MediaCCNY at TRECVID 2012: Surveillance Event Detection

Xiaodong Yang[†], Chucai Yi[§], Liangliang Cao[‡], and YingLi Tian[†§]

[†]Department of Electrical Engineering
The City College of New York, CUNY
{xyang02, ytian}@ccny.cuny.edu

[§]Department of Computer Science
The Graduate Center, CUNY
cyi@gc.cuny.edu

[‡]Multimedia Analytics Group
IBM T. J. Watson Research Center
liangliang.cao@us.ibm.com

Abstract. In this paper, we present a general event detection system evaluated by the Surveillance Event Detection (SED) task of TRECVID 2012 campaign. The proposed system is evaluated on all the seven event categories of the SED task. In our system, a sliding temporal window is employed as the detection unit, which is represented by a histogram of spatio-temporal features including STIP-HOG/HOF and SURF/MHI-HOG. We also investigate the spatial priors of various events by estimating spatial distributions of actions under different camera views in the training data. As non-linear SVMs usually have superior performances but in general are much slower in both training and testing, we therefore employ explicit feature maps to approximate large scale non-linear SVMs by linear ones. In order to deal with highly imbalanced data, our system performs detections by a set of cascade linear SVMs that are learned corresponding to specific events and camera views.

1 Introduction

Automatic event detection of video surveillance has many real-world security applications for public areas including airports, banks, supermarkets, etc. In the past decades, research of human action recognition mainly experiments on relatively simple and clear scenes where only limited actors with definite actions present. This constrained scenario seldom holds in real-world surveillance videos due to challenges of large variances of viewpoint, scaling, lighting, cluttered background, etc. To bridge research efforts and real-world applications, TRECVID [9] provides the Surveillance Event Detection (SED) task to evaluate event detection in real-world surveillance settings. In TRECVID 2012, SED provides a corpus with 144-hour videos under five camera views from the London Gatwick International Airport. In this dataset, 99-hour videos can be used as the development set with annotations of temporal extents and event labels. Our system is evaluated on all the seven events, i.e., CellToEar, Embrace, ObjectPut, PeopleMeet, PeopleSplitUp, PersonRuns, and Pointing.

The remainder of this paper is organized as follows. Section 2 introduces the overall system architecture. In Section 3 and 4, we provide detailed procedures of feature extraction and video representation. Section 5 describes the cascade SVMs algorithm and post processing. A variety of experimental results and discussions are presented in Section 6. Finally, Section 7 summarizes the remarks of our system.

2 System Architecture

As demonstrated in Fig. 1, our system includes 4 main components: (1) low-level feature extraction, (2) video (sliding window) representation, (3) learning event models, and (4) post processing to localize event temporal extents.

Figure 1: MediaCCNY surveillance event detection system architecture.

Most recent work on action recognition demonstrated that local spatio-temporal features are more robust to posture, occlusion, illumination, and cluttered background compared to global features. A spatio-temporal feature usually includes two phase: detection (i.e., a feature detector localizes interest points in a spatio-temporal space) and description (i.e., a feature descriptor computes representations of detected points). The Space-Time Interest Point (STIP) [5] employed 3D Harris corner detector to detect points with large gradient magnitude in both spatial and temporal domains. Histogram of Gradients (HOG) and Histogram of Optical Flow (HOF) were then calculated as descriptors. However, it is quite restrictive to have large intensity changes in both spatial and temporal dimensions, which often results in insufficient point detections. Instead of using spatio-temporal volumes, we proposed to extract spatial and temporal information separately as in [10]. The detector of the Speed Up Robust Features (SURF) [1] is first applied to extract visually distinctive points in the spatial domain. These SURF points are then filtered by temporal (motion) constraints from the Motion History Image (MHI) [2] that is generated by differencing adjacent frames. Those pixels with larger intensities in MHI represent moving objects with more recent motion. We only select those SURF points with most recent motions as detected interest points. To characterize shape and motion information, we compute HOG features for each interest point from both image channel and MHI channel. In

our system, STIP-HOG/HOF and SURF/MHI-HOG are used as the low-level features to represent human actions. In addition, because of specific camera views and scenes, the occurrence of specific events is usually biased to a certain range of locations. We further make use of this spatial prior to eliminate a large amount of interest points from background.

