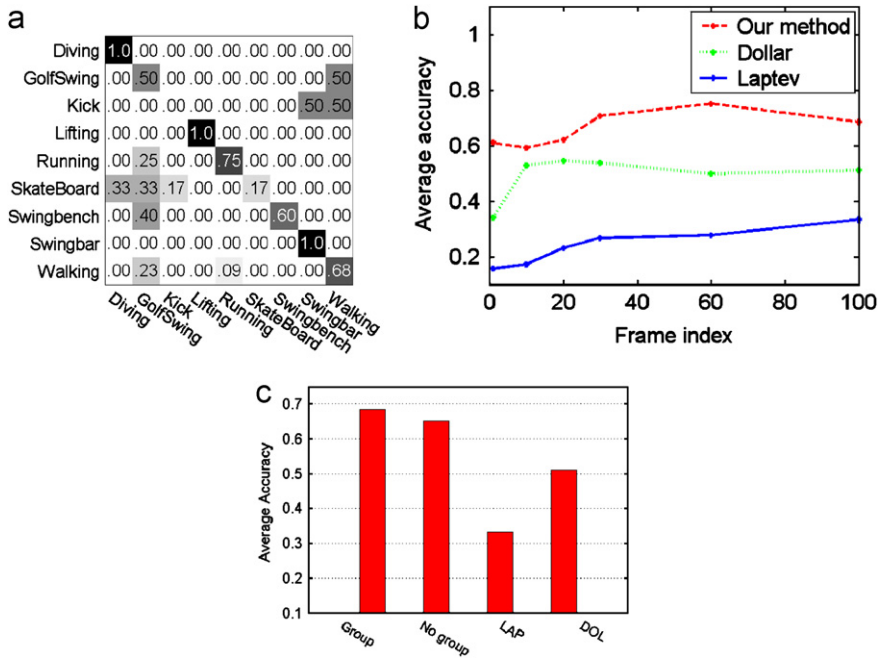

Fig. 9. Results of the video-based action recognition results on the UCF dataset. (a) Confusion matrix of the recognition accuracy of proposed method. (b) Recognition accuracy curves over different frame length. (c) Comparison of action recognition accuracy with the method in [2].

an efficient action recognition method by pursuing computationally efficient and discriminative simple features from couple of video frames. And we propose to group correlated features to improve the compactness of model. The proposed discriminative features learning method can be generalized to discover distinguishing features in other applications.

However, the proposed method has the following limitation. The discriminative features are learned only as spatial discriminative configurations of local features, and the recognition of the action is performed in a frame-based manner. The temporal distribution of the features are not taken into consideration in both the feature learning and the action recognition. In the future, we will extend the proposed method by studying efficient models that incorporate temporal information so as to enhance the current model.

Acknowledgement

The authors would like to thank for the support from research grants 973-2009CB320904, and the National Science Foundation of China NSFC-61272027, 61272321, 61103087.

References

[1] J. Candamo, M. Shreve, D. Goldgof, D. Sapper, R. Kasturi, Understanding transit scenes: a survey on human behavior-recognition algorithms, IEEE Transactions on Intelligent Transportation Systems 11 (1) (2010) 206–224.
[2] L. Wang, Y. Wang, W. Gao, Mining layered grammar rules for action recognition, International Journal of Computer Vision 93 (2) (2011) 162–182.
[3] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos in the wild, in: Proceedings of CVPR, 2009.
[4] R. Poppe, A survey on vision-based human action recognition, Image and Vision Computing 28 (6) (2010) 976–990.
[5] G. Lavee, E. Rivlin, M. Rudzsky, Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video, IEEE Transactions on Systems, Man, and Cybernetics 39 (5) (2009) 489–504.
[6] I. Laptev, M. Marszalek, C. Schmid, B. Rozeneld, Learning realistic human actions from movies, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[7] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Proceedings of the IEEE International Workshop on PETS, 2005, pp. 65–72.
[8] L. Fei-Fei, A. Iyer, C. Kock, P. Perona, What do we see in a glance of a scene? Journal of Vision 7 (10) (2007) 1–29.
[9] B. Horn, B. Schunck, Determining optical flow, Artificial Intelligence 17 (1981) 185–203.
[10] C. John, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.

