2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops

Formulating Action Recognition as a Ranking Problem

Ethem F. Can and R. Manmatha
School of Computer Science, University of Massachusetts
Amherst, MA, 01003, USA
[efcan, manmatha]@cs.umass.edu

Abstract

Action recognition is one of the major challenges of computer vision. Several approaches have been proposed using different descriptors and multi-class models. In this paper, we focus on binary ranking models for the action recognition problem and address the action recognition as a ranking problem. A binary ranking model is trained for each action and used to recognize the test videos for that action. Binary ranking models are constructed using dense SIFT (DSIFT) descriptors and histogram of oriented gradients / histogram of optical flows (HOG/HOF) descriptors. We show that using ranking models, it is possible to obtain higher recognition accuracies from a baseline that is based on multi-class models on the very recent and challenging benchmark datasets; Human Motion Database (HMDB) and The Action Similarity Labeling (ASLAN).

1. Introduction

Action recognition is the problem of classifying unlabeled videos into a predefined set of actions. For each action a set of training videos is provided to train a classifier. Recent work has focused on datasets containing unconstrained videos such as You-tube videos available online since they are much closer to real-word cases. One such dataset is the HMDB [10] which is considered to be more varied and complicated. Recently another benchmark dataset, ASLAN, for action similarity was released [8, 9].

Here we focus on binary ranking models rather than multi-class models, and perform recognition depending upon the outputs of such ranking models. We also provide two decision functions; 1) rank-pooling where we focus on the ranks of the test videos for recognition, and 2) score-pooling where we focus on the responses of the ranking models for recognition. Although multi-class models have shown to be highly promising, we show that binary ranking models with our decision functions provide better discriminative functions than multi-class models. The reason is that ranking models are more robust for handling an unbalanced number of positive examples (in the one-against-all case). The decision functions used in this study are better in terms of accuracy than the decision function of the one-against-one multi-class model that is based on a majority voting approach. Our binary ranking models can also be considered as weak-learners since we create a number of ranking models -one ranking model for each action class- and use the outputs and ranks of the models for recognition purposes. Experimental results show that ranking models with our decision functions for the action recognition problem outperform existing multi-class models in terms of recognition accuracy on the HMDB and the ASLAN datasets using the same descriptors: dense SIFT (DSIFT) and HOG/HOF using spatial pyramid representations.

ASLAN is a large dataset of actions where the task is to judge whether two videos have similar actions. Much more common is the action recognition task which involves determining whether a particular video is part of an action class X. We format the ASLAN dataset into 3 splits to make it useful for action recognition. Since we do 3-fold cross validation, all action classes with less than 3 videos are removed. Such a dataset is useful for many reasons: 1) The remaining dataset still has more actions (264) than any existing action recognition dataset. 2) Unlike previous datasets such as HMDB many of the classes are unbalanced (fewer positive samples for each action) therefore this provides a challenging dataset for our algorithms to be tested on 3) The action classes are very close to each other. Keeping these motivations in mind, we provide recognition results of the ASLAN dataset so that it might serve as a baseline for future work on action recognition.

The main contributions of this paper are as follows. First, we formulate action recognition as a ranking problem and focus on binary ranking models rather than multi-class models. We provide two decision functions; rank-pooling and score-pooling, for our ranking models. We show that our ranking models with these decision functions outperform a baseline which is based on multi-class models. Second, we provide an extensive comparison of different models with different settings and analyze them

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.44

on the recent benchmark datasets; HMDB and ASLAN. We obtain a recognition accuracy for the HMDB dataset that is higher than most of the previously reported numbers [12, 15, 17, 9]. Last but not least, we provide the baseline results for the ASLAN dataset when used as a recognition benchmark dataset.

2. Related Work

Action recognition has been studied for several years. Temporal information based features are commonly used for this problem in different forms. The combination of histogram of oriented gradients (HOG) and histogram of optical flow (HOF) has been successfully used for action recognition [12]. Wang et al. [19] showed that HOG/HOF provides better performance than a number of different features such as HOG3D and HOF. The recent survey papers [14, 20] provide an extensive discussion of methods used for action recognition.

