Localizing Actions through Sequential 2D Video Projections

Hakan Boyraz[1]
hboyraz@knights.ucf.edu

Marshall F. Tappen[1]
mtappen@eecs.ucf.edu

Rahul Sukthankar[2]
rahuls@cs.cmu.edu

[1]School of EECS, University of Central Florida
[2]Robotics Institute, Carnegie Mellon University

Abstract

Action detection in video is a particularly difficult problem because actions must not only be recognized correctly, but must also be localized in the 3D spatio-temporal volume. This paper introduces a technique that transforms the 3D localization problem into a series of 2D detection tasks. This is accomplished by dividing the video into overlapping segments, then representing each segment with a 2D video projection. The advantage of the 2D projection is that it makes it convenient to apply the best techniques from object detection to the action detection problem. Our experiments show that video projection outperforms the latest results on action detection in a direct comparison.

1. Introduction

As imaging systems become ubiquitous, the ability to recognize human actions is becoming increasingly important. Just as in the object detection and recognition literature, action recognition can be roughly divided into classification tasks, where the goal is to classify a video according to the action depicted in the video, and detection tasks, where the goal is to detect and localize a human performing a particular action.

The detection task is particularly challenging because the action must be detected and localized in a spatio-temporal volume. In the worst case, the system must search a six-dimensional space to locate the video volume. Recent work has built on Lampert *et al.*'s Efficient Subwindow Search method (ESS) [11] to make such searches efficient [3, 23]. While successful, these action detection methods are distinct from the popular techniques currently used for object detection, localization and classification in images, since the former employ mutual information or generative models rather than discriminative classifiers over feature descriptors.

In this paper, we show that actions can be localized without explicitly searching through time. Instead, actions can be detected by projecting chunks of the 3D spatio-temporal volume into a 2D representation, then performing a 2D search. The advantage of the proposed approach is that this 2D search can be performed using the same techniques that have proven successful in object detection. As an example, Section 3.3.1 shows how the Efficient Sub-Window Search algorithm [11] for object detection can be directly applied to action detection using this technique.

Section 3.3.2 also introduces a novel, straightforward method for searching the 2D projection to localize actions, termed TPSS. As shown in the experiments in Section 4, this approach leads to improved results over the ESS algorithm. In Section 3.4, we also show how these chunks can be chained together to identify the entire extent of the action.

The remainder of the paper is organized as follows. Section 2 reviews the literature in video action recognition and detection. Section 3 introduces the video projection method and discusses two efficient search algorithms, ESS and the novel TPSS algorithm. Section 4 describes our experimental methodology and shows a direct comparison against recent work in action detection. Section 5 concludes the paper.

2. Related Work

Our method is motivated by Schindler and van Gool's observation that action recognition can often be reliably performed in short image sequences [18]. Human action recognition is currently a very active topic (see [7, 17] for recent surveys). The majority of current research, driven by popular datasets such as KTH [19] and UCF YouTube Actions [14], focuses on whole-clip forced-choice classification of video into one of several categories (such as "jog", "run" or "clap"). Bag of visual words techniques, initially adapted from the text retrieval [13] and later the image classification [4] domains apply naturally to this problem since it demands neither spatial nor temporal localization. Video is represented as a count of discretized features aggregated over the clip and then typically passed as a high-dimensional feature vector to a discriminative classifier, such as an SVM; such methods and extensions report strong results (e.g. [5, 19, 20]). Unfortunately, as observed by the object detection community, recognition without localization is of limited utility, and strong approaches to forced-choice recognition do not naturally transfer to detection, due to the asymmetric nature of the detection problem (the prior probability that a given region contains the object of interest is minute). We focus our review of related work to action detection methods that find and localize actions in space and time.

Approaches to action localization are frequently related to successful ideas in the object detection literature. Sliding window approaches, such as [9] apply a cascade of boosted classifiers [22]. Methods such as [1] treat actions as spatio-temporal objects, while others [16] localize using votes in space-time. Template-matching, either in motion-history images [2] or directly against oversegmented space-time volumes [10] has also shown promise. Flow-based [6] and trajectory-based [15] methods exploit either the short- or longer-term motion of points to detect actions.

