A FAST VOTING-BASED TECHNIQUE FOR HUMAN ACTION RECOGNITION IN VIDEO SEQUENCES

Duc-Hieu Tran and Wooi-Boon Goh

Centre for Multimedia and Network Technology, School of Computer Engineering, Nanyang Technological University, Singapore, Singapore
{*tran0066, aswbgoh*}*@ntu.edu.sg*

Keywords: Human Action Recognition, Local Feature, Voting Framework, Nearest Neighbor.

Abstract: Human action recognition has been an active research area in recent years. However, building a robust human action recognition system still remains a challenging task due to the large variations in action classes, varying human appearances, illumination changes, camera motion, occlusions and background clutter. Most previous work focus on the goal of improving recognition rates. This paper describes a computationally fast voting-based approach for human action recognition, in which the action in the video sequence is recognized based on the support of the local spatio-temporal features. The proposed technique requires no parameter tuning and can produce recognition rates that are comparable to those in recent published literature. Moreover, the technique can localize the single human action in the video sequence without much additional computation. Recognition results on the KTH and Weizmann action dataset are presented.

1 INTRODUCTION

Human action recognition in video sequences is an active research area in the Computer Vision community. The approaches to this problem can be divided into holistic approach (Bobick and Davis, 2002; Weinland et al., 2006; Yilmaz and Shah, 2005; Blank et al., 2005; Aggarwal and Ryoo, 2011) and local feature approach (Laptev, 2005; Dollar et al., 2005; Willems et al., 2008; Yuan et al., 2009; Niebles et al., 2008). In a local feature approach, the action is represented as a set of local features extracted from the video sequence. The actions can be then classified using parametric learning methods (e.g. Support Vector Machine, Boosting, etc.) or a non-parametric technique such as the nearest neighbor classifier. In general, these machine learning techniques require the computation of similarity between two feature sets. To reduce the computational complexity, the feature vectors are often quantized to form a visual vocabulary and the video is then represented as a histogram of the visual codewords in this vocabulary. This is the well-known bag of word model technique (Wang et al., 2009).

Unfortunately, the quantization process often results in the loss of information. Moreover, as a property of clustering algorithms, the clusters will be formed from the dense feature clouds, which are often the common features in all videos. These general features are good for building a vocabulary but they are not discriminative enough to describe the various human actions in video sequences. On the other hand, the sparse but discriminative features will be often removed during the quantization process due to their larger distances from the cluster centroids (i.e., visual words) (Boiman et al., 2008). The large variation in human appearances and action viewpoints also means that the size of the vocabulary must be large enough to achieve reasonably discriminative codewords. As stated in (Wang et al., 2009), the vocabulary size should be around 4000 for a wide range of datasets. With such a large visual vocabulary, the computational time required to represent a video as a histogram of visual words can be quite long since it is necessary to compute the distance from each feature to every visual word.

Most previous work on human action recognition focus on new approaches to improve the recognition rates. Few address the computational speed issues of the recognition process. Inspired by the query-to-class distance proposed in (Boiman et al., 2008) and the use of random forest in object recognition to speed up the classification process, we proposed a fast voting-based framework, in which each local feature will cast a support for the action class in the video. In addition, the approach avoids the need to quantize

the feature vectors and thus avoids the need to tune parameters (Laptev et al., 2008; Yuan et al., 2009; Willems et al., 2008). The video will then be assigned to the action class of the major support within all the detected local features. Experiments on the KTH (Schuldt et al., 2004) and Weizmann (Blank et al., 2005) datasets gave comparable recognition rates to that in recent literature and the computational time for classification is much faster than the bag of word model approach.

2 RELATED WORK

In a voting framework where a video (or an image in object recognition case) is represented as a set of features, each feature will cast a vote for the action category (or object category, respectively) contained in the video or the image. Given a video that is represented as a set of features, the posterior probability $P(C|f)$ of a feature f to a class C can be viewed as a vote of that feature for the class C. We discuss previous work that classifies the image, objects or action based on the voting of individual feature to a class.

Yuan et al. (Yuan et al., 2009) computed the votes $s^C(f)$ of feature f for a class C and the action recognition is transformed into the problem of searching for a subvolume of maximum supporting score for the action class.

