

# Action Recognition with Cascaded Feature Selection and Classification

Matteo Bregonzio, Shaogang Gong, Tao Xiang

School of Electronic Engineering and Computer Science,  
Queen Mary University of London, United Kingdom

{bregonzio, sgg, txiang}@dcs.qmul.ac.uk

**Keywords:** Action recognition, Feature selection, Cascade classifier.

## Abstract

Much of the previous action recognition work focuses on action representation whilst using standard multi-class classifiers such as SVM and k-NN for action classification. We show that these standard classifiers are inadequate in addressing more challenging action recognition problems encountered in an unconstrained environment and propose a novel action classification approach based on cascaded feature selection and classification. Instead of separating multiple action classes simultaneously, the more difficult single task is decomposed automatically into easier sub-tasks of separating two groups of the most separable action classes at a time with different features selected for different sub-tasks. Experiments are carried out using challenging public datasets to demonstrate that with identical action representation, our cascaded classifier significantly outperforms standard multi-class classifiers.

## 1 Introduction

One of the most important task for automated video surveillance is to recognize human actions captured in videos. Human action recognition is a challenging problem as actions can be performed by subjects of different sizes, appearance and poses. In an unconstrained environment the problem is compounded by the inevitable occlusion, illumination change, shadow, and camera movement. Some of the challenges are highlighted in Fig. 1, which shows examples from the Hollywood action recognition dataset [5]. It can be seen clearly that even when the image quality is relatively good in these movie sequences, to recognise actions outside a well-controlled laboratory environment is far from trivial.

A multi-class action recognition model consists of two essential parts: representation and classification. Most recent work [3, 4, 13, 14, 10, 6, 7] mainly addresses the first problem (actions representation) and simply uses off-the-shelf standard classifiers such as multi-class Support Vector Machine (SVM) or k-Nearest Neighbours (k-NN) for action classification. Although these classifiers have been successful in solving many pattern classification problems, they are inadequate for action recognition in an unconstrained environment because of: 1) It is difficult to simultaneously estimate the op-



Figure 1. Examples from the Hollywood dataset [5]. (a) Hand shaking from an unusual view angle. (b) Kissing in a crowd. (c) Getting out of a car under challenging lighting. (d) Hug a person with occlusion.

timal decision boundaries that separate highly ambiguous multiple action classes. Whatever action representation approach is taken, a multi-class action dataset is featured with large intra-class variations and inter-class similarities due to the aforementioned challenges. This poses serious problems for simultaneous multi-class separation using the standard classifiers. 2) Different action classes are often visually similar due to the shared atomic action components. For instance, running and jogging would involve mostly the same body parts moving in a very similar way. Hugging and kissing may look identical at the beginning of the action sequences. It is therefore critical to perform feature selection in order to identify the most discriminative features per inter-class before classification. However, different feature sets are useful for separating different groups of actions, and there will rarely be features that are universally informative for separating all classes simultaneously. Therefore the only solution to this problem is to select different sets of features for classifying different subsets. This unconventional but necessary feature selection requirement is not well met by deploying a standard multi-class classifiers such as SVM and

k-NN.

To address these problems, in this paper we propose a novel action classification approach which utilises a cascade of feature selection and binary classifiers. Instead of separating multiple action classes simultaneously, the overall task is decomposed automatically into easier sub-tasks of separating two groups of the most separable action classes at a time with different features selected for different binary classification sub-tasks. More specifically, our classifier iteratively split a group of action classes into two sub-groups until each sub-group only contains a single action class. Compared with the standard multi-class classifiers, a binary classifier in the cascade only needs to draw a single decision boundary between two groups of data that are most separable at a time. In addition, it allows for the selection of different sets of optimal features for separating different classes of actions.