Our work is strongly influenced by Yuan *et al.* [23] and Cao *et al.* [3], where action detection is performed using an efficient branch-and-bound search. Although we propose a fundamentally different representation and do not explicitly support cross-dataset training, the goals of our work are sufficiently close as to enable direct comparison. Philosophically, our notion of a 2D video projection is also related to concepts such as motion-history images [2] and spin images [8], where 3D spatio-temporal or volumetric data is transformed into a 2D space that enables efficient search or recognition. Finally, one can view video projections as enabling spatial and temporal localization with bag of visual words models by efficiently classifying spatio-temporal subregions.

3. Method

The basic idea behind the proposed method is to treat the action detection task as a series of parallel localization subtasks, each examining a short chunk of the video. Each subtask is transformed into a 2D problem using video projections and solved efficiently. Finally, we connect the local detections in time using a chaining algorithm. The following subsections detail each of these steps.

3.1. Video Representation

Following Laptev *et al.* [12], we extract spatio-temporal interest points (STIPs) in each video and compute the HNF feature descriptor for each STIP by combining histogram of gradient (HoG) and histogram of flow (HoF) features. These are quantized using a visual codebook constructed over the training set, enabling us to represent each STIP p_j as the tuple (x_j, y_j, t_j, c_j), denoting that a STIP was observed at (x_j, y_j) in the t_j'th frame of video; the discrete label c_j corresponds to the codebook word nearest in feature space to p_j's descriptor. The core assumption behind our approach (similar to that in [3, 12, 23]) is that one can recognize whether a collection of STIPs corresponds to the action of interest using a classifier that takes as its input a histogram over these discrete labels. However, we do not accumulate the features into a single histogram, as would be typical in a bag of visual words model, but rather we aggregate them in localized spatio-temporal subvolumes, as described below.

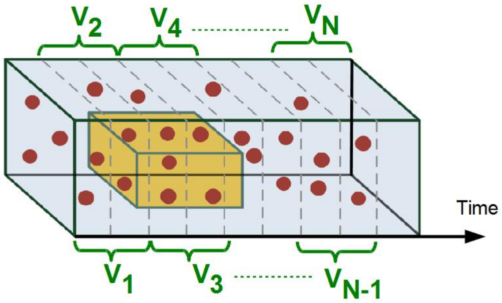


Figure 1: We represent a video sequence as a collection of overlapping video chunks and each video chunk by a collection of STIPs. The goal of action detection is to localize subvolumes that contain the action of interest.

We divide a given video sequence V into a series of overlapping video chunks $\{V_1, V_2, \ldots, V_N\}$ each with a temporal duration of F frames, as shown in Figure 1. Since each STIP retains its spatio-temporal location, the goal of action detection is to find those subvolumes that contain STIPs corresponding to the action of interest. More accurately, since a given action of interest is likely to be span several chunks, we aim to identify subvolumes within each chunk that are likely to be parts of the action.

3.2. Reducing Action Localization to 2D Search

An action can be modeled as a spatio-temporal bounding box (yellow volume in Figure 1). We refer to the subvolume of the action that is contained within a single video chunk as an action segment. Hence, we can consider an action instance as a chain of action segments contained in consecutive video chunks.

We analyze each video chunk independently to determine whether it contains an action segment. Since a chunk consists of only a small number of frames, we seek to localize the action segment only spatially within the chunk by assuming that it extends temporally throughout the chunk. Specifically, as shown in Figure 2, for each action segment in the chunk, we seek a subvolume cuboid of duration F frames that covers it. Since the classifier score of any cuboid in the chunk is determined solely by the STIPs contained within, we observe that rather than exhaustively considering

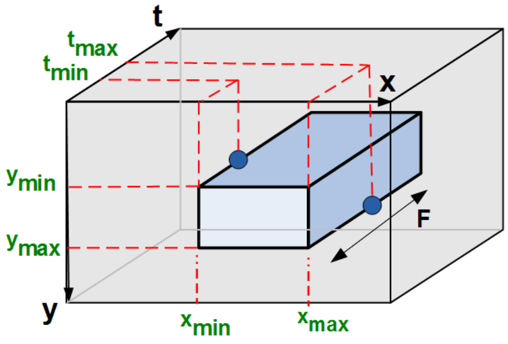


Figure 2: We need only consider subvolumes in a video chunk that touch a STIP on each face; through video projection, we model subvolumes as 2D rectangles.

every cuboid, we only need to consider cuboids that touch STIPs on each face. Since all subvolumes are of duration F, they can be modeled as a 2D rectangle.

