Human Activity Recognition Using Hierarchically-Mined Feature Constellations

Antonios Oikonomopoulos[1] and Maja Pantic[1,2]

[1] Comp. Dept., Imperial College London, UK
[2] EEMCS, University of Twente, The Netherlands

Abstract. In this paper we address the problem of human activity modelling and recognition by means of a hierarchical representation of mined dense spatiotemporal features. At each level of the hierarchy, the proposed method selects feature constellations that are increasingly discriminative and characteristic of a specific action category, by taking into account how frequently they occur in that action category versus the rest of the available action categories in the training dataset. Each feature constellation consists of n-tuples of features selected in the previous level of the hierarchy and lying within a small spatiotemporal neighborhood. We use spatiotemporal Local Steering Kernel (LSK) features as a basis for our representation, due to their ability and efficiency in capturing the local structure and dynamics of the underlying activities. The proposed method is able to detect activities in unconstrained videos, by back-projecting the activated features at the locations at which they were activated. We test the proposed method on two publicly available datasets, namely the KTH and YouTube datasets of human bodily actions. The acquired results demonstrate the effectiveness of the proposed method in recognising a wide variety of activities.

1 Introduction

Human action recognition has recently become a very important area of research, due to its importance to applications like scene understanding, video retrieval and human-computer interaction. However, it still remains a very difficult task, due to unsolved challenges like camera motion, clutter, and the inherent variability in the conduction of activities by different subjects. For more information, we refer the reader to [1].

Sparse spatiotemporal interest point representations have been widely used for human action recognition. Typical examples are the space-time interest points of Laptev and Lindeberg [2], and Lowe's Scale Invariant Feature Transform (SIFT) [3]. Dollar et al. [4] use 1D Gabor filters for capturing intensity variations in time. In [5] this approach is refined by using Gabor filters in both spatial and temporal dimensions. Inspired by SIFT, the Speeded Up Robust Features (SURF) of Bay et al. [6] utilize second order Gaussian filters and the Hessian matrix for detecting keypoints. Jhuang et al. [7] use Gabor filters for detecting their C-features. This method is extended by Schindler and Van Gool [8], by combining shape and optical flow responses. Finally, Ali and Shah [9] use kinematic features in order to represent human activities.

G. Bebis et al. (Eds.): ISVC 2013, Part I, LNCS 8033, pp. 150–159, 2013.
© Springer-Verlag Berlin Heidelberg 2013

Due to the inadequacy of sparse representations to model certain action instances, dense representations have recently gained a lot of attention. Gilbert et al. [10] detect dense Harris corners in order to represent target activities. Schechtman and Irani [11] extract self-similarity descriptors densely throughout images or videos, while Seo and Milanfar [12] propose the use of dense 3D Local Steering Kernels (LSK) for the same purpose. Finally, Amer and Todorovic [13] extract Stacked Convolutional ISA (SCISA) features at pre-defined grid locations in order to model and recognize complex activities.

The amount of information contained in dense representations has made the use of traditional feature selection methods, like Adaboost [14] prohibitive. For this reason, data mining methods have been proposed as an alternative. Quack et al. [15] use association rule mining in order to perform feature selection for object detection. Gilbert et al. [16] perform mining in various levels, creating a hierarchy of features. A slightly different approach is followed by Wang et al. [17], who use the Emerging Pattern mining method [18] for detecting activities. Their main idea is to select features that are much more frequent in the positive than in the negative set.

In this paper we propose the use of dense features for representing the visual information that is present in a given video. Given a training set of different activities, we perform data mining in order to select the most relevant and discriminative features for each class. Similar to [16], we apply our data mining method in a hierarchical manner, in which increasingly complex features are selected as the number of levels in the hierarchy increases. Contrary to the method in [16], where all features that fall within a local spatiotemporal neighborhood are taken into account, we create our complex features by considering instead n-tuples of features. With the exception of the first level, which directly operates on the detected dense features in the image sequence, the features that are being considered at each level of the hierarchy for creating a feature constellation, are the ones that were selected at the previous level. Therefore, at each subsequent level, the number of original features included in a new feature constellation is n-times larger compared to the previous level. The justification for this approach is a two-fold. Firstly, due to the use of a dense representation, this approach is very efficient in terms of storage and memory requirements. Secondly, by only encoding n-tuples of previously selected features, the constellations that we create are robust against accidental matches to features in the background. The use of a hierarchy allows us to discriminate between a large amount of action categories, as our results in several datasets show, namely the KTH [19] and YouTube [20] datasets of human actions. A block-diagram overview of the proposed method is illustrated in Fig. 1.