The idea of using cascaded classifiers for solving difficult vision problems has been exploited before by Viola and Jones [12] and Athitsos et al. [1]. Specifically, Viola and Jones [12] propose a cascade of AdaBoost classifiers to address the face detection problem. Athitsos et al. [1] employ a cascade of approximate k-NN classifiers for recognising handwritten digits. Similar to our approach, their classifiers utilise different sets of features at different stages in a cascade with the later stage facing harder classification problems. However, there are two critical differences between our approach and theirs: 1) The goal of using cascaded classifiers is to speed up the classification process in their work, whereas in our study it is to improve the classification performance. 2) In each cascade stage, their classifier classifies the same set of patterns (e.g. face and non-face), whilst in our cascade, different classifiers are presented with different classification tasks. Our approach is also closely related the classification trees used in the machine learning community [9]. Nevertheless, the design method and the cascaded feature selection used in our approach distinguish our classifier from a conventional classification tree principally because both are automatically performed, as presented in section 3. The proposed model has been evaluated using two public datasets for action recognition: the KTH dataset [10], and the Hollywood dataset [5]. The latter is perhaps the most challenging and realistic dataset publically available. Our experimental results demonstrate that based on the same action representational schemes, our cascaded classifier significantly outperforms standard multi-class classifiers.

## 2 Action Representation

Similar to most existing approaches, our action representation is based on interest point detection and Bag of Word (BOW) descriptors. That is, salient information is extracted through interest points sampling from a video sequence before clustered and represented as histograms of visual words.

An interest point is defined as a spatio-temporal position considered to be descriptive of the action captured in a video. Among various interest point detection methods, the one proposed by Dollar et al. [3] is perhaps the most widely used. The interest points detected using this method correspond to local 3-D patches that undergo complex motions. Despite its

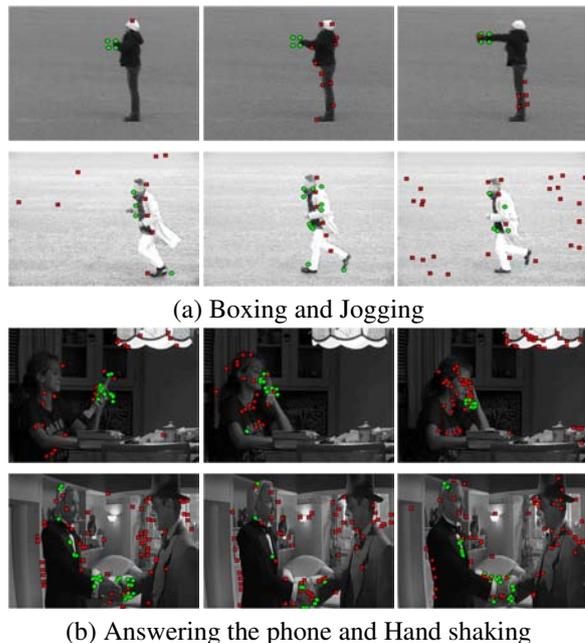


Figure 2. Examples of interest point detection. Red points are extracted using [3] while the green points using [2]. (a) are from the KTH dataset; (b) are from the Hollywood dataset.

popularity, the Dollar detector has a number of drawbacks: it ignores pure translational motions. Since it uses solely local information within a small region, it is prone to false detection due to video noise. It also tends to generate spurious detection background area surrounding object boundary and highly textured foreground areas. Moreover, it is particularly ineffective given slow object movement, small camera movement, or camera zooming.

To address some of these limitations, we adopt a recently proposed interest point detector by Bregonzio *et al.* [2]. In particular this detector employs different and more effective filters for detecting salient space-time local areas undergoing complex motions. More specifically, the detector consists of two steps: 1) frame differencing for focus of attention and region of interest detection, and 2) Gabor filtering on the detected regions of interest using 2D Gabor filters of different orientations. Via these two steps, saliency detection in both the temporal and spatial domains are combined together to give the filter response. The 2D Gabor filters are composed of two parts. The first part  $s(x, y; i)$  represents the real part of a complex sinusoid, known as the carrier:

