Sparsifying Dense Features for Action Classification

Debaditya Roy
Visual Intelligence and Learning Group
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
cs13p1001@iith.ac.in

M. Srinivas
Visual Intelligence and Learning Group
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
cs10p002@iith.ac.in

C. Krishna Mohan
Visual Intelligence and Learning Group
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
ckm@iith.ac.in

ABSTRACT

We propose an approach for sparse representation of dense features for action classification. Sparse representation has already been shown in literature as a good approximation for signals for various computer vision applications. This property is leveraged to represent a dense feature like action bank in the form of sparse dictionaries. These dictionaries are learnt using on-line dictionary learning (ODL) which further facilitates incorporating new training examples into existing dictionaries for more robust representation of various categories of action as and when required. Evaluation of the proposed method on realistic action datasets like UCF50 and HMDB51 shows that considering sparse representation of a dense feature is more suitable for classification than the feature itself.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications—*Computer Vision*

Keywords

Action Recognition, Dictionary Learning, Sparse Representation

1. INTRODUCTION

Action classification is an important tool for comprehensive image and video understanding. Diverse applications like automated video indexing of huge on-line video repositories like Youtube, Vimeo etc. to video surveillance systems in public places can benefit from adapting action classification approach for video classification. Actions are single-person activities such as "walking", "waving" and "punching". Interactions are human activities that involve two or more persons and/or objects. The goal of human activity classification is to automatically analyze ongoing activities from an unknown video. In the case where the video contains only one distinct human action, the main task is to classify the video into one of the different categories of action.

PerMIn '15 February 26 - 27 2015, Kolkata, West Bengal, India

http://dx.doi.org/10.1145/2708463.2709047.

Schuldt et al. [23] introduce the KTH dataset which consists of six action categories. Local space-time features are considered along with Support Vector Machine (SVM) for classification. In [7], Kläser et al. present histogram of oriented 3D spatio-temporal gradients which is essentially a collection of quantized 2D histograms collected from each frame of the video. After extracting the histogram gradient features from the KTH dataset an SVM classifier is applied for categorization of the videos. In [22], Savarese et al. use a global information descriptor known as spatial-temporal correlograms to encode flexible long range temporal information into the spatial-temporal motion features. Spatial-temporal correlograms are extracted from the KTH dataset and classification into appropriate classes is done using an unsupervised generative model.

In [9], Kovashka and Grauman propose the use of local motion and appearance features to develop a visual vocabulary and then form candidate neighborhoods consisting of the words associated with nearby points and their orientation with respect to the central interest point. The vocabularies are then combined using multiple kernel learning method to determine the best possible discriminative means of comparing videos. The neighborhoods obtained are then applied to a multi-channel generalized Gaussian kernel SVM to separate the action classes in KTH .

In [28], Wu et al. present the global Gaussian mixture model (GMM) for representing all the videos using the relative coordinate features from all the training videos where each video is viewed as the normalized parameters of a video-specific GMM adapted from the global GMM. The learning algorithm used is machine kernel learning with augmented features for classification of action classes in KTH. Kuehne et al. [10] introduce the HMDB51 dataset for action classification. Features such as HOG, HOF and C2 are extracted from the videos of all the 51 classes. An radial basis function kernel SVM is used for classification of these classes. Sadanand et al. [21] propose creation of an "Action bank" of videos, which in combination with a linear SVM classifier is used to classify the KTH and HMDB51 dataset.

According to Kliper-Gross et al. [8], calculating motion interchange patterns, based upon the characterization of change of one motion leading to another, is the best way to describe a distinct action. In this method, $N(=51)$ linear SVMs are used for one-vs-all classification of all the classes in UCF50 and HMDB51 using motion interchange patterns. Solmaz et al. [25] present the idea of gist, a global video descriptor which essentially computes the 3-D DFT of a given video clip using 68 3-D Gabor filters placed in 37 and 31 orientations. Then a linear SVM is used for classification of UCF50 and HMDB51 datasets.