The rest of this paper is organized as follows: Section 2 describes our feature detection process. In section 3 we describe the utilized mining method, while our hierarchical approach to mining and the consequential creation of complex features is described in section 4.1. Section 5 contains our experimental results and in section 6, we draw our conclusions.

2 Feature Detection

Features based on Local Steering Kernels (LSK) have been extensively used for object [21] and action detection [12]. The idea behind 3D LSKs is to robustly estimate the

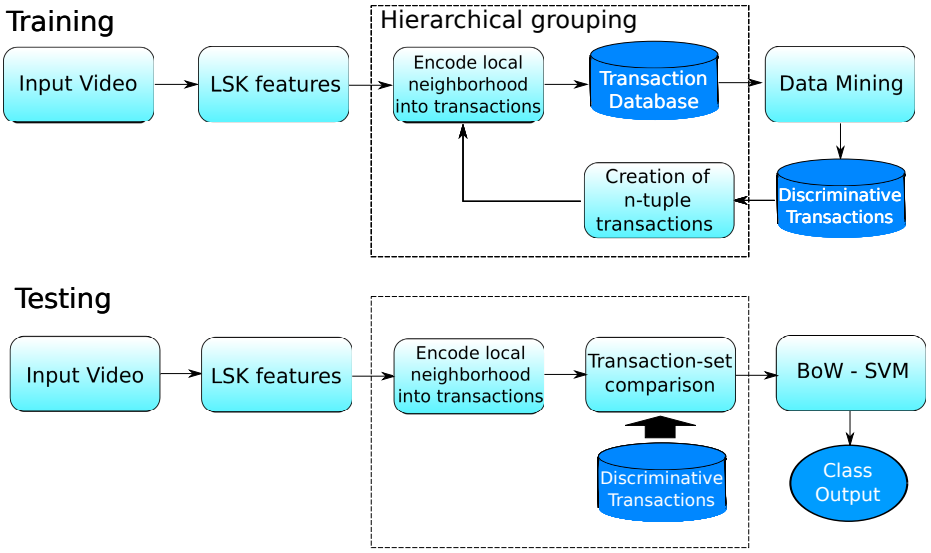

Fig. 1. Overview of the proposed method

local spatiotemporal structure of a scene by analyzing the estimated gradients in space and time, and use this information to compute the shape and size of a canonical kernel:

$$K(\mathbf{x}_l - \mathbf{x}_i) = \sqrt{det(\mathbf{C}_l)} exp\left\{\frac{(\mathbf{x}_l - \mathbf{x}_i)^T \mathbf{C}_l (\mathbf{x}_l - \mathbf{x}_i)}{-2h^2}\right\}, \tag{1}$$

where $\mathbf{x}_i$ is a pixel of interest, $\mathbf{x}_l, l = 1 \ldots P$, are pixels belonging in a spatiotemporal neighbourhood Ω_l around $\mathbf{x}_i$, h is a normalization parameter, and $\mathbf{C}_l \in \mathbb{R}^{3\times 3}$ is a local covariance matrix estimated from a collection of first derivatives along the spatiotemporal axes within Ω_l. Similar to [12], $\mathbf{C}_l$ is calculated by invoking the Singular Value Decomposition (SVD) with regularization of a matrix $\mathbf{J}_l$ that collects the first derivatives along the space-time axes. $\mathbf{C}_l$ is given by:

$$\mathbf{C}_l = \gamma \sum_{q=1}^{3} \alpha_q^2 \mathbf{v}_q \mathbf{v}_q^T \quad \in \mathbb{R}^{3\times 3}, \tag{2}$$

$$\begin{aligned} \alpha_1 &= \frac{s_1 + \lambda'}{\sqrt{s_2 s_3} + \lambda'}, \quad \alpha_2 = \frac{s_2 + \lambda'}{\sqrt{s_1 s_3} + \lambda'}, \\ \alpha_3 &= \frac{s_3 + \lambda'}{\sqrt{s_1 s_2} + \lambda'}, \quad \gamma = \left(\frac{s_1 s_2 s_3 + \lambda''}{P}\right)^{\beta}, \end{aligned} \tag{3}$$

where λ', λ'' are parameters used to dampen noise and β is used in order to restrict the value of γ. Furthermore, s_1, s_2, s_3 and $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are, respectively, singular values and singular vectors, given by the compact SVD of $\mathbf{J}_l$.