In (Gall and Lempitsky, 2009) the posterior probability $P(C|f)$ of a feature f belonging to class C is computed from a random forest (Breiman, 2001). Given a built random forest, if the feature f falls into a leaf node L then $P(C|f)$ will be estimated as the ratio of features from class C at leaf node L. Yao et al. (Yao et al., 2010) extended the idea to human action recognition, however their approach is time consuming and computational costly since it requires human body detection and tracking in each video frame.

In an unsupervised approach, Niebles et al. (Niebles et al., 2008) applied the probabilistic Latent Semantic Analysis (pLSA) model to the problem of human action recognition. From their model, one can extract the value $p(C|w,v)$, the probability of action C for a visual word w in a particular video v. The action can be localized as the cluster of the estimated local features.

In a bag of word model, the vocabulary is often built by a clustering method such as k-means. Each cluster centroid is considered as a visual word and a feature is quantized to the most similar visual word (i.e., the closest by a distance measure such as L_1 or Euclidean distance). This process can be seen as a way to separate the feature space into regions of similar features. The k-means clustering approach tends to group region with dense features into a cluster. As a result, discriminative features are often suppressed over commonly occurring features due to their large distance (dissimilarity) to a centroid. Moreover, k-means clustering methods are computationally costly, especially when the number of clusters is large.

In (Moosmann et al., 2006) the authors proposed a method called Extremely Randomized Clustering Forest to effectively build visual vocabulary using a random forest (Breiman, 2001). At each node of a random tree, the feature set is splited into two discriminative sets by an optimal binary function. The leaf nodes of the trees will form the visual vocabulary where each leaf node is a visual word.

Lepetit et al. (Lepetit et al., 2005) performed local feature classification in real-time by randomized trees to localize an instance image in a video. Their binary function for an image patch is slightly different in that it is the comparison of the intensity value of two random pixels in the image patches.

Although using different binary functions, the building of visual vocabulary or the classification of local patches by random forests are based on the same observation that the local image patch can be seen as a vector of its flattened intensity values or encoding descriptor. Then the patch can be represented as a point in N-dimensional feature space, where N is the number of pixels in the image patch or the dimension of the feature descriptor. The binary function at each tree node will define a hyperplane that splits the feature space into two discriminative regions. From that point of view, each leaf node of a tree can be seen as a region in feature space in which the features are similar but highly distinguishing to those in other regions.

Also inspired by the use of random forests, in (Silpa-Anan and Hartley, 2008) the authors have proposed a framework to approximately find the nearest neighbors using randomized kd-trees. Instead of uniformly choosing the dimension to split as in traditional kd-trees, they randomly choose the dimension from D dimensions in which the data has the greatest variance. The experiments in (Muja and Lowe, 2009) showed that it is efficient to fix $D = 5$. The idea of building randomized kd-trees is very similar to that of building a visual vocabulary in (Gall and Lempitsky, 2009).

3 THE PROPOSED APPROACH

Inspired by the fast recognition of keypoints in images (Lepetit et al., 2005) and the Naive Bayes Nearest Neighbor classifier (Boiman et al., 2008) that is

based on the image-to-class distance, we proposed a voting framework called Vote-1NN.

Given a test video Q consisting of local spatio-temporal features $f_1, \ldots, f_K$ which are independent identically distributed, under the Bayesian assumption and the uniform of the prior probability $P(c)$, the Maximum Likelihood classifier is formulated as:

$$\hat{c} = \arg\max_c P(Q|c) = \arg\max_c \prod_{i=1}^{K} P(f_i|c)$$
$$= \arg\max_c \prod_{i=1}^{K} \frac{P(c|f_i)P(f_i)}{P(c)} \quad (1)$$

Assuming the uniform of $P(f_i)$ and $P(c)$, take the log probability of the rule, we have:

$$\hat{c} = \arg\max_c \sum_{i=1}^{K} \log P(c|f_i) \quad (2)$$

Assume the local features are extremely discriminative and each local feature can be classified to one of the action classes:

$$P(c|f_i) = \begin{cases} 1, & \text{if } f_i \text{ is from class } c \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The posterior probability $P(c|f_i)$ can be intuitively interpreted as the support of feature f_i to class c. With an arbitrary distribution of local features, the support of a local feature f_i in a test video to an action class c can be assigned as the same as the support of its nearest neighbor to c. This is the key idea of the classification by the nearest neighbor, but here we apply it to the local features but not the video sequences. From Equation (2) and (3), we define function $g(c, f_i)$ as follows:

$$g(c, f_i) = \begin{cases} 1, & \text{if the nearest neighbor of } f_i \\ & \text{is from class } c \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The original classification problem is formulated as:

$$\hat{c} = \arg\max_c \sum_{i=1}^{K} g(c, f_i) \quad (5)$$

The formulation in (5) can interpreted as follows: the human action in a video sequence will be recognized based on the majority votes of the local features.