Each subvolume is represented in the classifier by a histogram of the codebook counts for the STIPs contained within, $\mathbf{h} = [h_1, \dots, h_K]$, where h$_i$ is the number of STIPs within the subvolume assigned to cluster center i.

$$h_i = \sum_{j=1}^{N} l_{ij} \text{ where } l_{ij} = \begin{cases} 1 & \text{if } c_j = i, \\ 0 & \text{if } c_j \neq i, \end{cases} \tag{1}$$

N is the number of STIPs within the subvolume and the c_j is the cluster index of the jth STIP in the subvolume. We compute the histograms for ground truth action segments that are extracted from the training videos and train a linear support vector machine (SVM) using the resulting histograms.

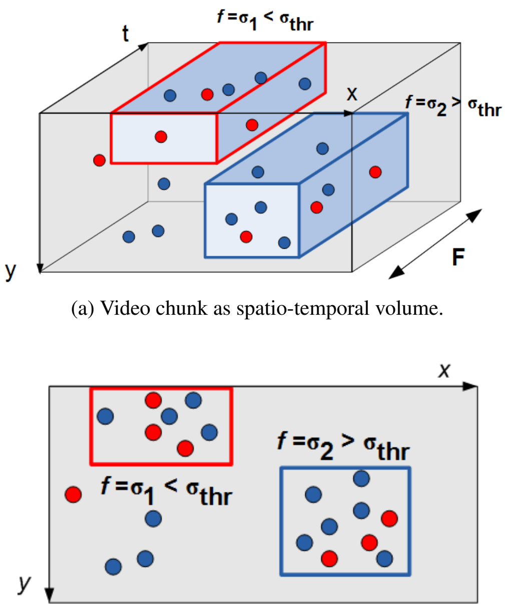
We define the action detection and localization problem within a video chunk as finding a subvolume that would maximize the SVM classifier score, f, given by:

$$f = \beta + \sum_{i=1}^{K} w_i h_i, \tag{2}$$

where K is the number of cluster centers, h_i is the count of STIPs within the subvolume that belong to cluster center i, and w_i is the SVM weight corresponding to cluster center i. Using Equations 1 and the linearity of the scalar product, we can rewrite Equation 2 as follows:

$$f = \beta + \sum_{j=1}^{N} w_{cj}, \tag{3}$$

where N is the number of STIPs contained in the subvolume and w_{cj} is the SVM weight corresponding to the cluster index c_j. Equation 3 says that each STIP contributes to the SVM score by its corresponding SVM weight w_{cj}. Intuitively, some STIPs are positively associated with a given action (their SVM scores are positive) while others are negatively associated with the action (negative SVM score). Thus, the goal is to identify subvolumes containing a high sum of weights.



(a) Video chunk as spatio-temporal volume.

(b) 2-D representation after video projection

Figure 3: Video projection converts subvolume localization to 2D search problem. Blue (Red) points correspond to STIPs with positive (negative) SVM weights.

Since we are using fixed depth subvolumes with same duration as the video chunk and the SVM score of the subvolume depends only on the weight of each STIP within the subvolume (Equation 3), we can redefine the three-dimensional subvolume search problem as a two-dimensional search problem by projecting the data along the temporal dimension, as shown in Figure 3. Figure 3a shows two candidate subvolumes in a video chunk: the blue one has an SVM score greater than the threshold and red one has an SVM score less than the threshold. Figure 3b shows the corresponding subwindows with the same SVM scores in the projected representation. Thus, the subvolume search problem in a video chunk reduces to a subwindow (rectangle) search problem in its 2D projection.

3.3. Subvolume Search in 2D Video Projection

The previous section showed how the subvolume search problem in a video chunk could be reduced to a subwindow search problem in the 2D projection of the video chunk. We can use any two-dimensional search method to find rectangular regions of interest whose SVM scores, as given by

Equation 3, would exceed a specified threshold. One obvious candidate is Lampert *et al.*'s Efficient Subwindow Search (ESS) algorithm [11]. However, given the sparsity of STIP features in our chunks, we propose a new fast method, Two-Point Subwindow Search (TPSS) that outperforms the ESS strategy, as will be shown in Section 4.