$$s(x, y; i) = \cos(2\pi(\mu_0 x + \nu_0 y) + \theta_i) \quad (1)$$

where  $\theta_i$  defines the orientation of the filter and 5 orientations are considered:  $\theta_{i=1, \dots, 5} = \{0^\circ, 22^\circ, 45^\circ, 67^\circ, 90^\circ\}$ , and  $\mu_0$  and  $\nu_0$  are the spatial frequencies of the sinusoid controlling the scale of the filter. The second part of the filter  $G(x, y)$  represents a 2D Gaussian-shaped function, known as the envelope:

$$G(x, y) = \exp\left(-\frac{x^2}{\rho^2} + \frac{y^2}{\rho^2}\right) \quad (2)$$

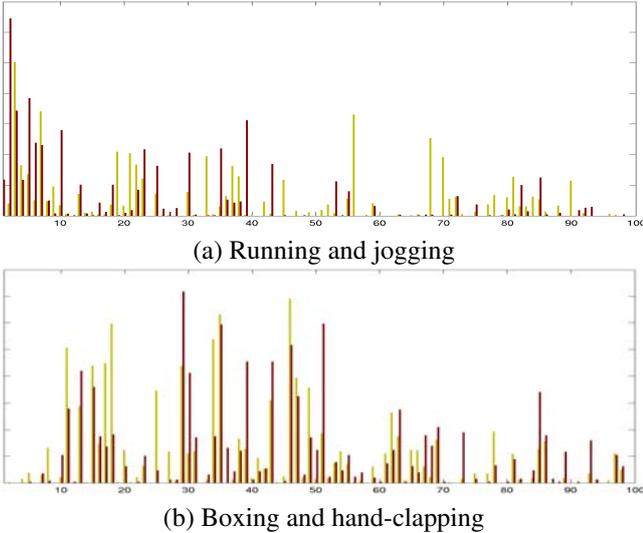


Figure 3. Examples of actions from the KTH dataset represented as histograms of visual words. The histograms have 100 bins (i.e. the size of the codebook is 100). (a) The light colour corresponds to running action while the dark one is for jogging. (b) The light colour is for boxing while the dark colour corresponds to hand-clapping.

where  $\rho$  is the parameter that controls the width of  $G(x, y)$ . We have  $\mu_0 = v_0 = \frac{1}{2\rho}$ ; therefore the only parameter controlling the scale is  $\rho$ , which is set to 11 pixels in this paper. Fig. 2 shows some examples of the interest point detection. It is evident that the detected interest point using the adopted method (green) are more meaningful and descriptive compared to those detected using the Dollar detector (red).

After interest point detection, a cuboid is extracted at each interest point which contains the spatio-temporally windowed pixel values. The size of the cuboid is set to  $8 \times 8$  pixel in the image coordinate and 12 frames along the time axis. Gradient based descriptors are then employed to map these cuboids into a high dimensional feature space which generates the local feature vectors [3]. Next, using K-means these feature vectors are clustered to generate a codebook. Finally, each video sequence is represented as a histogram of words with  $K$  bins, where  $K$  is the size of the codebook.

Examples of action descriptors as histograms of visual words computed using the interest point detector of [2] are shown in Fig. 3. It is evident by comparing Fig. 3(a) and (b) that running and jogging have a very similar distribution of visual words, which turns out to be quite different from that of boxing and hand-clapping. These four actions thus form two action class groups. This suggests that separating the two action classes within each group is much harder than distinguishing the two groups. Fig. 3 also highlights that 1) feature selection is necessary for action recognition; and 2) different sets of features should be selected for distinguishing different classes of actions. For instance, histogram bins numbered 1-20 should be selected if the task is to separate the two groups. However,

they are not very helpful in separating running and jogging. Similarly, bins numbered 65-70 are useful for classifying running and jogging, but not for boxing and hand-clapping. In order to explore these characteristics more effectively in assisting action recognition, we propose in the following a cascaded feature selection and classification model.