Similar to [12], we set λ', λ'' and β to 1, 10^{-8} and 0.29 respectively. To address scaling, we apply this process to a space-time pyramid that is created from each available video.

3 Data Mining

We use the Emerging Pattern (EP) mining method [17] to select features characteristic of a human action, due to its effectiveness in selecting suitable features in large datasets. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n attributes called items, and let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database. Each transaction $T \in D$ contains a subset of the items in I. A subset X of I is also called an itemset. If $X \subseteq T$ then we say that the transaction T contains the itemset X. The support of an itemset X is defined as: $\rho_D = count_D(X)/\,||D||$, where $count_D(X)$ is the number of transactions in D that contain X. Given two datasets D_1, D_2, the growth ratio of X from D_1 to D_2 is defined as:

$$\upsilon_{D_1/D_2}(X) = \begin{cases} 0, & \text{if } \rho_{D_1}(X) = 0 \text{ and } \rho_{D_2}(X) = 0 \\ \infty, & \text{if } \rho_{D_1}(X) = 0 \text{ and } \rho_{D_2}(X) \neq 0 \\ \frac{\rho_{D_2}(X)}{\rho_{D_1}(X)}, & \text{otherwise} \end{cases} \tag{4}$$

The main idea of the EP mining method is to find itemsets whose growth ratio from a negative set D_1 to a positive set D_2 is larger than a constant ε. Let us denote with $U(\alpha)$ the set of unique transactions that are selected using eq. 4, where α is the action label. Then, $U(\alpha) = \{X : \upsilon_{D_1/D_2}(X) > \varepsilon\}$, where ε is larger than 1. In this work, we set $\varepsilon = 2$, as a compromise for selecting features that have a good support in their respective class and do not overfit the training set.

4 Recognition

Given an $N \times N$ block of LSK descriptors, a transaction is defined as a collection of items, each of which describes a single element in the block, and represented by a string of numbers. A transaction consists of N^2M^2P items at most, where $M \times M \times P$ is the size of a single LSK descriptor. The reason for considering blocks is to encode groups of neighbouring features that co-occur. Each cell in the block, depending on its position, is assigned a unique ID. This ID constitutes the first part of the string that describes the items in the cell, and takes values from 1 to N^2. The second part of each item is assigned a similar number depending on its spatiotemporal position in the cell. This process is outlined in Fig. 2. Since an LSK descriptor is of size $M \times M \times P$, this number takes values from 1 to M^2P. In this work we set $M = 3$ and $P = 7$. Therefore the second part of each item takes values from 1 to 63. Finally, the third and last part of each item consists of the quantized value of each element of the LSK descriptor.

In order to perform mining, we initially cluster the transactions of each action class. Two transactions belong to the same cluster if the number of their items and their values are similar. We use an Euclidean metric to compare item values, and consider two transactions similar if the number of the matching elements is equal or above the 60%

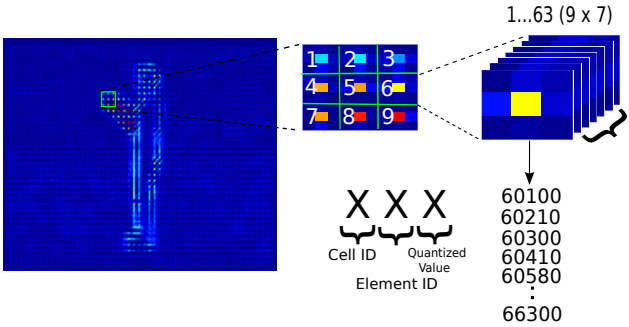

Fig. 2. Each cell in a $N \times N$ block is a single LSK descriptor of dimensions $M \times M \times P$. In the example above, $N = 3$, $M = 3$ and $P = 7$. Each item consists of: Block ID (1-9)-Cell ID (1:63) and the quantized value of the descriptor element. The final transaction consists of all items that describe the block elements.

of the length of the shortest transaction. The support of each cluster is then calculated as the number of the transactions that constitute it over the total number of transactions. Following a similar approach, we compare each cluster with the transactions in the negative set. Finally, using eq. 4, we compute the growth ratio of each cluster center and keep the ones that exceed ε.