The Bag-of-Words (BOW) is used to organize low-level features to represent each sliding window. This approach commonly consists of two phases, i.e., feature coding and feature pooling. In our system, a visual codebook with the size of 3000 is first computed by KMeans. We then employ the local soft assignment scheme [6] to code low-level features. The local soft assignment coding is able to achieve comparable performance but with much less computational cost compared to other much more complicated coding methods such as sparse coding, locality-constrained linear (LLC) coding, super vector coding, and Fisher coding [3]. After feature coding, we choose the max pooling to aggregate coded features. Before learning event models, we first apply the explicit feature maps [11] to these BOW features. This is motivated by approximating large scale non-linear SVMs through linear ones which enjoy much more computational efficiency in both training and testing.

Having obtained above video representations, we can learn event models by linear SVMs solvers. However, the data is highly imbalanced because positive events are far less frequent than negative ones. Therefore, we propose a cascade SVMs algorithm to overcome this high imbalance. In each stage of this algorithm, positive and negative samples with the same amount are used to train a classifier that favors to positive samples. This leads each individual classifier to a high detection rate but also a high false alarm rate. By cascading multiple classifiers (e.g., 5-7), we are able to filter out considerable false alarms but maintain a reasonable detection rate.

A post processing is performed over the classifier predictions to determine temporal localization of each event and further remove false alarms. It is assumed that most positive samples would continuously last for a certain number of frames as temporal extents of most events could cover several sliding windows. We therefore merge neighboring positive predictions into a single positive detection. Based on our empirical observation, we also remove those isolated positive predictions or other positive ones mixed with too many negative predictions.

3 Feature Extraction

We extract two types of low-level features including STIP-HOG/HOF and SURF/MHI-HOG. STIP detects interest points by searching significant variations in both space and time. SURF/MHI detects interest points with spatially distinctive shapes and temporally sufficient motions. The two detectors therefore provide complementary interest points. STIP detector combined with HOG/HOF descriptors has been widely used in action recognition and detection tasks [10]. However, it often suffers insufficiency due to rigorous assumptions of large gradients in space and time, as well as inefficiency because of computational cost in its detector and descriptor. The method proposed in [10] solved the two flaws and further provided complementary feature to STIP-HOG/HOF.

MHI is a real-time motion template generated by stacking consecutive frame differences [2]. The brighter pixels on MHI correspond to more recent motion. MHI gradients also reflect directional information of human action. 2D Harris corner detector combined with temporal information from MHI was used in our previous work [10] to recognize actions in cluttered videos. In this system, we employ a SURF detector to spatially localize interest points. Compared to 2D Harris detector, SURF detector is able to generate additional scale information and maintain computational efficiency. The dominant orientations of interest points are however discarded as motion directions also provide important clue for action recognition. MHI is then used as a motion mask to remove interest points from static background, i.e., only SURF points with more recent motions or large MHI intensities are chosen as interest points. Fig. 2 demonstrates interest points detected by SITP and SURF/MHI. As shown in this figure, SURF/MHI provides denser and complementary interest points to STIP. In addition, SURF/MHI detects fewer points from background.

HOG (72 dimensions) and HOF (90 dimensions) descriptors are able to represent local appearance and motion properties respectively in STIP volumes. HOG can be also well adapted to characterize local shape information from image channel and local motion information from MHI channel by computing distributions of local gradients. In our system, each support region associated with an interest point on image and MHI channels is subdivided into 4×4 grids. Image and MHI gradients are evenly sampled in 8 orientation bins. So each SURF/MHI-HOG point generates a feature vector of 2×4×4×8 = 256 dimensions.