3.3.1 Efficient Subwindow Search

Efficient Subwindow Search (ESS), as proposed by Lampert *et al.*, was designed for efficient object localization in images. ESS uses a branch-and-bound algorithm to find the rectangular region of interest with the highest SVM score from Equation 3. In order to detect multiple action instances within the same video chunk, the ESS search can be run multiple times, removing the detected rectangle from I at every iteration until the SVM score of the detected rectangle falls below the threshold.

3.3.2 Two Point Subwindow Search

Even though, theoretically, the search space for an $M \times N$ image using the sliding window approach is M^2N^2, in practice there are only sparse number of STIPs with non-zero SVM weight and only a small fraction of those are positive. Using this observation, we propose a new search method called Two-Point Subwindow Search (TPSS). In this search method, two STIPs with positive SVM weights define a rectangle, with STIPs at opposite corners. The SVM score of such rectangular regions can be efficiently computed using an integral image [22]. Our proposed search method has two stages: (1) we compute the SVM score for all candidate rectangles that are defined by pairs of STIPs, each with positive SVM weight, if the SVM score is over a threshold, then the rectangle is considered as a detection; (2) we perform a non-maximum suppression algorithm on the detection set to eliminate overlapping rectangles.

The primary benefit of the first stage is that the set of candidates (rectangles defined by two STIPs with positive SVM weights) is much smaller than the number of sliding windows (or even rectangles defined by all STIPs). Intuitively, the TPSS algorithm restricts its search to promising rectangles. For instance, consider a rectangle bounded by a STIP with a negative score; shrinking such a rectangle so as to exclude this STIP should result in improving the score.

The second stage performs non-maximum suppression to reduce the number of redundant detections. During non-maximum suppression, we first select the rectangle with the highest SVM score, i.e., R_{max}, from the detection set. Then, we identify and remove all the rectangles from the detection set that are connected to R_{max}. Once the set of rectangles that are connected to R_{max} are found, we replace R_{max} with a bounding box that contains all rectangles that are connected to R_{max}, and push the updated R_{max} to the final detection list. We repeat the process by selecting the next rectangle with the maximum SVM score from the remaining rectangles until no rectangles remain in the detection set. Thus, unlike ESS, TPSS does not require multiple reruns to detect multiple action instances in a video chunk.

3.4. Chaining Subvolumes Across Time

In many applications, it is sufficient to detect and spatially localize any actions occuring during a given interval. If it is necessary to find the entire temporal extent, the detected regions from consecutive video chunks can be assembled into a complete action. We use a greedy strategy that initializes the search with the first video chunk and selects the detection with the highest SVM score. Then, we find all the detections from the next video chunk that are connected to the currently selected detection. We treat two detections in adjacent video chunks as connected if their volume overlaps by a sufficient threshold. We select the detection with the highest SVM score from the connected detections list and continue our search with the next video chunk until there are no more connected detections left in subsequent video chunks.

4. Experimental Results

Our experimental methodology follows that of Cao *et al.* [3] to facilitate direct comparisons. We use the KTH [19] and Microsoft Research Action Dataset II (MSR) [3] in our experiments. The MSR data set contains three action types: hand waving, hand clapping and boxing, performed in a more challenging setting with multiple users in a cluttered and dynamic scene. By design, the three actions in MSR are the same as those in KTH, to explore cross-dataset performance of action detection algorithm. The MSR dataset contains 54 video clips, with each clip exhibiting several instances of each action type (71 waving, 51 clapping and 81 boxing action instances).

Following [3], our training set consists of the waving, clapping and boxing clips from KTH augmented with four randomly-selected video clips from the MSR dataset. The testing dataset consists of the remaining videos in MSR. We first construct a standard vocabulary using K-means clustering ($K = 1000$) on HNF descriptors computed at space-time interest points (STIP) [12] extracted on the training set, where the descriptors are a compound of HOG and HOF [12] features.