Sadanand and Corso [15], inspired by Objectbank, constructed action templates and used these templates to classify actions (know as Actionbank). Kilper-Gross et al. [9] focused on capturing local changes in motion orientations. Every pixel in every video frame was encoded by eight strings of eight trinary digits each. Using these encodings they detected the motion changes on consecutive frames. Sun et al. [18] focused on 2D and 3D SIFT descriptors computed on 2D SIFT interest points. However, they did not use a dense SIFT descriptor which has been shown to perform better in object recognition. Scovanner et al. [16] introduced a 3-dimensional SIFT descriptor and evaluated it for action recognition. Solmaz et al. [17] computed GIST3D descriptors based on GIST using 3D Gabor filters. To the best of our knowledge spatial pyramid versions of DSIFT and HOG/HOF have not been used together in action recognition studies.

There has been some attempt at combining rankings for classification [3]. Bucak et al. [1] focused on multi-label ranking applied to object recognition. Almost all of the recent previous work in action recognition used multi-class models -either one-against-one or one-against-all but particularly SVM for training and testing.

3. Problem Formulation

Previous work on action recognition [15, 17] has for the most part focused on a classification approach. Usually a multi-class SVM model is created from the training examples and the test data is classified using this model. Here, we formulate action recognition as a ranking problem using binary ranking models with two different decision functions. The motivation for using this approach is that, we claim that ranking models are more robust to the problem of unbalanced number of positive examples for the classes; moreover, using the responses and ranks of such models is better than using the majority voting technique. Experimental results prove our claim on these issues as presented in the results and discussion section (Section 6.).

For each action we create a ranking model. The videos belonging to an action are counted as relevant while creating the ranking model for that particular action and non-relevant otherwise. Then we use the responses and ranks from the models for recognition. We compare our models with different types of multi-class SVM such as one-against-one and one-against-all.

3.1. Multi-class Models

In this work, multi-class models (one-against-one and one-against-all) are used for comparison. In this section, we briefly describe the procedure used in those models.

One-against-all Approach: In the one-against-all multi-class SVM, k binary models are created, one for each class in the training set. The objective function for a training set of n examples and k classes is as follows:

$$\min \{\frac{1}{2} \sum_{i=1...k} w_i w_i + \frac{C}{n} \sum_{i=1...n} \xi_i\} \quad (1)$$

where C is the regularization parameter. Here the objective function considers all of the classes together in the training set.

One-against-one Approach: In the one-against-one multi-class SVM there are $\frac{k(k-1)}{2}$ two-class models i.e. one model for each pair of actions. Here the objective function for each two-class model is treated independently as opposed to the one-against-all approach. The decision function is based on a majority voting technique. An unlabeled video is tested against all the two-class models. Then, for each response, the unlabeled video gets a vote regarding the response of the model. For example, say model M is created for class A and B and for a given test video, if the output is positive then class A gets the vote, and class B otherwise. The final decision is made by considering the class having the highest number of votes (majority vote).

3.2. Ranking Models

In our work we focus on two different types of ranking models. The first one is a binary ranking model (B_r) that is SVM-rank and the other one involves a modification of SVM binary classification for ranking (B_m). Even though a binary SVM classifier is designed for recognition rather than ranking, we can use it as a ranking model as well if we consider the prediction score, $wx + b$, rather than the sign of it, $sgn(wx + b)$. Therefore, the examples can be ranked according to their $wx + b$ scores.

The one-against-all multi-class model we focus on in this work considers each class together while optimizing the objective function that is defined to be

$\min \frac{1}{2} \sum_{i=1...k} w_i w_i + \frac{C}{n} \sum_{i=1...n} \xi_i$ where k is the number of classes and n is the number of samples in the training set (see Equation 1).

However, in B_m, a model is created for each class independent of other classes in the training set. In the contrary, the objective function of the one-against-all multi-class model depends on all the models together which is defined to be $\min \frac{1}{2} ww + C \sum_{i=1...n} \xi_i$. B_m is different than an one-against-one multi-class model because B_m uses score-pool and rank-pool decision functions; whereas one-against-one multi-class model uses a majority voting based decision function.