### 3 Cascaded Action Feature Selection and Classification

Given a training set containing sequences of  $M$  action classes  $A = [a_1, \dots, a_m, \dots, a_M]$ , we build a classification cascade, in each stage of which one or more binary classifiers are deployed to separate a group of action classes into two most separable sub-groups. Any sub-group that is composed of more than one action classes will be further divided into two in the next cascade stage. Critically, for each classifier in each stage, feature selection is performed to allow for different features been selected. The total number of stages  $S$  in the cascade will depend on in which stage all sub-groups contain only a single action class. The value of  $S$  thus ranges from  $\log_2 M$  to  $M - 1$  whilst the total number of binary classifiers is always  $M - 1$ .

The structure of the cascade is determined automatically using spectral clustering. Specifically, in order to group the  $M$  action classes into two sub-groups in the first stage of the cascade, the similarity between each pair of the  $M$  classes is computed which gives rise to a  $M \times M$  similarity matrix  $\mathbf{S} = \{S_{i,j}\}$  with  $S_{i,j}$  measuring the similarity between the  $i$ -th and the  $j$ -th action classes. To compute  $S_{i,j}$ , the averaged descriptors  $\mathbf{p}_i$  and  $\mathbf{p}_j$  are obtained for the  $i$ -th and the  $j$ -th action classes respectively. We then have:

$$S_{i,j} = 1 - \|\mathbf{p}_i - \mathbf{p}_j\| \quad (3)$$

where  $\|\mathbf{p}_i - \mathbf{p}_j\|$  is the Euclidean distance between the two averaged descriptors. The elements of  $\mathbf{S}$  are then normalised to be in the range of  $[0, 1]$ . Using the similarity matrix  $\mathbf{S}$  as input, the normalized cut algorithm [11] is employed to cluster the  $M$  action classes into two sub-groups. The same process is repeated for each sub-group that has more than two members.

After the structure of the cascade is determined, a binary classifier is to be trained for each group division in order to learn a more accurate decision boundary. The most informative and relevant features (histogram bins in our case) for training the classifier is automatically determined via feature selection for each classifier. To that end, a mutual information based feature selection method [8] is utilised to rank all the features. The optimal number of features to be kept for training the classifier is then determined by cross-validation. The whole process of training a cascaded action classifier is illustrate in Fig. 4.

Once the cascaded classifier is learned, it is straightforward to classify an unknown action sequence into one of the  $M$  action classes. Specifically, the action in the sequence is first classified into one of the two sub-groups by the binary classifier in stage 1. If the sub-group into which it is classified contain more than one action classes, it is further classified by the binary classifier learned for that sub-group. This process is repeated until the action falls into a sub-group with only one action class (not necessarily in the last cascade stage). Therefore

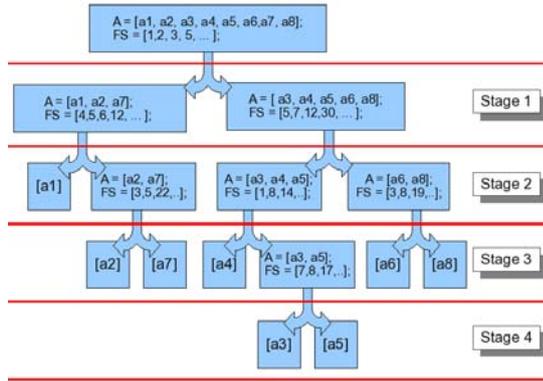


Figure 4. An example of cascaded action classifier. Starting with a group of 8 action classes in stage 1, all action classes are separated by stage 4. Note that for dividing each sub-group into two, a different set of features are used. In this example, the cascade consists of 4 stage and 7 binary classifiers.

a minimum of one and a maximum of  $M - 1$  binary classifications are needed for assigning a label for an action sequence.