A trajectory based local descriptor TrajMF is proposed by Jiang et al. [6] which works on top of local feature descriptors like HOG, HOF etc. and captures global and local reference points to characterize motion information. To classify the videos in the HMDB51 dataset, a χ^2 kernel SVM is used. Wang et al. [26] employ the idea of dense trajectories by estimating human motion using SURF for each frame of video. Further improvement in this method results from more accurate camera motion estimation and removing inconsistent matches due to humans. A radial basis function-χ^2 SVM is used for categorization of UCF50 and HMDB51 datasets. In [27], Wu et al. denote each action class as an event and assign a latent variable to it. The crucial motion patterns in each event are then captured using latent models. These latent models are then applied to three different classfiers - latent structural SVMs, max-margin hidden conditional random fields and latent SVMs, separately to classify the action categories in HMDB51.

Shi et al. [24] introduce sampling strategies for real-time action classification where each action is sampled using variable sized grids with increasing granularity to capture HOG3D, HOG, HOF and Motion Boundary Histogram (MBH) features from action videos. Dense and random sampling strategies are applied on KTH, UCF50 and HMDB51 datasets and all the descriptors are combined to obtain effective action classification on these datasets. Murthy et al. [15] extends on the work of Wang et al. [26] by removing the trajectories of background clutter and define the remaining as ordered trajectories. This results in reduced number of trajectories per video clip which are also combined from various scales in space to produce the best representation.

The motivation for this work lies in the fact that the degree of freedom of the various body parts of a person paritcipating in an action limits the total number of independent directions which are required for describing a given action. This means that even a dense high dimensional feature representing an action can represented by a sparse vector while still maintaining discriminative information. Since, sparse dictionaries have been shown to be suitable for modeling high dimensional dense features with relatively low information loss it is adapted for use. For large video databases like UCF50 and HMDB51, the number of training instances for each class is quite high and whenver, new training instances are added recomputation of dictionary is computationlly expensive. ODL builds the dictionary, one training example at a time and thus can handle large number of training instances with lower computation cost as compared to other dictionary learning methods.

2. RELATED WORK

In the field of sparse modeling of human actions, there have been a few developments recently. Qiu et al. [17] propose a probabilistic approach for learning sparse dictionaries for action attributes by maximizing mutual information between the learned and unlearned attributes. The method is also used for recognizing unknown actions on the Weizmann[2] and UCF Sports dataset[19]. Peng et al. [16] present a joint evaluation of feature encoding and dictionary learning techniques for action recognition. Sparse coding is shown to be effective as an encoding method on feature descriptors like Histogram of Oriented Gradients (HOG) and Histogram of Oriented Optical Flow (HOOF) and is also found to give reasonable performance dictionary learning methods. However, it was also shown that sparse coding takes significantly more computation time than other encoding methods on HOG and HOOF descriptors. So, instead a dense feature describing the whole video as a single feature vector viz. action bank was chosen to form sparse dictionaries.

In [4] sparse modeling of motion imagery is carried out to form inter-class and class-specific dictionaries to classify actions. However, the method is not scalable to a large number of classes which means action datasets like UCF50 and HMDB51 cannot be addressed. So, using ODL for forming dictionaries cubrs this problem as we can incrementally learn dictionaries even from large training datasets with large number of classes.

2.1 Action Bank Representation of Videos

Action bank is a high-level representation of videos proposed by Sadanand and Corso [21]. This representation of videos is achieved by applying 73 action detectors on a video clip. There are 205 action templates having an average spatial resolution of approximately 50×120 pixels and a temporal length of $40 - 50$ frames contributing to a $14965 \times 1(73 \times 205)$ feature vector of each video clip under consideration. The templates perform classification by detection and give a global description of videos. Action bank produces a single feature vector for an entire video clip which is quite large (14965×1) as compared to the number of video clips per class in any of the standard datasets (≈ 100). The resultant matrix is a "fat" matrix (14965×100) giving an under-determined system where the coefficients that are to be calculated (14965) are much more than the number of equations (100). This is the classic setting under which any sparse dictionary is formed and hence, action bank is a natural choice. Another alternative is to concatenate features obtained from consecutive frames into a single feature vector. However, in such a case there is high correlation between the feature dimensions and the resulting dictionary atoms will not contain very distinctive information.