For the binary ranking model B_r the aim is to ensure that videos belonging to the target class (relevant videos) are more likely to be ranked higher than videos belonging to other classes (non-relevant videos). SVM-rank creates a model based on pair-wise comparisons of the training examples. Joachims [7] showed that this ranking problem can be formulated by maximizing the number of the following inequalities satisfied;

$$\forall (v_i, v_j) : wx_i > wx_j \tag{2}$$

where x is the feature vector, w is the weight vector, v_i is a relevant and v_j is a non-relevant example video. Non-negative slack variables (ξ) are employed to solve the optimization problem by transforming inequalities to equalities in a similar way to SVM classification. SVM-rank produces prediction scores which are used to rank the videos. It requires pairwise preferences for training. In our case we assume that any video (in the training set) which is relevant to a particular action is ranked higher than a video which is not relevant to the action (one model is created for each action class). We do not impose a ranking between two videos which are both relevant to the action class or are both non-relevant to the query since there is no way of imposing an order in such situations.

In order to formulate the action recognition problem using ranking models, we first create individual models for each action class. For an action class i, an example video is counted as relevant if that video is labeled as action i. Example videos that are labeled with other actions j where $i \neq j$ are counted as non-relevant for that particular action class i. If there are n action classes $1, ..., n$, then we have n binary ranking models $m_1, ..., m_n$; one model for each action class. Consider an example video in training set v labeled as "walking", then video v is relevant to the ranking model for action class k assuming "walking" is the k^{th} action class and is non-relevant to every other action class.

3.3. Decision Functions

So far we described the training phase of the ranking models, here we explain the testing part. We formulate action recognition as a ranking problem. We perform the recognition based on multiple ranking models.

The recognition using ranking models in our work is performed using two decision functions; rank-pool and score-pool. For both models, B_m or B_r, we make use of these two decision functions and explain them below.

3.3.1 Score-Pooling

In the testing phase, we make use of ranking models to recognize the action of an unlabeled test video. Each test video is run against all action models. The score-pool decision function is as follows;

$$\begin{aligned} y &= i \ \ where \\ w_i x + b_i &= \max_{j=1,...,n} w_j x + b_j \end{aligned} \tag{3}$$

where y is the label, i is the class id, x is the feature vector of video v, and w_i is the weight vector for class i.

We recognize the action of a test example by comparing the outputs of the ranking models and assume that the test video belongs to the class that provides the maximum response among the ranking models.

3.3.2 Rank-Pooling

This decision function requires ranking the test videos according to their score. A higher rank means that a video is more relevant to the class whose model is considered. The rank-pool decision function can be summarized as follows;

$$\begin{aligned} y &= i \ \ where \\ r_i(v) &= \max_{j=1,...,n} r_j(v) \end{aligned} \tag{4}$$

where r_j is the rank of video v for the ranking model of the class j. Note that this decision function requires a number of videos available in the test bed. If there is only one video available in the test bed, a single test video gets always the first position in the ranked list. However, this case can be handled by sub-sampling a number of videos from the training set to construct a test bed consisting of more than one test example.

4. Descriptors

In this section, we explain the descriptors used for both multi-class and ranking models. We focus on two types of descriptors; densely sampled SIFT (DSIFT) histogram of oriented gradients / optical flow (PHOG/HOF). While PHOG/HOF exploits temporal information since it is computed on the space-time interest points, DSIFT exploits the information from densely sampled points. Raw DSIFT and PHOG/HOF descriptors are quantized into visual words. These visual words are then used for representation. Both

descriptors are based on the spatial pyramid histogram of words representation. Spatial pyramid representations have proven to be successful on a wide range of problems including object recognition in computer vision. In our work we employ three levels of a spatial pyramid (0, 1, and 2). There are 1, 4, and 16 regions for each level respectively. The final spatial pyramid representation is formed by concatenating the histograms for each region.

DSIFT Descriptors: We first subsample the video clips into video frames. For each video frame we densely sample SIFT descriptors. The step size for the extraction is set to 5 pixels. We use three scales (actual size, 50% of the actual size, and 25% of the actual size) in the process. We make use of hierarchical k-means to quantize the raw descriptors into visual words. We set k to 1,000 in the quantization process.