The comparisons on computational costs and detected number of interested points of the two methods are listed in Table 1. The statistics are based on a portion of TRECVID-08-Dev-Set. As shown in this table, SURF/MHI-HOG is over 10 times faster in terms of processing each frame and about 20 times faster in terms of computing each interest point than STIP-HOG/HOF. The number of detected points per frame of SURF/MHI-HOG is about 2 times as that of STIP-HOG/HOF.

Table 1: Comparisons between STIP-HOG/HOF and SURF/MHI-HOG.

Feature	Speed (frames / sec)	Speed (ms / point)	Number (points / frame)
STIP-HOG/HOF	0.6	29.9	56
SURF/MHI-HOG	6.4	1.5	107

Figure 2: Interest points detected by STIP and SURF/MHI. Brighter pixels on MHI correspond to more recent motion. Gradients on MHI also provide motion direction information.

Due to highly cluttered background, a significant amount of interest points are detected from irrelevant actions. In order to remove those noisy points, we build hot region masks based on spatial priors of specific events and cameras. As the surveillance videos were recorded by fixed cameras in specific public areas, we observe the occurrence of specific events concentrates in some specific regions as shown in Fig. 3. The bounding boxes of people performing actions are annotated to construct these hot regions. A hot region map $H_{c,e}$ of camera view c and event e is obtained by $H_{c,e} = \left(\sum_i A^i_{c,e}\right)/N_{c,e}$, where $A^i_{c,e}$ is the ith annotated frame (a binary map) in camera view c and event e with foreground pixels in a bounding box region, and $N_{c,e}$ is the total number of annotated frames for camera view c and event e. This spatial prior information can be used to distinguish action and non-action regions by thresholding $H_{c,e} > \mu_{c,e}$. The interest points from non-action regions are removed in the following process as illustrated in Fig. 1.

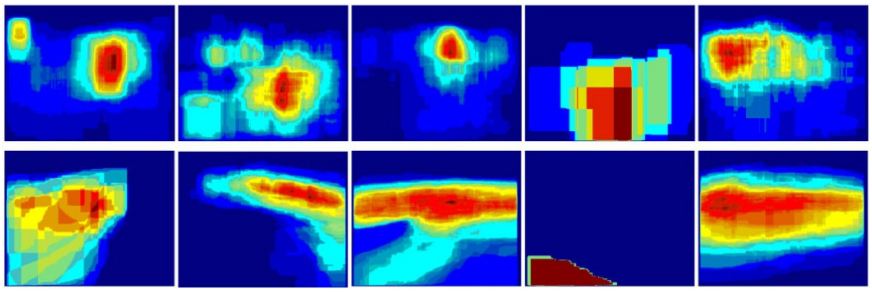

Figure 3: Examples of hot regions of event ObjectPut (top) and event PersonRuns (bottom) corresponding to camera views of 1-5 (from left to right).

4 Video Representation

We employ BOW combined with spatial pyramids [7] to represent each sliding window. A codebook $B_{d\times n} = (b_1, b_2, \ldots, b_n)$ for each kind of low-level features is computed by KMeans. The codebook size n is set to be 3000 and $b_i \in \mathcal{R}^d$ is a visual word with feature dimension d. The local soft assignment scheme [6] is used to code each feature x_i to u_i by:

$$u_{i,j} = \frac{\exp\left(-\beta \hat{d}(x_i, b_j)\right)}{\sum_{l=1}^{n} \exp\left(-\beta \hat{d}(x_i, b_l)\right)},$$

$$\hat{d}(x_i, b_l) = \begin{cases} d(x_i, b_l) & \text{if } b_l \in \mathcal{N}_k(x_i) \\ +\infty & \text{otherwise} \end{cases} \qquad (1)$$

where $u_{i,j}$ denotes the jth coefficient of code u_i, $\hat{d}(x_i, b_l)$ is a local version of the original distance $d(x_i, b_l) = \|x_i - b_l\|^2$, $\mathcal{N}_k(x_i)$ are the k-nearest neighbors of x_i measured by $d(x_i, b_l)$, and β is a smoothing factor. In our system, we empirically set $k = 200$ and $\beta = -1$.