2.2 K-SVD

Given a set of vectors $\{\mathbf{v}_i\}_{i=1}^n$, the K-SVD based dictionary learning method [1] finds the dictionary D by solving the following optimization problem:

$$(\hat{D}, \hat{\Phi}) = \arg\min_{D,\Phi} \| V - D\Phi \|_F^2 \; subject\ to \;\; \|\gamma_i\|_0 \leq T_0 \forall i,$$

where γ_i represents i^{th} column of Φ , V is the matrix whose

columns are v_i, and T_0 is the sparsity parameter. Φ represents sparse representation vectors. Here, $\|A\|_F$ denotes the Frobenius norm which is defined as $\|A\|_F = \sqrt{\sum_{ij} A^2{}_{ij}}$. The K-SVD algorithm alternates between sparse coding and dictionary update steps.

2.3 On-line Dictionary Learning

On-line dictionary learning, which an adaptive version of dictionary learning is proposed by Mairal et al. [14]. The sparse stage in ODL is a Cholesky-based implementation of LARS-lasso algorithm which is the same as K-SVD. However, in the dictionary update stage, block coordinate descent is used whcih does not require learning rate tuning and learns one example at a time giving the online nature akin to on-line stochastic approximations algorithms. Moreover, the dictionary at any time instant t D_t is calculated with D_{t-1} used as a warm restart.

$$\hat{\Phi}_t = \arg \min_{D_{t-1},\Phi} \| V - D\Phi\|_1 + +\lambda\|\Phi_i\|_1$$

$$\hat{D}_t = \arg \min_{\boldsymbol{D}\in\boldsymbol{C}} \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}\| \boldsymbol{V} - \boldsymbol{D}\Phi_{t-1}\|_2^2 + \lambda\|\Phi_i\|_1$$

C determines the number of action classes considered. Further improvements are made using mini-batch training, purging the dictionary of unused atoms and only handling fixed-size datasets.

3. PROPOSED METHOD

The proposed classification scheme consists of two phases - training and testing. In the training phase, dictionaries are constructed for each class and then combined to form a single dictionary using on-line dictionary learning (ODL). Testing phase comprises of computing the sparsity of a test clip with the dictionaries of each class using the ℓ_1-lasso distance. The class assigned to the video is the one having maximum sparsity for the given test clip i.e. minimum ℓ_1-lasso distance. This process is explained in figure 1.

3.1 Sparsity Based Classification

Sparse linear combination of training data acquired from a dictionary constitutes the representation of test data in the proposed sparsity based classification scheme using ODL. Class $C = [C_1, C_2, ..., C_N]$ consisting of training samples (action bank features) available for the given N classes is constructed. The samples belonging to the same class C_i lie approximately close to each other in a low-dimensional subspace. Let the p^{th} class have K_p training samples and the total number of training samples is denoted by $\{y_i^N\}$ where $i = 1, 2, \ldots, K_i$ and $K_1, K_2, \ldots, K_N$ are training samples corresponding to classes $C_1, C_2, \ldots, C_N$.