4.1 Complex Features

We employ a hierarchical framework for selecting complex features and increase the discriminative ability of our method. Let us denote with X_l, U_l the transaction dataset and the set of selected transactions at level l respectively. The dataset X_{l+1}, on which the next level of mining will be performed is created using the unique transactions of U_l. Each member of X_{l+1} is a complex feature that consists of n-tuples of transactions from U_l. We use all possible combinations between the members of U_l for constructing X_{l+1}. This number is equal to $\binom{M_l}{n}$, where M_l is the number of unique transactions in U_l. These transactions are appended with the ID of the block in which they appear in the training set, but with an increased block size. By doing this, we encode the relative location of the features within the larger block. Let us denote with Λ_i, $i = 1....n$ the n features that are to be included in a new feature constellation. Then, the new feature constellation is represented as $\{r_1\Lambda_1, \ldots, r_n\Lambda_n\}$, where $r_j \subset \{1...C\}$, $j = 1...n$ is the ID position of the block and C is the number of cells in the block. At each level, the block size is increased, and created features cover a larger area of the image sequence. After the dataset X_{l+1} is created, the mining process of section 3 is repeated, resulting in a new set of transactions U_{l+1}.

4.2 Classification and Localization

To classify a given video, its detected 3D LSK features are converted into transactions and initially matched to the discriminative features of the first level. If a match is found, its index in the training set is stored and associated with the test transaction. Given a

matched test feature i and the n r_j indexes that constitute a complex training feature of the next level, the algorithm searches within the local neighborhood around i to find test features that have the same indexes as in the ones in the complex feature. If they are found, an additional test on their spatiotemporal configuration is performed. This process is subsequently repeated for each level. By using such an approach, we avoid the multiple-level encoding of the test transactions, and speed up recognition.

We use a Bag of Word (BoW) approach in order to classify a given image sequence. Each BoW is a histogram of the transactions of each level that match the features in the test set, weighted by the degree of their match. This representation is used as input to a multiclass Support Vector Machine with a Gaussian radial basis function kernel for classification.

For localization, given the matches of the test features to the transaction database, we backproject their locations onto the image plane in order to infer where the action is. Since the matched features are concentrated on areas of a significant amount of motion, this approach tends to localize moving parts rather than fitting a bounding box around the subject.

5 Experimental Results

We evaluate the proposed method on the KTH [19] and YouTube [20] datasets. The KTH dataset contains 6 different actions, performed by 25 different subjects under four different settings. The Youtube dataset contains 11 different actions. The clips in the dataset were collected from the internet, and thus contain a large amount of variability, different viewpoints, scale changes and a significant amount of clutter.

We follow a leave-one-subject-out cross validation approach to classify the examples in the KTH dataset. At each fold, all examples performed by a single subject are retained for testing, and the rest are used for training. As can be seen from the confusion matrix of Fig. 3(a), there is significant confusion between *jogging* and *running*, which is expected, since the two classes are very similar. The average recall achieved is 90.36% and the average precision is 90.44%.

Localization examples are shown in Fig. 4. As can be seen, the activated features of the respective class are localized on body parts that are more characteristic of the actions, like the hands for the static (e.g. *boxing*) and the whole body for the non static actions (e.g. *jogging*). Apart from those, a small percentage of features of the other classes are also activated. For instance, *boxing* features are commonly activated around the hands of the *handclapping* sequences, due to the similarity of the hand motion in these two classes.

As can be seen in Table 1, the performance of the proposed method is superior to the one reported in [19], [4], [9] and similar to the one reported in [7], [22]. Compared to [16], the performance of our method is about 4% lower. This is due to a larger confusion between the static actions, attributed to the fact that several features that are selected correspond to primeval actions, like e.g. body parts moving into a certain direction, and therefore can be shared between different classes.

We follow a similar approach to classify examples in YouTube dataset. The classification protocol is provided by the database authors, and consists of 25 categories

	box	clap	wave	jog	run	walk
box	0.93	0.06	0.13	0.0	0.0	0.0
clap	0.09	0.91	0.0	0.0	0.0	0.0
wave	0.03	0.01	0.95	0.0	0.0	0.0
jog	0.0	0.0	0.0	0.80	0.14	0.06
run	0.0	0.0	0.0	0.06	0.93	0.01
walk	0.0	0.0	0.0	0.75	0.01	0.91