The max pooling is then used to aggregate coded features u_i within a spatial grid w_m of a sliding window to a histogram h_m by:

$$h_{m,j} = \max_{i \in w_m} u_{i,j} \text{ , for } j = 1, 2, \dots n \qquad (2)$$

where $h_{m,j}$ denotes the jth coefficient in h_m, and $w_m, m = 1,2, \dots, M$ is the mth spatial grid. As two levels of grids (i.e. 1×1 and 2×2) are used, each sliding window generates $M = 5$ spatial pyramids tiles as shown in Fig. 4. The five histograms h_m are then concatenated into h as the BOW representation for each sliding window.

In our system, a sliding window with the size of 60 frames steps in every 15 frames. This generates a great amount of data, e.g., 600K samples from training set. It is infeasible for non-linear SVMs to learn or evaluate on such large scale data. On the other hand, linear SVMs are in general much faster in both training and testing. However, they usually have inferior accuracies then non-linear ones. In order to solve this difficulty, we approximate non-linear kernel distances by the explicit feature maps [11] to enable more efficient linear SVMs with little loss in accuracy. The basic idea of this method is to lift each feature vector $h \in \mathcal{R}^D$ to a feature space with moderately higher dimensions through an explicit feature map $\psi: \mathcal{R}^D \to \mathcal{R}^{D(2r+1)}$ such that inner product in this space can approximate well the non-linear kernel distance $\mathcal{K}$, i.e., $\langle \psi(h), \psi(h') \rangle \approx \mathcal{K}(h, h')$. We set $r = 3$ in our system to approximate the χ^2 kernel. After the feature mapping, $\psi(h)$ with a dimension of 105K becomes the video representation of each feature type.

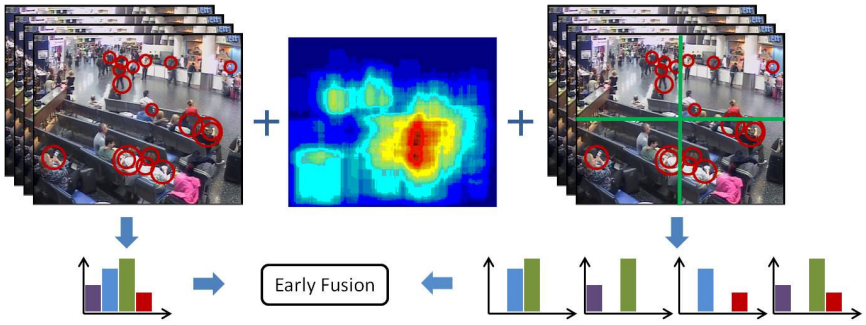

Figure 4: The spatial pyramids of 2 levels with 1×1 and 2×2 tiles in each level. A camera and event dependent hot region map is used to remove points from background in coding and pooling process.

5 Learning Event Models and Post Processing

As the sliding window scheme in our system generates quite imbalanced data, e.g., negative samples are over 60 times than positive ones, we propose a cascade SVMs algorithm to handle this high imbalance. The camera and event dependent models are learned to reduce intra-class variance and memory consumption in training. So our system includes 35 models for 7 events under 5 camera views.

Algorithm 1: Learning event model by cascade SVMs

Input: a training set $S \coloneqq \{S^+, S^-\}$ and maximum iteration number c.
Initialization: $C_0 \coloneqq \{\}$ and $S_1^- \coloneqq$ {randomly select $|S^+|$ samples from S^-}.
for $i \coloneqq 1$ to c **do**
  $w^+ \coloneqq 1$ and $w^- \coloneqq 1$.
  for $j \coloneqq 1$ to t **do**
    $M_i \coloneqq$ **LIBLINEAR** (S^+, S_i^-, w^+, w^-).
    positive accuracy $\coloneqq M_i\ (S^+)$.
    if positive accuracy > θ
      break.
    end
    $w^+ = w^+ + \tau$.
  end
  $C_i \coloneqq C_{i-1} \cup M_i$.
  $S^- \coloneqq \{s \mid s \in S^-$ and $C_i(s) =$ positive$\}$.
  SORTSAMPLES (S^-).
  $S_{i+1}^- \coloneqq S^-(1, 2, \dots, \text{num})$, where num $\coloneqq \min(|S^+|, |S^-|)$.
  if $|S_{i+1}^-| < |S^+|$
    break.
  end
end
Output: the camera dependent event model C, i.e., cascade linear SVMs.