Let b be a input vector belonging to the p^{th} class, then it is represented as a linear combination of the training samples belonging to class p.

$$b = D_p\Phi_p$$

where D_p is a $m \times K_p$ dictionary whose columns are the training samples in the p^{th} class and $\boldsymbol{\Phi}_p$ is a sparse vector for the same class. The two main steps involved in the proposed method are :

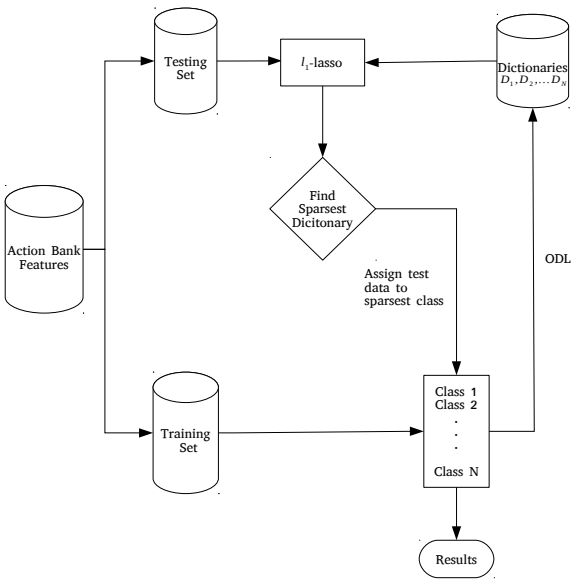

Figure 1: Flowchart of the proposed approach

1. *Dictionary Construction:* Construct the dictionary for each class of training features using ODL [14]. Then, the dictionaries $\boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_N]$ are computed using the equation.

$$(\hat{\boldsymbol{D}}_{\boldsymbol{i}}, \hat{\boldsymbol{\Phi}}_{\boldsymbol{i}}) = \arg \min_{\boldsymbol{D}_i,\Phi_i} \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\|\boldsymbol{C}_i - \boldsymbol{D}_i\Phi_i\|_2^2 + \lambda\|\Phi_i\|_1$$

$$\text{where } \boldsymbol{C}_i = \hat{\boldsymbol{D}}_i\hat{\boldsymbol{\Phi}}_i, \quad i = 1, 2, \ldots, N.$$

2. *Classification:* In the classification process, the sparse vector $\boldsymbol{\Phi}$ for given test feature is found in the test dataset $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l]$. Using the dictionaries of training samples $\boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_N]$, the sparse representation $\boldsymbol{\Phi}$ satisfying $\boldsymbol{D\Phi}$=$\boldsymbol{B}$ is obtained by solving the following optimization problem:

$$\Phi_j = \arg\min_{\Phi} \frac{1}{2}\|\boldsymbol{b_j} - \boldsymbol{D}\Phi_j\|_2^2$$
$$\text{subject to } \|\Phi_j\|_1 \leq T$$

and

$$\hat{i} = \arg\max_i \|\delta_i(\Phi_j)\|_1 \quad j = 1, \cdots, l$$

where $\boldsymbol{\delta_i}$ is a characteristic function that selects the coefficients for class C_i, T represents the sparsity threshold and l is the number of testing samples. A test clip $\boldsymbol{b_j}$ is assigned to class C_i if the absolute sum of sparsity coefficients associated with the i^{th} dictionary is maximum.

4. RESULTS AND DISCUSSION

In this section, we examine the efficacy of the proposed method on KTH, UCF50 and HMDB51 datasets.A detailed description of the datasets is presented below:

4.1 Datasets

KTH: The KTH dataset is a controlled dataset consisting of six human action classes, namely, walking, jogging, running, boxing, hand waving, and hand clapping. Each action is performed by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. The background is static and homogeneous. Since, there is no test-train split for KTH, from each class $\frac{2}{3}$ of the clips were chosen for training and $\frac{1}{3}$ were chosen for testing.

HMDB51: The HMDB51 dataset is a very large human action dataset containing 51 action categories, with at least 101 clips for each category. The dataset includes a total of 6,766 video clips extracted from Movies, the Prelinger archive, Internet, Youtube and Google videos. Such a variety of sources which have contributed to this database make it very realistic and challenging. To make the comparison with other classification techniques easy, the training and testing splits have been taken directly from the official website for HMDB51, wherein each split has 70 training and 30 testing clips for each category of action.