	basketball	biking	diving	golf swing	horse riding	soccer juggling	swing	tennis swing	trampoline jumping	volleyball spiking	walking
basketball	0.68	0.02	0.03	0.03	0.02	0.04	0.02	0.08	0.0	0.07	0.01
biking	0.04	0.72	0.03	0.0	0.07	0.03	0.04	0.0	0.03	0.02	0.01
diving	0.04	0.01	0.81	0.02	0.02	0.01	0.02	0.01	0.02	0.04	0.01
golf swing	0.02	0.0	0.01	0.73	0.01	0.05	0.03	0.05	0.02	0.03	0.04
horse riding	0.02	0.08	0.02	0.0	0.81	0.02	0.01	0.0	0.02	0.0	0.02
soccer juggling	0.04	0.02	0.04	0.05	0.02	0.67	0.04	0.03	0.04	0.02	0.02
swing	0.01	0.06	0.04	0.01	0.01	0.06	0.75	0.0	0.05	0.0	0.01
tennis swing	0.06	0.01	0.04	0.04	0.0	0.02	0.01	0.77	0.02	0.04	0.0
trampoline jumping	0.01	0.02	0.02	0.01	0.02	0.02	0.07	0.01	0.78	0.01	0.0
volleyball spiking	0.12	0.03	0.11	0.05	0.01	0.0	0.01	0.01	0.04	0.60	0.01
walking	0.0	0.11	0.03	0.05	0.08	0.03	0.08	0.05	0.05	0.03	0.50

Fig. 3. Confusion matrix of KTH and YouTube datasets

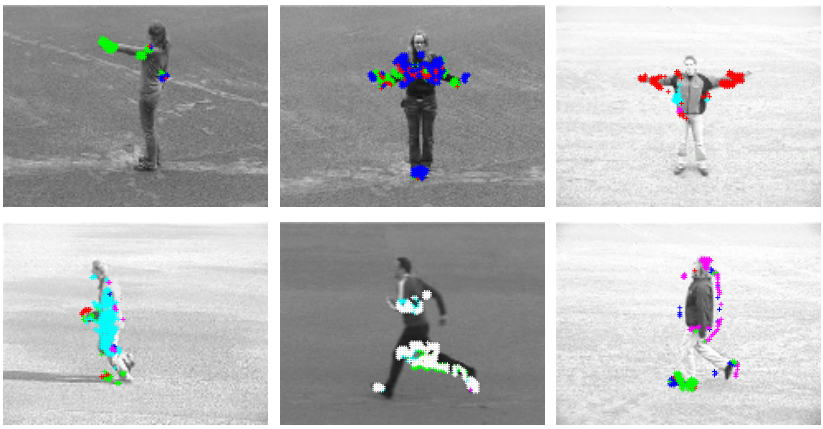

Fig. 4. Localization results from the KTH dataset. green: *boxing*, blue: *handclapping*, red: *handwaving*, cyan: *jogging*, white: *running*, magenta: *walking*.

per class, depending on subject, background type, or the photographer that captured the sequences. During testing, one set is retained for evaluation and the rest are used for training. As can be seen from Fig. 3(b), there are mutual confusions between *basketball*, *volleyball*, and *diving*, since these actions include some form of jumping. In general there are small confusions between the majority of classes, but this is expected due to the challenging nature of the dataset. The average recall achieved is 71.2% and the average precision is 71.5%.

Localization results for *basketball*, *tennis swing* and *horse riding* are shown in Fig. 5. As can be seen, the detected features are localized on the subjects that perform the action to which the features correspond to. False matches due to noisy background and significant camera motion are also evident.

As can be seen in Table 2 the performance of the proposed method is superior or similar to the one reported in [20],[23]. Wang et al. [23] report an accuracy of 84.2%,

Table 1. Comparisons of the proposed method to various methods proposed elsewhere for the KTH dataset

Methods	Accuracy (%)
Our method	90.36
Gilbert et al. [16]	95.7
Schuldt et al. [19]	71.83
Dollar et al. [4]	81.17
Jhuang et al. [7]	91.7
Ali and Shah [9]	87.7
Laptev et al. [22]	91.8

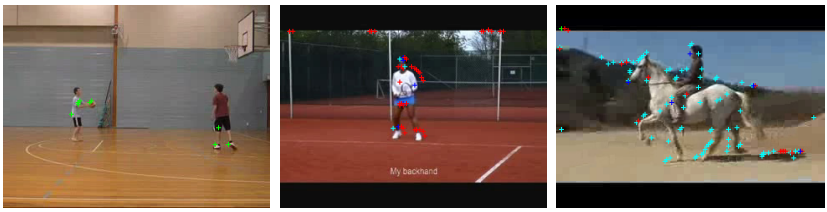

Fig. 5. Localization results from the YouTube dataset. green: *basketball*, red: *tennins swing*, cyan: *horse riding*.