In the pre-processing phase, a training feature set is formed from all the features extracted from training video sequences. These features are assigned to the action class of the video from which they are extracted. Given a feature f_i from a test video sequence, based on the action class of its nearest neighbor in the training feature set, we can compute the value of $g(c_k, f_i)$ for all action classes c_k at once. In our experiment, the search of the nearest neighbor can be performed reliably and speedily by building the randomized kd-trees (Silpa-Anan and Hartley, 2008). Our proposed approach is computational fast, it does not require a complicated training phase but only a pre-processing phase in which the randomized kd-trees are built.

After recognizing the human action in the given video sequence, the center of the human action can be localized based on the positions of the local features that support the recognized action class. According to the experiment on the KTH dataset in (Schindler and Gool, 2008), human action can be accurately recognized in about 7 frames of a video sequence. As such, the center of the human body performing the recognized action in a given frame is also estimated as the centroid of all supporting features (i.e., the features that support the recognized action class) about 7 neighboring frames. The proposed process is straightforward, has low computational cost and yet produces acceptable localization accuracy.

4 EXPERIMENT AND DISCUSSION

We performed the experiments on two popular human action datasets: KTH (Schuldt et al., 2004) and Weizmann (Blank et al., 2005). We also compared our approach with an extension of the NBNN image classifier (Boiman et al., 2008).

4.1 Extended Naive Bayesian Nearest Neighbor (NBNN)

Since the work of (Boiman et al., 2008) is similar to the Vote-1NN approach with terms of query-to-class distance, we extended the Naive Bayes Nearest Neighbor (NBNN) image classifier to perform human action recognition in video sequences. The NBNN classifier for human action recognition was then tested on the KTH and Weizmann action dataset in the following manner:

1. for a test video sequence, extract the local features $f_1, f_2, \ldots, f_n$
2. action class $\hat{C} = \arg\max_C \sum_{i=1}^{n} ||f_i - \mathrm{NN}_C(f_i)||$, where $\mathrm{NN}_C(f_i)$ is the nearest neighbor of f_i within all features of class C. We use L_1 distance instead of L_2 as in (Boiman et al., 2008).

4.2 KTH Action Dataset

The **KTH Action Dataset** consists of 2391 video sequences of six human actions: boxing, hand waving, hand clapping, running, jogging and walking. The actions were performed by 25 subjects in four different scenes. Each action is performed three or four times by each subject in each scene. Following the standard setup in (Schuldt et al., 2004), videos of sixteen subjects were used for training and videos of nine remaining subjects were used for testing. The recognition rates are reported as average class accuracy.

The features were detected by Harris3D interest point detector (Laptev, 2005) and represented by different types of descriptors: Histogram of Oriented Spatial Gradient (HOG), Histogram of Optical Flow (HOF), HOGHOF (i.e., the combination of HOG and HOF) (Laptev et al., 2008) and Histogram of 3D Gradients (HOG3D) (Klaser et al., 2008).

The nearest neighbor search in our experiment is performed by building the randomized kd-trees (Silpa-Anan and Hartley, 2008). We used the existing library FLANN provided by Muja and Lower (Muja and Lowe, 2009) with the following settings: the number of random dimensions is 5, the number of randomized trees is 10, the distance between features is L_1.

The results for the HOG, HOF, HOGHOF and HOG3D descriptors respectively are 82.16%, 89.57%, 91.08% and 89.34%. Table 1 shows the confusion table for HOGHOF descriptor.