UCF50: The UCF50 dataset introduced by Reddy et al. [18] consists of 50 realistic human action categories. The 6000+ videos in the collection cover a variety of human actions which are collected mainly from youtube. The videos are divided into 50 classes with 25 groups in each class wherein each group is a single actor performing the action, being captured from different camera positions. The dataset was tested with leave one group out cross validation as mentioned in [18].

4.2 Results and Discussion

The experiments were conducted using action bank features on the three datasets described above with dictionary sizes of 60, 80, 100, 120, 140 and 160. The best performance observed for KTH, UCF50 and HMDB51 are 100%, 72.46% and 80.07%, respectively. The recorded performance is better than the state-of-the-art reported for KTH in literature [21]. In case of HMDB51, the accuracy of the proposed is significantly higher than the state-of-the-art [26]. This can be attributed to sparse representation which captures the distinct signatures i.e. sparsities, for each of the action classes and dictionary learning which preserves the essence of human action in the dictionaries learned using the training samples.

Tables 1, 2 and 3 enumerate the results of various existing classification schemes applied on KTH, UCF50 and HMDB51, respectively, in comparison to the proposed method. In case of KTH, as shown in table 1, the proposed method gives better performance than the method in [21] where the action bank features are used in conjunction with SVM. In table 2, it can be observed that the methods proposed in [24] and [15] use a combination of different features like HOG, HOF, MBH etc. to give better classification performance than the methods which use one feature, say action bank. However, in table 3, it can be observed that the development of the dictionaries leads to only a few misclassified examples in the whole HMDB51 dataset as compared to both ensemble methods in [24] and [15].

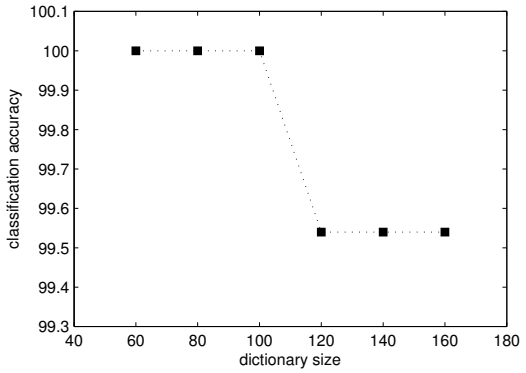

Figure 2: KTH : classification accuracy vs. dictionary size

The method proposed in [21] uses SVM as a classifier for action bank features. Whereas, we use sparse representation of action bank features which proves to be more suitable than directly applying SVM on action bank features. Action bank captures all the motion information in the video clip in 14965 coefficients where only similar actions will have significant values. This means that the feature is sparse to start with and hence, can be sparsely coded to greater effect. Sparsity preserves only the essential coefficients which capture the action information generated by the actor in the scene and removes the background information from the dictionary.

4.3 Comparison of performance based on dictionary size

Figures 2, 3 and 4 portray the variation of classification accuracy in terms of dictionary size for KTH, UCF50 and HMDB51 datasets, respectively. In case of KTH, the highest classification rate is observed for dictionary sizes of 60, 80 and 100 with sparsity set at 20 for each. For HMDB51, the maximum performance is noted for dictionary size of 100 with sparsity set at 2, after which the performance degrades when dictionary size increases. In case of UCF50, ODL demonstrates most suitable representation leading to best classification for dictionary size of 120 with sparsity set at 8. These results show that it is not possible to set sparsity to any specific value for all datasets even if they correspond to the same semantic groups i.e. actions in our case.

The performance, as it can be seen from the figures 2, 3 and 4 generally remains constant and deteriorates after a certain dictionary size for all the datasets considered. This property can be used to design the optimal dictionary size by considering performance in terms of accuracy depending upon the application in question. However, since the size of dictionary where the performance saturates is not even same for the three datasets which are related in terms of content, it is difficult to pinpoint the optimal dictionary size in advance.