PHOG/HOF Descriptors: Even though we compute histogram of oriented gradients / histogram of optical flow (HOG/HOF) descriptors, we call them PHOG/HOF since we focus on a spatial pyramid representation unlike early attempts with the same descriptor [12]. In order to compute the PHOG/HOF descriptors, we first compute the Space-Time Interest Points (STIPs), then corresponding local space-time descriptors. HOG/HOF descriptors are computed on a 3D video patch in the neighborhood of each detected STIP [12, 11]. The grid setting for spatio-temporal blocks is 3x3x2. For oriented gradients there are 4-bins and for optical flow there are 5-bins. In total we have a HOG descriptor of size 72 and a HOF descriptor of size 90. HOG/HOF is a concatenation of HOG and HOF; therefore, the final size is 162. Clustering and quantization steps are exactly the same as for DSIFT. We again set the vocabulary size k to 1,000.

Normalization: Video clips in the dataset have different numbers of video frames and hence some kind of normalization is needed. We compute the histogram for a video clip by pooling the histogram values over the frames. This pooling is done by summing up the individual values of the histograms. The use of logarithm compresses the range of values reducing the length difference between videos. Then we apply the L_∞ normalization to each level in the spatial pyramid histogram. L_∞ is based on dividing the values in a histogram by its maximum value.

5. Experimental Settings

Here we first describe the datasets used in this study, then provide details of the experimental environment.

5.1. Datasets

We focus on two recent and challenging benchmark datasets -HMDB and ASLAN- to evaluate our system. HMDB consists of about 6700 unconstrained video clips. These clips are collected from the web and movies. There are 51 actions and each action has about 100 videos.

The other benchmark dataset we focus on is ASLAN [8]. Even though the main motivation of the ASLAN dataset is to evaluate action similarity rather than action recognition, we reformatted the splits of the dataset as described in Introduction to ensure that it can be used for evaluation of the action recognition problems. The total number of videos is more than 3000. We believe that the ASLAN dataset is a much more challenging dataset and a better benchmark for evaluating models for action recognition.

5.2. Experimental Environment

For ranking models B_m and B_r we make use of the implementation in [5] and add the efficient intersection kernel (IK) implementation proposed in [13]. For one-against-all multi-class SVM we use[6]. For one-against-one multi-class SVM we use Libsvm package [2]. For model parameters, we use the default settings; C=0.01 for the binary models and C=1 for the multi-class models. In the evaluation stage, for HMDB we use the training-test splits in [10]. They specify three splits each of which has 70 videos for training and 30 videos for testing for each action. For ASLAN we randomly split the data evenly into 3 folds and perform 3-fold cross-validation and will make the splits publicly available. The accuracies provided in this paper are averaged over the folds.

6. Experimental Results and Discussion

In this section we provide recognition accuracies for the action recognition problem on the HMDB and ASLAN datasets. We also show that multi-class model accuracies are outperformed by the ranking models.

HMDB: In Table 1 we provide the accuracy scores for the HMDB dataset. When we consider the individual performance of ranking models; B_m with IK kernel using score-pool provides the best recognition accuracy when DSIFT is used as descriptors. For PHOG/HOF descriptors, B_m with linear kernel using rank-pool provides the best recognition accuracy. IK does not improve the results significantly for PHOG/HOF case and it can be explained with the fact that PHOG/HOF histograms are much sparser than DSIFT histograms. When the histograms are very sparse then IK provides similar results as we obtain with a linear kernel. It is also important to note that DSIFT with ranking models using IK with both decision functions provides better accuracies than most of the previously reported action recognition methods (see Table 1 and 2).

When we consider the time to extract DSIFT versus previous methods, we can claim that DSIFT extraction is much more efficient than the previous methods. To be more specific, it takes one or two minutes to process 100 frames for DSIFT -without using GPU-; whereas, it takes 204 minutes

on the average to process a video in a dataset where the average length is 7.5 seconds for Actionbank [15].

Desc.	Dec. Func.	Model	Kernel	Accuracy
DSIFT	N/A	one-vs-all	lin.	23.16%
DSIFT	N/A	one-vs-one	lin.	23.94%
DSIFT	score-pool	B_m	lin.	26.27%
DSIFT	score-pool	B_r	lin.	17.89 %
DSIFT	rank-pool	B_m	lin.	**26.30** %
DSIFT	rank-pool	B_r	lin.	26.19 %
PHOG/HOF	N/A	one-vs-all	lin.	9.61%
PHOG/HOF	N/A	one-vs-one	lin.	23.69%
PHOG/HOF	score-pool	B_m	lin.	**26.18**%
PHOG/HOF	score-pool	B_r	lin.	22.86%
PHOG/HOF	rank-pool	B_m	lin.	24.35 %
PHOG/HOF	rank-pool	B_r	lin.	23.50 %
DSIFT	N/A	one-vs-one	IK	26.52%
DSIFT	score-pool	B_m	IK	**30.24**%
DSIFT	rank-pool	B_m	IK	29.67%
PHOG/HOF	N/A	one-vs-one	IK	22.67%
PHOG/HOF	score-pool	B_m	IK	**26.20**%
PHOG/HOF	rank-pool	B_m	IK	24.63 %

Table 1. Recognition accuracies on the HMDB set.