Table 1: The result of our approach (Vote-1NN) with Harris3D interest point detector and HOGHOF descriptors on the KTH dataset. The average accuracy is 91.08%.

gt\res	box	hclap	hwave	jog	run	walk
box	**97.9**	0	0	0	0	2.1
hclap	1.39	**97.92**	0.69	0	0	0
hwave	0	6.94	**93.06**	0	0	0
jog	0	0	0	**95.14**	2.78	2.08
run	0	0	0	37.5	**62.5**	0
walk	0	0	0	0	0	**100**

There is no misclassification of the moving actions (i.e., jogging, running, walking) to the stationary actions (i.e., boxing, handclapping, handwaving) with any of the four types of feature descriptors. This means that despite the use of appearance features or motion features, our approach can accurately distinguish these two kinds of actions. The results on the running action are poor and it is mostly misclassified to the jogging action. The higher accuracy of running action achieved by HOG and HOG3D features (i.e., 63.19% and 75.69% respectively) is probably due to the more reliable differences in human poses of running and jogging (stride variations) compared to temporal changes due to leg motions. HOGHOF descriptor that encodes both appearance and motion information gave the best results with overall accuracy of 91.08%, which is comparable to the state of the art results shown in Table 3. Although the video sequences in KTH dataset contain slight camera motion and variant view points, the high recognition rates achieved by our approach (i.e., except for misclassification of running to jogging, the average accuracy for other five actions is about 97%) suggest that our approach is reasonably robust to the problems of camera motion and view variance. These problems are more difficult to solve in a framework that uses vector quantization. The results of extended NBNN for KTH dataset using descriptors HOG, HOF, HOGHOF and HOG3D are 84.94%, 89.80%, 92.24% and 91.31% respectively. This could be due to the fact that in NBNN approach, the distances from the local features in the query video to their nearest neighbors in all action classes are computed and accumulated. In that manner, the features that are not highly discriminative (i.e., its distance to the nearest neighbors in all action classes are only slightly different) will not contribute significantly to the final decision of the action class. In contrast, the local features in our approach are each assigned to only one action class. As a result, we reap the advantage of lower computational complexity since searching for the nearest neighbor is done only once.

4.3 Computational Speed

Based on a KTH dataset test case where the features have been extracted by Harris3D and described by the HOGHOF descriptor, we compared the performance of our approach with the bag of word model approaches by following the experiment setup on the KTH action dataset described in (Wang et al., 2009) (i.e., the number of visual words is 4000 and the visual vocabulary is built by k-means clustering). The classifications were performed using Support Vector Machine (Chang and Lin, 2001) with χ^2 kernel (Laptev et al., 2008) and Pyramid Match Kernel (PMK) (Grauman and Darrell, 2007). Our approach does not require us to build the visual vocabulary – a process that requires a lot of time, especially for a large vocabulary size. Moreover, it only takes an average of 0.05 seconds for our Vote-1NN approach to classify an action in a video sequence with an average of 100 frames. Using the same computational resources and test video sequences, the bag of word model with SVM+χ^2 kernel and SVM+PMK took an average of 0.95 and 2.5 seconds respectively to

Table 2: The performance of different approaches on the KTH action dataset. Time of performing recognition is estimated on video sequences of average 100 frames. Our approach Vote-1NN can perform much faster than the bag of word model approaches and is able to localize the center of human action in video sequences.

	Vote-1NN	NBNN	SVM + χ^2	SVM+PMK
time (sec) *	**0.05**	0.25	0.95	2.5
accuracy (%)	91.08	92.24	91.80	91.08
localization	**YES**	NO	NO	NO

classify an action. The comparative recognition rates of the various approaches are comparable, as shown in Table 2, while our approach can compute the results much faster (i.e., 5, 20 and 50 times faster than NBNN, SVM+χ^2 and SVM+PMK respectively) and it is able to localize the center of human action.

Table 3: Recognition rates on the KTH dataset of the extended NBNN and our Vote-1NN approach compared to other well known techniques. Except for (Niebles et al., 2008), all experimental setups are consistent.

Extended NBNN	92.24%
Vote-1NN	**91.08%**
Subvolume search (Yuan et al., 2009)	93.3%
Harris3D + SVM-χ^2 (Laptev et al., 2008)	91.8%
Unsupervised learning (Niebles et al., 2008)	83.3%
Harris3D + local SVM (Schuldt et al., 2004)	71.7%

4.4 Human Action Localization

We localized the recognized action based on our Vote-1NN approach by estimating the action centers in the test KTH video sequences using the centroid of the spatial positions of the local features that have been classified to the action class. The centroid position is computed from the cluster of features within 7 consecutive frames centered about the current frame. The location of the action is approximately bounded by a spatial window containing the features used to compute the spatial mean. The action center (x_C, y_C) is computed as average of the spatial locations (x_i, y_i) of all local features that support the classified action in the video sequences. The local window around the action is defined by the upper-left (x_L, y_U) and the lower-right (x_R, y_L), where x_L and x_R are the average of x-position of the local feature points on the left and right respectively of the action center point (x_C, y_C), y_U and y_L are the average of y-position of the local points above and below respectively of the point (x_C, y_C). From our empirical studies, the local action window specified by (x_L, y_U) and (x_R, y_L) is enlarged to twice the size to better capture the spatial area where the human action can be localized.