5. CONCLUSION

We have presented an approach for sparse representation

Table 1: Comparison of classification accuracy on the KTH dataset

Method	Features	Accuracy (%)
Schuldt et al. [23]	Local space-time features	71.7
Kläser et al. [7]	Spatio-temporal descriptor	84.3
Savarese et al. [22]	Spatial-temporal correlograms	86.8
Ryoo et al. [20]	Spatio-temporal local features	91.1
Liu et al. [12]	Latent attributes	91.6
Bregonizo et al. [3]	Space-time interest points	93.2
Liu et al. [12]	Latent attributes	93.8
Le et al. [11]	Invariant spatio-temporal features	93.9
Liu and Shah [13]	Spatio-temporal	94.3
Gilbert et al. [5]	Space-time features	94.5
Kovashka et al. [9]	Space-time feature neighborhood	94.5
Wu et al. [28]	Context and Appearance distribution features	94.5
Sadanand et al. [21]	Action bank	98.2
Proposed	Action bank	**100**

Table 2: Comparison of classification accuracy on the UCF50 dataset. (Only the results with leave-one-group-out validation are considered for comparison)

Method	Features	Accuracy (%)
Kliper-Gross et al. [8]	Motion Interchange Patterns	72.6
Solmaz et al. [25]	Global video descriptor	73.7
Reddy and Shah [18]	MBH	76.9
Shi et al. [24]	HOG + HOF + HOG3D + MBH	83.3
Murthy et al. [15]	Trajshape + MBH + HOG + HOF	**85.5**
Sadanand et al. [21]	Action bank	57.9
Proposed	Action bank	72.46

Table 3: Comparison of classification accuracy on the HMDB51 dataset. (Only the results which follow three-fold validation as mentioned by the authors have been considered for comparison)

Method	Features	Accuracy (%)
Kuehne et al. [10]	HOG/HOF	20.20
Kuehne et al. [10]	C2	23.18
Kliper-Gross et al. [8]	Motion Interchange Patterns	29.17
Solmaz et al. [25]	Global video descriptor	29.20
Jiang et al. [6]	Trajectory + Motion reference points	40.70
Wang et al. [26]	Dense Trajectory	44.75
Murthy et al. [15]	Trajshape + MBH + HOG + HOF	47.3
Shi et al. [24]	HOG + HOF + HOG3D + MBH	47.6
Wu et al. [27]	Multi-level features	49.46
Wang et al. [26]	Improved Dense Trajectory	57.20
Sadanand et al. [21]	Action bank	26.90
Proposed	Action bank	**80.07**