Note that we do not provide the accuracy scores for one-vs-all case and B_r case with IK kernel. Even though we tried the efficient implementation of IK, creating models takes much more time than for the other cases. Later we discuss the time we spent to create models in detail.

Even though our aim here is to show that the ranking models with two decision functions provide better accuracies than the multi-class models, we also compare our findings with the previous work on the HMDB dataset. On the recent attempts for action recognition, the most common approach is to fuse the outputs of different descriptors and/or classifiers. We also perform a very similar experiment where we fuse the outputs of DSIFT and PHOG/HOF models by taking the arithmetic mean of the scores.

System	Model	Kernel	Accuracy
HOG/HOF [12]	one-vs-all	χ^2	20.44%
ActionBank [15]	one-vs-all	lin.	26.90%
GIST3D+STIP[17]	one-vs-one	IK,lin.	29.20%
MIP [9]	one-vs-all	lin.	29.17%
TrajMF [4]	one-vs-all	lin., IK, χ^2	40.7 %
PHOG/HOF	rank-pool, B_m	lin.	26.30%
DSIFT	score-pool, B_m	IK	30.24%
DSIFT + PHOG/HOF	score-pool, B_m	IK	39.00%

Table 2. Comparison of the recognition accuracies of our methods with the previous work on the HMDB set. Last three rows are the recognition accuracies for our models.

In Table 2 we provide the recognition accuracies of our methods as well as the previous recognition accuracies for the HMDB set. We obtain a recognition accuracy of 39% when we fuse the descriptors. It is important to note that ranking models constructed using DSIFT and PHOG/HOF descriptors perform better than many of the previous methods [12, 15, 17, 9] and they are comparable to the state-of-the-art accuracy [4].

ASLAN: In Table 3 we provide the accuracies for the ASLAN dataset using the multi-class models as well as ranking models with score-pool and rank-pool. Our ranking models provide higher accuracies than the multi-class models as in the case of the HMDB dataset using the same set of descriptors. Here the difference is much more significant than the HMDB case. Perhaps it can be explained by the fact that, the number of relevant videos for each action class is uneven and the number of positive examples are much less than for the HMDB case. This means that multi-class models cannot create a good discriminative function over the action classes in this type of scenario. However our ranking models consider each action individually. These results highlight the performance difference of the ranking models in scenarios with fewer positive examples. These cases are highly likely in real-world applications; therefore, underlining the importance of using the ranking models with the score-pool and rank-pool decision functions.

Desc.	Dec. Func.	Model	Kernel	Accuracy
DSIFT	N/A	one-vs-all	lin.	9.63%
DSIFT	N/A	one-vs-one	lin.	5.87%
DSIFT	score-pool	B_m	lin.	20.00%
DSIFT	score-pool	B_r	lin.	**26.23** %
DSIFT	rank-pool	B_m	lin.	20.06 %
DSIFT	rank-pool	B_r	lin.	22.77 %
PHOG/HOF	N/A	one-vs-all	lin.	6.81%
PHOG/HOF	N/A	one-vs-one	lin.	4.11%
PHOG/HOF	score-pool	B_m	lin.	4.36%
PHOG/HOF	score-pool	B_r	lin.	**18.90**%
PHOG/HOF	rank-pool	B_m	lin.	8.25 %
PHOG/HOF	rank-pool	B_r	lin.	11.55 %
DSIFT	N/A	one-vs-one	IK	32.79%
DSIFT	score-pool	B_m	IK	**38.97**%
DSIFT	rank-pool	B_m	IK	28.59 %
PHOG/HOF	N/A	one-vs-one	IK	15.08%
PHOG/HOF	score-pool	B_m	IK	**22.76**%
PHOG/HOF	rank-pool	B_m	IK	12.42 %

Table 3. Recognition accuracies on ASLAN set.