Next, we train a set of linear one-vs-all SVM classifiers using videos from KTH and the ground truth volumes from the MSR training subset. For each, we extract overlapping video chunks with durations of F frames and compute a bag-of-video-words histogram by accumulating the counts of HNF descriptors in the volume, quantized using the above dictionary. Two factors affect the choice of the frame size, F: classifier performance and localization performance. Choosing too small an F could degrade classifier performance since there would only be a few interest points in a video chunk. On the other hand, choosing too large a value for F could increase the localization error due to quantization in the temporal domain. In our experiments we set the F to 32 frames, which provides a good classification performance and acceptable localization error. We employ randomly extracted video chunks of the same size as negative examples and expand this negative sample set using bootstrapping [21].

To compare our results to the ones in Cao *et al.* [3] we use the same precision and recall criteria. For the precision score, a detection is regarded as a true positive if at least 1/8 of its volume overlaps with that of the ground truth. For recall, the ground truth label is regarded as retrieved if at least 1/8 of its volume is covered by one of the detected volumes. The precision and recall values are computed for different SVM thresholds to generate the P-R curves shown below.

4.1. Cross-Dataset Action Detection Comparisons

Figure 4 shows the precision-recall curves for Cao *et al.* [3] and the two search variants of the proposed video projection (VP) method: VP+ESS and VP+TPSS. We make several observations. First, we see that in general, both VP methods outperform [3], particularly in the high recall regime. Second, we observe that the proposed two point subwindow search (TPSS) appears to be slightly better than ESS [11] on this task, in spite of its simplicity. Finally, we note that even though our approach is not explicitly designed with cross-dataset action detection in mind, it surprisingly outperforms [3] on the task, prompting us to investigate this issue further.

4.2. Additional Results

Our last set of experiments explores a few remaining questions, such as: how well does our method perform with shorter-duration video chunks (smaller values of F)? Figure 5 shows the precision/recall curves for the waving action (trained as in Section 4.1) for different chunk lengths. We observe that while the performance is reduced for smaller window sizes, it is still reasonable. This indicates that the proposed approach should be suitable for online recognition systems, where low latency is essential; in such settings, a detection can be flagged using only a portion of the action of interest.

Finally, Figure 6 shows examples of action detections on the MSR Actions II dataset using VP+TPSS.

5. Conclusion

Detecting and localizing actions in video is daunting when it is posed as finding an optimal 3D subvolume in a much larger video volume. This paper demonstrates that action detection and localization can instead be decomposed into a series of independent localization subtasks on relatively small chunks of video. Moreover, using video projections, we transform the detection task in each chunk into a 2D search problem that can be efficiently solved.

As we have demonstrated, the primary benefit of this approach is that leading methods in object detection can be applied directly to the action detection problem. As shown in our experiments, our proposed method is not only straightforward to implement, but also leads to improved results in a direct comparison. In future work, we plan to leverage the flexibility of our chained representation to tackle more complex actions, such as those performed by moving actors in dynamic scenes.

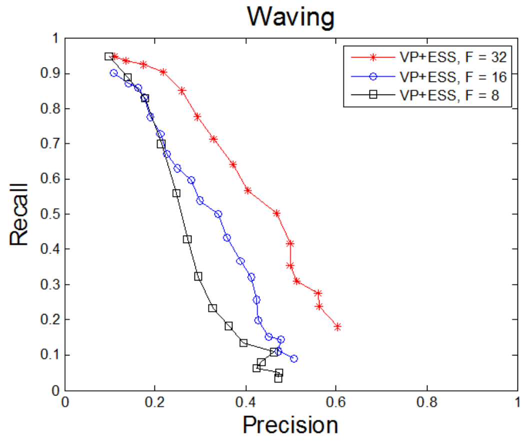


Figure 5: VP+ESS on the waving action for different video chunk sizes. The system performs best when longer chunks are used, but performance is acceptable if smaller chunks are needed, such as for on-line recognition.