## 4 Experiments

### 4.1 Datasets

**KTH Dataset** – The KTH dataset was provided by Schudt et al. [10] in 2004. It contains 6 types of actions (boxing, hand clapping, hand waving, jogging, running and walking) performed by 25 subjects in 4 different scenarios including indoor, outdoor, changes in clothing and variations in scale. Each video clip contains one subject performing a single action. Each subject was captured in a total of 23 or 24 clips, giving a total of 599 video clips. Each clip is sampled at 25Hz and lasts between 10 to 15 seconds with an image frame size of  $160 \times 120$ . Examples of the 6 action classes can be seen in Fig. 5.

**Hollywood Dataset** – The Hollywood actions dataset [5] contains eight different action classes: answering the phone, getting out of the car, hand shaking, hugging, kissing, sitting down, sitting up, and standing up. These actions were collected from 32 different Hollywood movies. The full dataset contains 663 video clips sampled at 25 Hz and each of them has a different frame size and duration. The dataset is divided into a clean and an automatically labelled clips [5]. In our experiments, we use the clean set only. Our training set contains 219 clips while the testing set contains 211 clips. As shown in Fig. 5, compared with the KTH dataset, many actions were performed by more than one person with multiple moving people in the background. The variations in lighting, view angle and camera movement are also far greater. All these make this dataset the most difficult dataset available so far.

### 4.2 Experimental Settings

Any type of binary classifier can be used to form the classification cascade. In our experiment we used the absolute distance based k-NN and SVM with polynomial kernel. The optimal k

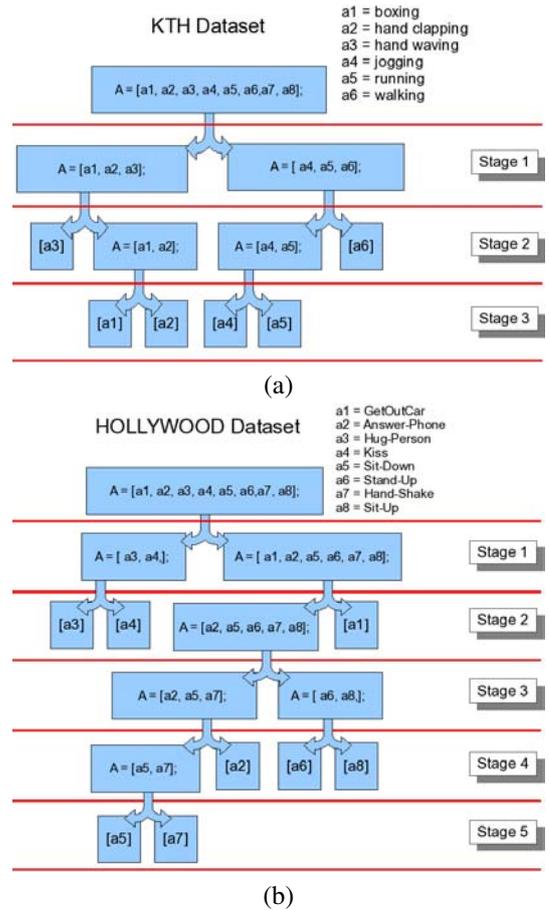


Figure 6. Cascade classifier structure. (a) KTH dataset (b)Hollywood dataset.

value and polynomial degree were determined by cross validation. Different codebook sizes were tested for the BOW representation. The results are reported using a codebook size of 200 for KTH and 300 for Hollywood when not stated otherwise. In order to present a fair comparison with the state-of-the-art results we followed the most widely used validation procedure. This means that for the KTH dataset the average class accuracy (ACA) was used [3], and the average precision (AP) of the precision/recall curve was used for the Hollywood dataset [5]. Since the codebook was generated using K-means which is sensitive to initialisation, the results are reported based on the average of 20 trials.

### 4.3 Recognition Performance Evaluation

**Learning classifier structures:** Fig. 6 show the learned structures of the cascades for both datasets, which were automatically determined (see Sec. 3). It can be seen that both learned structures reflect accurately the natural grouping of the different action classes. For instance, Fig. 6 (a) shows that the 6 action classes in the KTH dataset were divided into two sub-groups in the first cascade stage: jogging, running and walking in one sub-group which all involve movement from legs, and boxing, clapping and waving in the other which are featured mainly with movements from the upper body. For the Holly-



Figure 5. Examples from the KTH dataset (top) and the Hollywood dataset (bottom).