Table 2. Comparisons of the proposed method to various methods proposed elsewhere for the YouTube dataset

Methods	Accuracy (%)
Our method	71.2
Liu et al. [20]	71.2
H.Wang et al. [23]	84.2
L.Wang et al. [17]	67.2
Q.V.Le et al. [24]	75.8
N.Ikizler-Cinbis and S.Scarloff [25]	75.2

however, their method is based on tracking instead of local features. Furthermore, the work in [25] performs stabilization to limit feature detection in the foreground. By contrast, our method does not perform motion compensation, stabilization or tracking.

6 Conclusions

In this paper we presented a hierarchical data mining method for human action recognition. The proposed method selects features of increasing complexity as the number of levels increases. Data mining allows us to select a small number of discriminative features out of a huge initial feature set. The followed mining approach offers an advantage compared to alternative feature selection methods in terms of simplicity and

efficiency. The method was tested on two datasets of human actions, achieving good results. Furthermore, despite of not explicitly formulating a detection mechanism, the proposed method is able to perform localization by back-projecting activated features onto the image plane.

Acknowledgments. This work has been funded in part by the European Community's 7th Framework Programme [FP7/20072013] under the grant agreement no 231287 (SSPNet). The work of Maja Pantic is funded in part by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB).

References

1. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. Comp. Vision, and Image Understanding 115, 224–241 (2011)
2. Laptev, I., Lindeberg, T.: Space-time Interest Points. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 432–439 (2003)
3. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
4. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS, pp. 65– 72 (2005)
5. Bregonzio, M., Gong, S., Xiang, T.: Recognising action as clouds of space-time interest points. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2009)
6. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. Comp. Vision, and Image Understanding 110, 346–359 (2008)
7. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A Biologically Inspired System for Action Recognition. In: Proc. IEEE Int. Conf. Computer Vision, pp. 1–8 (2007)
8. Schindler, K., Gool, L.V.: Action snippets: How many frames does human action require? In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
9. Ali, S., Shah, M.: Human action recognition in videos using kinematic features and multiple instance learning. IEEE Trans. Pattern Analysis and Machine Intelligence (2010)
10. Gilbert, A., Illingworth, J., Bowden, R.: Fast realistic multi-action recognition using mined dense spatio-temporal features. In: Proc. IEEE Int. Conf. Computer Vision (2009)
11. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
12. Seo, H., Milanfar, P.: Action recognition from one example. IEEE Trans. Pattern Analysis and Machine Intelligence 33, 867–882 (2011)
13. Amer, M., Todorovic, S.: Sum-product networks for modeling activities with stochastic structure. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1314–1321 (2012)
14. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Stanford University Technical Report (1993)
15. Quack, T., Ferrari, V., Leibe, B., Gool, L.V.: Efficient mining of frequent and distinctive feature configurations. In: Proc. IEEE Int. Conf. Computer Vision (2007)
16. Gilbert, A., Illingworth, J., Bowden, R.: Action recognition using mined hierarchical compound features. IEEE Trans. Pattern Analysis and Machine Intelligence 33, 883–897 (2011)
17. Wang, L., Wang, Y., Jiang, T., Gao, W.: Instantly telling what happens in a video sequence using simple features. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3257–3264 (2011)

18. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: ACM SIGKDD, pp. 43–52 (2004)
19. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. In: IEEE Conf. on Computer Vision and Pattern Recognition, vol. 3, pp. 32–36 (2004)
20. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos ”in the wild”. In: IEEE Conf. on Computer Vision and Pattern Recognition (2009)
21. Seo, H., Milanfar, P.: Training-free, generic object detection using locally adaptive regression kernels. IEEE Trans. Pattern Analysis and Machine Intelligence 32, 1688–1704 (2010)
22. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
23. Wang, H., Klaeser, A., Schmid, C., Liu, C.: Action recognition by dense trajectories. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3169–3176 (2011)
24. Le, Q., Zou, W., Yeung, S., Ng, A.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3361–3368 (2011)
25. Ikizler-Cinbis, N., Sclaroff, S.: Object, scene and actions: Combining multiple features for human action recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 494–507. Springer, Heidelberg (2010)