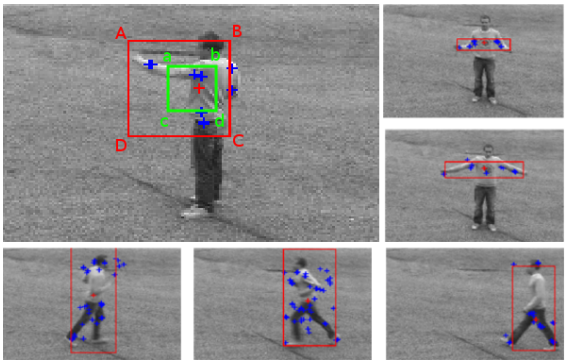

Figure 1: The localization of six action classes in the KTH dataset. Computed centers of actions (x_C, y_C) are marked with a red cross-hair and the local features supporting the action are blue. The upper-left image illustrates the localization of actions in video sequences. The small window (a,b,c,d) is estimated by values $(x_L, y_U), (x_R, y_L)$, and the local action window (A,B,C,D) that is at twice size of (a,b,c,d) will capture the spatial area where the human action can be localized.

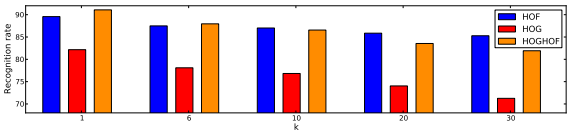

Figure 2: The recognition rates with respect to value of K on KTH action dataset.

We manually checked the action location estimated for 100 randomly selected video sequences from the KTH dataset and found that the centers were appropriately located at the human body performing the actions. This suggests that the Vote-1NN approach can be used to localize the action centers as shown in Figure 1.

We analyzed the effect of the number of nearest neighbors to the recognition rates on the KTH action dataset. Instead of assigning the action class of the nearest neighbor to a test local feature from the test video sequences, as is the case for Vote-1NN, we assigned the action class represented by the majority among K nearest neighbors (i.e., the standard K-nearest neighbor setting). The recognition rates with respect to different value of K are shown in Figure 2. Increasing value of K decreases the recognition rates and the best performance is achieved by our proposed Vote-1NN approach (i.e., $K = 1$). This result justifies our proposed approach of using only a single nearest neighbor.

4.5 Weizmann Action Dataset

We also performed the experiment on the **Weizmann action dataset** (Blank et al., 2005). The Weizmann dataset consists of 93 video sequences of ten different action classes performed by 9 subjects in the scene with static background. The results of Vote-1NN and the extended NBNN on Weizmann dataset are shown in Table 4. We use the leave-one-subject-out setup for the Weizmann dataset, i.e., for each test, video sequences of one subject are used for testing while the training feature set is formed from the video sequences performed by the remaining subjects. The recognition rates shown are computed as an average accuracy of all tests. Like the results from the KTH dataset, results from the Weizmann dataset show that both Vote-1NN and NBNN have comparable performance as well. The action localization is performed on the Weizmann dataset and manually checked. The samples of action localizations for a random selected subject in the dataset are shown in Figure 3. The comparison of our approach on the Weizmann dataset with several well-known bag of word model approaches is shown in Table 5.

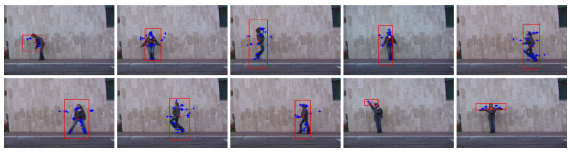

Figure 3: The localization of ten action classes in the Weizmann dataset. Computed centers of actions is marked with a red cross-hair. The local action windows are marked by red rectangles.

Table 4: The results of our Vote-1NN approach and NBNN on the Weizmann action dataset with interest point detector Harris3D and descriptors HOG, HOF and HOGHOF.