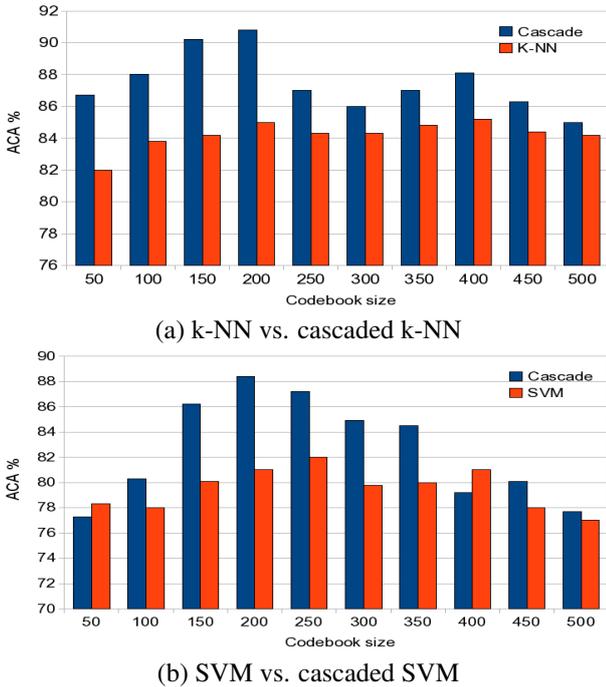


Figure 7. Comparing cascaded and standard classifiers on the KTH dataset given different codebook sizes.

wood dataset in the first stage of the cascade two action classes HugPerson and Kiss are grouped together and separated from the rest 6 classes. These two classes are visually very similar whilst being distinctive from other action classes (see Fig. 5). It can also be seen from Fig. 6 that similar action classes such as StandUp and SitUp, jogging and running stay grouped until a binary separation between them is carried out.

**Cascaded classifiers vs. Standard classifiers:** In this experiment we compare the performance of our cascaded classifiers with that of standard multi-class classifiers including k-NN and SVM using identical action representation (Section 2). Fig. 7 shows the performance of cascaded k-NN, cascaded SVM, standard k-NN and SVM on the KTH dataset. The results obtained with different codebook sizes are also shown to examine its effect on different classifiers. The confusion matrices obtained using our cascaded classifiers are shown in Fig. 8. It is

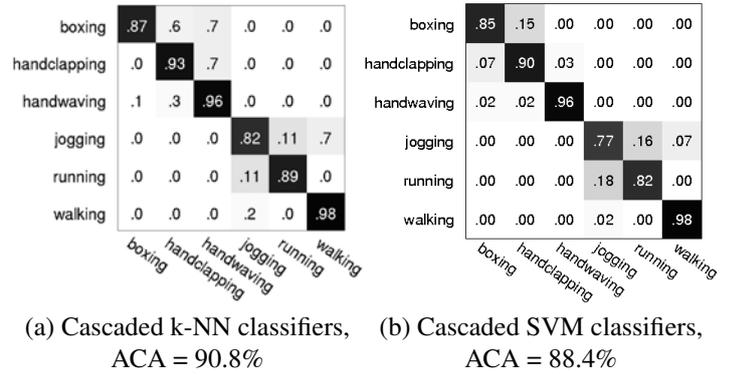


Figure 8. Confusion matrix on KTH dataset.

evident from Fig. 7 that our classifiers significantly outperform the standard k-NN and SVM. The results obtained on the Hollywood dataset are shown in Table 1. Again, it is evident that a large improvement in performance regardless the type of classifier used. With the same action representation and the same type of classifier (for binary classification in our classifiers), this improvement can only be contributed by the proposed cascaded feature selection and classification method.