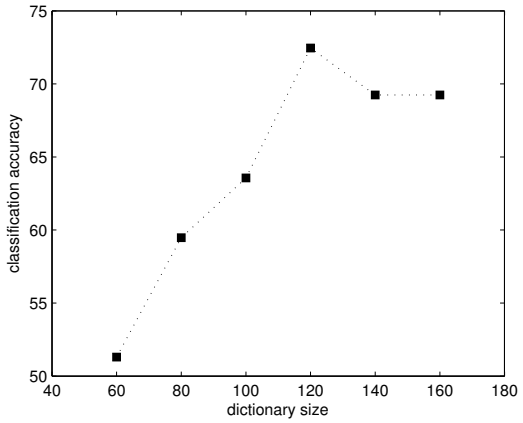

Figure 3: UCF50 : classification accuracy vs. dictionary size

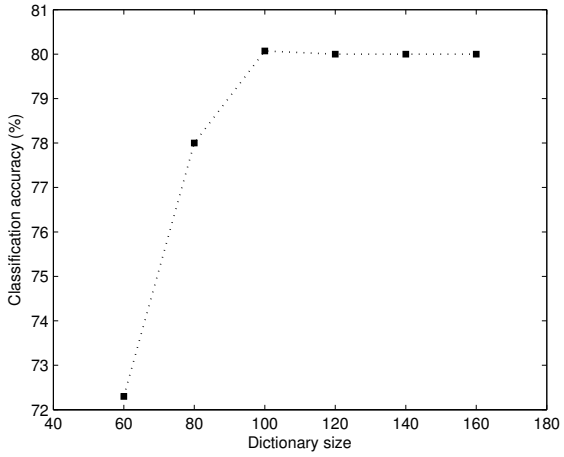

Figure 4: HMDB51 : classification accuracy vs. dictionary size

of dense features for action classification using on-line dictionary learning. The approach was demonstrated using sparse representation of a dense feature like action bank. Sparse represenation of action bank as compared to naive action bank features shows better classification performance under same conditions on three action datasets namely, KTH, UCF50 and HMDB51. In future, ODL can be utilized for efficient implementation of action classification systems because of its ability to incorporate new training data into existing dictionaries. Also, since the method produces scalable dictionaries it can be used for a large number of classes without losing discriminatory information. Since, action bank was just used to demonstrate the effectiveness of sparsity, any other dense features like 3DHOG and 3DSIFT can also be explored to the same effect.

6. ACKNOWLEDGEMENT

We would like to thank Mr. Jason Corso for making the action bank features available for KTH, UCF50 and HMDB51 datasets.

7. REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. k -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, Nov 2006.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1395–1402 Vol. 2, Oct 2005.

[3] M. Bregonzio, S. Gong, and T. Xiang. Recognising action as clouds of space-time interest points. In *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*, pages 1948–1955, Jun. 2009.

[4] A. Castrodad and G. Sapiro. Sparse modeling of human actions from motion imagery. *International Journal of Computer Vision*, 100(1):1–15, 2012.

[5] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *Computer Vision (ICCV). IEEE International Conference on*, pages 925–931, Oct. 2009.

[6] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *Computer Vision (ECCV). European Conference on*, Oct. 2012.

[7] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference (BMVC)*, pages 275:1–10, May. 2008.

[8] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *Computer Vision (ECCV). European Conference on*, pages 425–438, Oct. 2012.

[9] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*, pages 2046–2053, Jun. 2010.

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Computer Vision (ICCV), IEEE International Conference on*, pages 2556–2563, Nov. 2011.

[11] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 3361–3368, Jun. 2011.

[12] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 3337–3344, Jun. 2011.

[13] J. Liu and M. Shah. Learning human actions via information maximization. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1–8, Jun. 2008.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 689–696, New York, NY, USA, Jun. 2009. ACM.

[15] O. Murthy and R. Goecke. Ordered trajectories for large scale human action recognition. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 412–419, Dec 2013.

[16] X. Peng, L. Wang, Y. Qiao, and Q. Peng. A joint evaluation of dictionary learning and feature encoding for action recognition. In *ICPR*, 2014.

[17] Q. Qiu, Z. Jiang, and R. Chellappa. Sparse dictionary-based representation and recognition of action attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 707–714, Nov 2011.

[18] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[19] M. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2008. IEEE Conference on*, pages 1–8, Jun. 2008.

[20] M. S. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Computer Vision (ICCV). IEEE International Conference on*, pages 1593–1600, Oct. 2009.

[21] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*, pages 1234–1241, Jun. 2012.

[22] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei. Spatial-temporal correlatons for unsupervised action classification. In *Proceedings of the 2008 IEEE Workshop on Motion and Video Computing (WMVC)*, pages 1–8, Washington, DC, USA, Jan. 2008. IEEE Computer Society.

[23] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition (ICPR), 17th International Conference on Volume 3*, pages 32–36, Washington, DC, USA, Aug. 2004. IEEE Computer Society.

[24] F. Shi, E. Petriu, and R. Laganiere. Sampling strategies for real-time action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2595–2602, June 2013.

[25] B. Solmaz, S. Assari, and M. Shah. Classifying web videos using a global video descriptor. *Machine Vision and Applications*, 24(7):1473–1485, 2013.

[26] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Computer Vision (ICCV). International Conference on*, Sydney, Australia, Oct 2013.

[27] J. Wu and D. Hu. Learning effective event models to recognize a large number of human actions. *Multimedia, IEEE Transactions on*, 16(1):147–158, 2014.

[28] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 489–496, Jun. 2011.