Approaches	HOG	HOF	HOGHOF
Extended NBNN	81.88%	90.00%	90.26%
Vote-1NN	83.85%	**91.11%**	89.15%

Table 5: The comparison of our approach with selected bag of word model approaches on the Weizmann action dataset. Based on similar experimental setups, the recognition rates of Vote-1NN is comparable to the best of the bag of word model approaches.

Vote-1NN	**91.11%**
Extended NBNN	90.26%
3D SIFT + BoW (Scovanner et al., 2007)	82.60%
HOG3D + BoW (Klaser et al., 2008)	84.30%
Unsupervised learning (Niebles et al., 2008)	90.00%
Multiple features (Liu et al., 2008)	90.40%

5 CONCLUSIONS AND FUTURE WORK

In this paper we have presented a novel method for human action recognition based on the support of each local feature for its closest action class. Recognition rates of the KTH and Weizmann action datasets obtained by our approach are comparable to the state of the art results. The action classification speed is about 20 to 50 times faster than the various bag of word model approaches tested and 5 times faster than the extended NBNN approach. In addition, the proposed Vote-1NN approach is able to localize the human actions without much additional computational cost. We have also verified experimentally that using one nearest neighbor (i.e., Vote-1NN) produces the best recognition performance compared to case, where more nearest neighbors were used. Our next challenge is to apply this voting approach to recognize and localize multiple simultaneous actions in video sequences based on the classification of individual local features.

REFERENCES

Aggarwal, J. and Ryoo, M. S. (2011). Human Activity Analysis : A Review. *ACM Computing Surveys*.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *CVPR*, volume 29, pages 1395–1402.

Bobick, A. and Davis, J. (2002). The recognition of human movement using temporal templates. *TPAMI*, 23(3):257–267.

Boiman, O., Shechtman, E., and Irani, M. (2008). In defense of Nearest-Neighbor based image classification. In *CVPR*, pages 1–8.

Breiman, L. (2001). Random forests. *ML*, 45(1):5–32.

Chang, C.-C. and Lin, C.-J. (2001). LIBSVM: a library for support vector machines.

Dollar, P., Rabaud, V., Cottrell, G., and Serge, B. (2005). Behavior Recognition via Sparse Spatio-Temporal Features. In *VS-PETS*, pages 65–72.

Gall, J. and Lempitsky, V. (2009). Class-Specific Hough Forests for Object Detection. In *CVPR*, pages 1022–1029.

Grauman, K. and Darrell, T. (2007). The pyramid match kernel: Efficient learning with sets of features. *JMLR*, 8(2):725–760.

Klaser, A., Marszalek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. In *BMVC*.

Laptev, I. (2005). On Space-Time Interest Points. *IJCV*, 64(2-3):107–123.

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

Lepetit, V., Lagger, P., and Fua, P. (2005). Randomized trees for real-time keypoint recognition. In *CVPR*, pages 775–781.

Liu, J., Ali, S., and Shah, M. (2008). Recognizing human actions using multiple features. In *CVPR*, pages 1–8.

Moosmann, F., Triggs, B., and Jurie, F. (2006). Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, pages 985–992. MIT Press.

Muja, M. and Lowe, D. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, pages 331–340.

Niebles, J. C., Wang, H., and Fei-Fei, L. (2008). Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *IJCV*, 79(3):299–318.

Schindler, K. and Gool, L. V. (2008). Action Snippets: How many frames does human action recognition require? In *CVPR*, pages 1–8.

Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local SVM approach. In *ICPR*, volume 3, pages 32–36.

Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *ACM MM*, page 357.

Silpa-Anan, C. and Hartley, R. (2008). Optimised KD-trees for fast image descriptor matching. In *CVPR*.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *BMVC*, pages 1–11.

Weinland, D., Ronfard, R., and Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2-3):249–257.

Willems, G., Tuytelaars, T., and Gool, L. V. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, pages 650–663.

Yao, A., Gall, J., and Gool, L. V. (2010). A Hough Transform-Based Voting Framework for Action Recognition. In *CVPR*.

Yilmaz, A. and Shah, M. (2005). Actions sketch: A novel action representation. In *CVPR*, volume 1, pages 984–989.

Yuan, J., Liu, Z., and Wu, Y. (2009). Discriminative subvolume search for efficient action detection. In *CVPR*, pages 2442–2449.