**Comparison to the state-of-the-art:** We compared our method to the state-of-the-art on the KTH and Hollywood datasets in Table 2. It indicates that for the KTH dataset, our result are much better than the existing methods that are based on a similar action representation [7, 3]. For those that give slight better result [4, 2, 5], much more sophisticated action representation methods are employed. In particular, they all explored spatial distribution information of the visual words. In contrast, this information has not been taken into account by our current model. As for the Hollywood dataset, so far only two previous studies have reported results. Among them [4] does not exploit the spatio-temporal distribution of interest points but uses a more sophisticated interest point descriptor than us. However, our result is still clearly superior to that in [4]. The result obtained in [5] is better than ours. However, different spatio-temporal bag-of-feature representations were exhaustively examined in their work and only the best one for each action class was used to produce their result. It is therefore unfair to compare the two results.

	Cascaded k-NN	k-NN	Cascaded SVM	SVM
GetOutCar	19.3 %	17.4 %	22.3 %	20.2 %
AnswerPhone	22.5 %	18.2 %	38.4 %	32.4 %
HugPerson	21.6 %	12.7 %	33.5 %	28.8 %
Kiss	47.6 %	41.8 %	46.7 %	32.1 %
SitDown	31.5 %	30.1 %	37.9 %	17.3 %
StandUp	41.3 %	33.4 %	42.3 %	32.0 %
HandShake	13.5 %	12.1 %	21.0 %	19.5 %
SitUp	4.2 %	4.2 %	8.4 %	6.6 %
Average	25.19 %	21.24 %	<b>31.31 %</b>	23.61 %

Table 1. Comparing cascaded and standard classifiers on the Hollywood dataset.

	Our	Laptev et al. [5]	Kläser et al. [4]	Bregonzio et al. [2]	Niebles et al. [7]	Dollar et al. [3]
KTH	<b>90.8%</b>	91.8%	91.4%	93.17%	81.5%	81.17%
Hollywood	<b>31.31%</b>	38.39%	24.7%	-	-	-

Table 2. Comparative results on the KTH and Hollywood datasets.

Note in this paper our goal is to improve the performance of action recognition via the novel cascaded feature selection and classification method regardless of the adopted action representation and binary classifier in each stage. Our results in Fig. 7 and Table 1 have clearly demonstrated that this goal has been achieved. We expect that our results will be further improved when more descriptive action representation methods such as those in [5, 6] are employed.

## 5 Conclusion

We have proposed a novel action classification approach based on cascaded feature selection and classification. Instead of separating multiple action classes simultaneously, the difficult task is decomposed automatically into easier sub-tasks of separating two groups of the most separable action classes at a time with different features selected for different sub-tasks. Experiments are carried out using challenging public datasets to demonstrate that with identical action representation, our cascaded classifier significantly outperforms standard multi-class classifiers. The ongoing work includes investigation on the effect of different action representation and feature selection methods on the performance of our cascaded classifier.

## References

- [1] V. Athitsos, J. Alon, and S. Sclaroff. Efficient nearest neighbor classification using a cascade of approximate similarity measures. In *CVPR*, 2005.
- [2] M. Bregonzio, S. Gong, and T. Xiang. Recognising action as clouds of space-time interest points. In *CVPR*, 2009.
- [3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.
- [4] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.
- [5] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.
- [6] Jingen Liu and Mubarak Shah. Learning human actions via information maximization. In *CVPR*, 2008.
- [7] J. C. Niebles, H. Wang, H. Wang, and F. F. Li. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.
- [8] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. In *PAMI*, 2005.
- [9] S.R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Trans. Systems Man Cybernet*, 1991.
- [10] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.
- [11] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *PAMI*, 2000.
- [12] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.
- [13] Ziming Zhang, Yiqun Hu, Syin Chan, and Liang-Tien Chia. Motion context: A new representation for human action recognition. In *ECCV*, 2008.
- [14] Z.P. Zhao and A.M. Elgammal. Information theoretic key frame selection for action recognition. In *BMVC*, 2008.