Acknowledgments

M.F.T. was supported by NSF grants IIS-0905387, IIS-0916868, and a grant under the NGA NURI program.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proceedings of International Conference on Computer Vision*, 2005. 35

[2] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3), 2001. 35

[3] L. Cao, Z. Liu, and T. Huang. Cross-dataset action detection. In *CVPR*, 2010. 34, 35, 37, 38, 39

[4] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision*, 2004. 34

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Workshop on VS-PETS*, 2005. 35

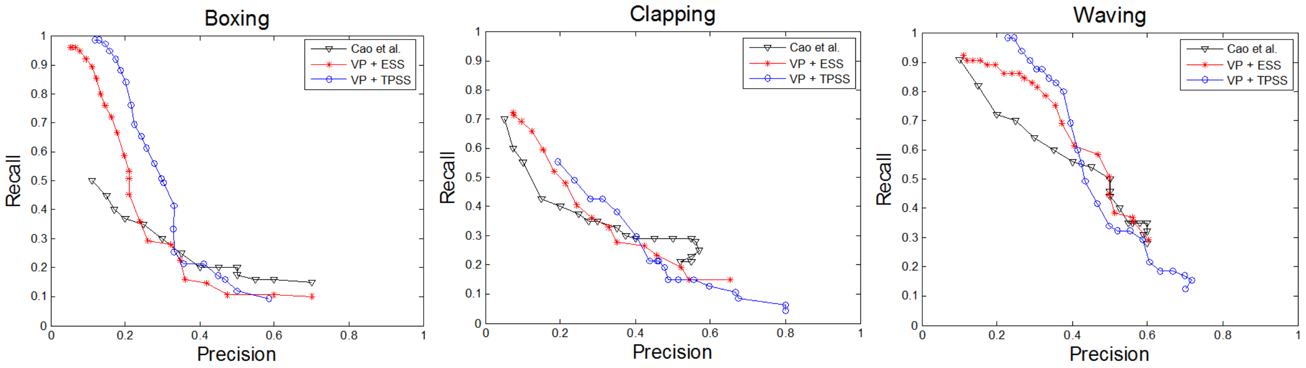


Figure 4: Direct comparison on MSR Action II, trained using KTH+4 clips of MSR. Both variants of proposed method, VP+ESS and VP+TPSS, outperform Cao *et al*. [3] despite their simplicity.

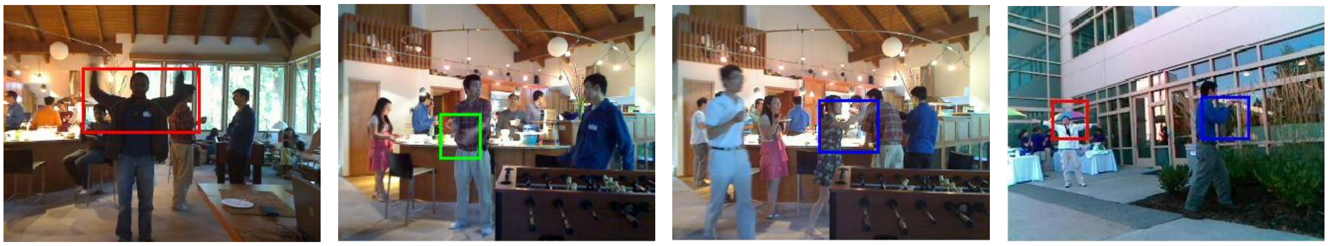


Figure 6: Examples of action detections on MSR dataset using VP+TPSS. Colored boxes denote detected actions: boxing (blue), clapping (green), and waving (red). Qualitatively, our results are comparable to Cao *et al*. [3].

[6] A. Efros, C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proceedings of International Conference on Computer Vision*, 2003. 35

[7] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Systems Man and Cybernetics, Part C*, 34(3), 2004. 34

[8] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 1999. 35

[9] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proceedings of International Conference on Computer Vision*, 2005. 35

[10] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *Proceedings of International Conference on Computer Vision*, 2007. 35

[11] C. Lampert, M. Blaschco, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008. 34, 37, 38

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 35, 37

[13] D. Lewis. Naive Bayes at Forty: The independence assumption in information retrieval. In *European Conference on Machine Learning*, pages 4–15, 1998. 34

[14] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from “videos in the wild”. In *CVPR*, 2009. 34

[15] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *Proceedings of International Conference on Computer Vision*, 2009. 35

[16] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *CVPR*, 2008. 35

[17] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010. 34

[18] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, 2008. 34

[19] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Proceedings of International Conference on Pattern Recognition*, 2004. 34, 35, 37

[20] X. Sun, M. y. Chen, and A. Hauptmann. Action recognition via local descriptors and holistic features. In *CVPR workshop on Human Communicative Behavior Analysis*, 2009. 35

[21] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 1998. 38

[22] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 35, 37

[23] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009. 34, 35