[11] D. Malonek, R. Tootell, A. Grinvald, Optical imaging reveals the functional architecture of neurons processing shape and motion in owl monkey area mt, Proceedings of the Royal Society of London 258 (1352) (1994) 109–119.
[12] J. Lange, K. Georg, M. Lappe, Visual perception of biological motion by form: a template-matching analysis, Journal of Vision 6 (8) (2006) 836–849.
[13] Y. Ke, R. Sukthankar, M. Hebert, Volumetric features for video event detection, International Journal of Computer Vision 88 (3) (2010) 339–362.
[14] J. Han, J. Pei, Y. Yin, R. Mao, Mining frequent patterns without candidate generation: a frequent-pattern tree approach, Data Mining and Knowledge Discovery 8 (2004) 53–87.
[15] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, in: ACM SIGMOD, 1993, pp. 26–28.
[16] H. Cramer, Mathematical Methods of Statistics, Princeton University Press, 1946.
[17] P. Viola, M. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.
[18] Y. Wang, G. Mori, Learning a discriminative hidden part model for human action recognition, in: NIPS, 2008.
[19] K. Schindler, L. Van Gool, Action snippets: How many frames does human action recognition require? in: Proceedings of CVPR, 2008.
[20] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: Proceedings of the International Conference on Computer Vision, 2005.
[21] S. Carlsson, J. Sullivan, Action recognition by shape matching to key frames, in: Proceedings of the Workshop on Models versus Exemplars in Computer Vision, 2001.
[22] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: Proceedings of ICPR, 2004, pp. 32–36.
[23] Y. Wu, S. Si, H. Gong, S. Zhu, Learning active basis model for object detection and recognition, International Journal of Computer Vision 90 (2) (2010) 198–235.
[24] F. Torre, T. Kanade, Multimodal oriented discriminant analysis, in: Proceedings of the International Conference on Machine Learning, 2005, pp. 177–184.
[25] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[26] K. Sakai, S. Tanaka, Spatial pooling in the second-order spatial structure of cortical complex cells, Vision Research 40 (7) (2000) 855–871.
[27] M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex, Nature Neuroscience 2 (1999) 1019–1025.
[28] M. Rodriguez, J. Ahmed, M. Shah, Action mach: a spatio-temporal maximum average correlation height filter for action recognition, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2008.

Liang Wang received the B.E., M.E., and Ph.D. degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 2005, 2007, and 2011, respectively. He is now a Postdoc in the Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, China. His research interests include action recognition, pose estimation and video surveillance.

Yizhou Wang received the B.E. degree from the Electrical Engineering Department of Tsinghua University, Beijing, China, in 1996, the M.E. degree from the National University of Singapore in 2000 and the Ph.D degree from University of California, Los Angeles (UCLA) in 2005. He worked as a computer hardware consultant for Hewlett-Packard, Singapore, from 1996 to 1998. From 2005 to 2007, he was a research staff in Palo Alto Research Center (Xerox PARC). Currently, he is a professor jointly in National Engineering Lab for Video Technology and Key Laboratory of Machine Perception (MoE), School of EECS, Peking University. His research interests include computer vision and computational visual arts.

Tingting Jiang received the B.S. degree in computer science from University of Science and Technology of China in Hefei, China, in 2001 and the Ph.D. degree in computer science from Duke University, Durham, North Carolina, USA, in 2007. She is now an assistant professor of computer science at Peking University, Beijing, China. Her research interests include computer vision, image and video quality assessment.

Debin Zhao received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 1985, 1988, and 1998, respectively. He is now a professor in the Department of Computer Science, HIT. He has published over 200 technical articles in refereed journals and conference proceedings in the areas of image and video coding, video processing, video streaming and transmission, and pattern recognition.

Wen Gao received the Ph.D. degree in electronics engineering from the University of Tokyo, Japan, in 1991. He is a professor of computer science at Peking University, China. Before joining Peking University, he was a professor of computer science at the Harbin Institute of Technology from 1991 to 1995, and a professor at the Institute of Computing Technology of Chinese Academy of Sciences. He has published extensively, including four books and more than 600 technical articles, in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. He served or serves on the editorial board for several journals, such as the IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Multimedia, IEEE Transactions on Autonomous Mental Development, EURASIP Journal of Image Communications, and Journal of Visual Communication and Image Representation. He has chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations. He is a fellow of the IEEE.

Please cite this article as: L. Wang, et al., Learning discriminative features for fast frame-based action recognition, Pattern Recognition (2012), http://dx.doi.org/10.1016/j.patcog.2012.08.016