For ASLAN, B_r outperforms B_m. The constraints for B_r are the pairs of examples -one relevant and one non-relevant- that are used to make sure that the relevant video is ranked higher than a non-relevant video. Therefore, when there are a few number of relevant examples B_r outperforms B_m.

7. Computation Cost Analysis

Here we discuss running time costs for each model. We use a recent linux box to perform the experiments and we use the same machine for all experiments. We run each experiment twice and provide the average of them to eliminate any caching overheads. In Table 4 we provide running times for different models for DSIFT and PHOG/HOF descriptors.

One-against-one model requires much more time than one-against-all model since it creates more models. The running time provided in the table for B_m and B_r is for

creating a model. Therefore, the final running time should be multiplied by the number of classes. Even though creating ranking models seems to be costly, this issue can be handled by parallelizing the processes. Each ranking model can be trained at the same time on different machines. In this way the final running time stays the same. DSIFT histograms requires much more time than PHOG/HOF histograms since PHOG/HOF histograms are sparser than DSIFT histograms. The average number of non-zero values for DSIFT histogram is about 5100; whereas, it is about 750 for PHOG/HOF histograms.

System	Model	Kernel	Seconds
DSIFT	one-vs-all	lin.	51.82
DSIFT	one-vs-one	lin.	819.51
DSIFT	one-vs-one	IK	681.09
DSIFT	B_r	lin.	39.20
DSIFT	B_m	lin.	11.01
DSIFT	B_m	IK	45.71
PHOG/HOF	one-vs-all	lin.	8.97
PHOG/HOF	one-vs-one	lin.	57.56
PHOG/HOF	one-vs-one	IK	65.35
PHOG/HOF	B_r	lin.	1.98
PHOG/HOF	B_m	lin.	1.35
PHOG/HOF	B_m	IK.	2.85

Table 4. Computation costs for the ASLAN dataset.

8. Conclusions

In this work we formulate action recognition as a ranking problem and make use of ranking models with two decision functions; score-pool and rank-pool. We create ranking models for each action class in a dataset. For a test video, action recognition is performed using the scores and ranks of the binary models. We show that our ranking models with two decision functions perform better than a baseline that uses multi-class models on the benchmark datasets; HMDB and ASLAN using the same set of descriptors; DSIFT and PHOG/HOF. In this work we also reformat the ASLAN dataset so that it might serve for evaluation of the action recognition methods. We believe that the ASLAN dataset is a very challenging benchmark dataset for the action recognition problem because of the uneven number of positive examples, small number of training examples, the large number of action classes, and the closeness of the action classes.

9. Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

[1] S. Bucak, P. Mallapragada, R. Jin, and A. K. Jain. Efficient multi-label ranking for multi-class learning: application to object recognition. In *ICCV*, pages 2098–2105, 2009.

[2] C. Chang and C.J.Lin. Libsvm: a library for support vector machines. *ACM Transs on Intell. Sys. and Tech.*, 2:1–27, 2011.

[3] T. Ho, J. Hull, and S. Srihari. Decision combination in multiple classifier systems. *PAMI*, 16(1):66–75, 1994.

[4] Y. Jiang, Q. Dai, X. Xue, W. Liu, and C. Ngo. Trajectory-based modeling of human actions with motion reference point. In *ECCV*, 2012.

[5] T. Joachims. http://www.cs.cornell.edu/people/tj/.

[6] T. Joachims. http://svmlight.joachims.org/svm_multiclass.html.

[7] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002.

[8] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. The action similarity labeling challenge. *TPAMI*, 34(3):615–621, 2012.

[9] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012.

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[11] I. Laptev. On space-time interest points. *IJCV*, 64(2/3):107–123, 2005.

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1 –8, 2008.

[13] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, pages 1–8, 2008.

[14] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[15] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[16] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM Multimedia*, pages 357–360, 2007.

[17] B. Solmaz, A. Modiri, and M. Shah. Classifying web videos using a global video descriptor. *MVAP*, 2012.

[18] X. Sun, M. Chen, and A. Hauptmann. Action recognition via local descriptors and holistic features. In *CVPR Workshop H4B*, pages 58–65, 2009.

[19] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[20] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.