Suppose we have a training set $S = \{S^+, S^-\}$ for each event under each camera view. The cascade SVMs algorithm adaptively divides the negative set into a series of partitions S_i^- with the same size of $|S^+|$ according to the ranked prediction scores and iteratively learns a group of binary SVMs classifiers M_i that favors to positive samples. These classifiers are cascaded as the event model $C = \{M_1, M_2, \dots, M_{|C|}\}$. The outline of our proposed learning process is shown in Algorithm 1.

The classifier M_i in each stage is learned by an adaptive weighting method. This is used to ensure a classifier in each stage can correctly predict most positive samples. We initialize both positive and negative weights as 1. After training a classifier by LIBLINEAR [4], we evaluate this classifier only on positive samples. If the accuracy is greater than a threshold $\theta = 0.9$, this classifier is assigned to M_i. If not, positive weight is increased by $\tau = 0.05$ and the classifier is retrained with updated class weights. This process is repeated until the accuracy on positive samples is up to θ or the maximum iteration number t is reached. As positive samples are much fewer than negative ones, we employ all positive samples for training. In each stage, the negative samples S_i^- are kept the same amount as positive ones. To update S_i^-, we first filter S^- by only preserving those negative samples that cannot be correctly classified using current classifier pool C_i. We then sort the left negative samples from S^- in descending order based on their scores and choose the first $|S^+|$ samples (more confusing ones) as S_{i+1}^-. This cascading process terminates if the maximum iteration number $c = 10$ is reached or the left negative samples are fewer than positive ones. In our system, the size $|C|$ of event models is between 5 and 10.

Because of the sliding window scheme used in our system, an event might be chopped into several different windows. Therefore after classifier predictions, we

employ a post processing to group continuous positive windows to determine the final temporal localization of a detected event. In the merging process, we use a tolerance γ_t (3 used in our system) which means two positive predictions disconnected by less than γ_t negative predictions can still be merged together. The other benefit of post processing is to further remove false alarms. After the merging process, a group will be removed from positive detections if the ratio of negative predictions (holes) in a merged group is greater than γ_h (0.3 used in our system).

6 Experimental Results

In TRECVID SED 2012, 15-hour of videos with frame resolution 720×576 at 25fps captured by 5 fixed cameras are provided as the evaluation set. This is a subset (about 1/3) of the evaluation set in 2011.

We first compare our results with last year's best results [8] by the primary metric Actual DCR (ADCR) and the secondary metric Minimum DCR (MDCR) in Table 2. We achieve the best ADCR in the event CellToEar. The performances of our system evaluated in all 7 events are among the top 3 compared to other systems in 2011. Table 3 presents comparisons between our system and the best systems in 2012. The rank column denotes our rankings among all participants in terms of ADCR.

Table 2: Comparisons between our results and the best results in 2011.

Event	Actual DCR		Minimum DCR	
	2011 Best	Ours	2011 Best	Ours
CellToEar	1.0365[1]	1.0086	1.0003[1]	1.0003
Embrace	0.8840[1]	0.9552	0.8658[1]	0.9351
ObjectPut	1.0006[2]	1.0158	0.9983[2]	1.0003
PeopleMeet	0.9820[2]	1.0082	0.9724[1]	0.9885
PeopleSplitUp	0.9099[3]	0.9843	0.8809[5]	0.9787
PersonRuns	0.8924[1]	0.9702	0.8370[1]	0.9623
Pointing	0.9783[4]	1.0895	0.9730[4]	0.9987

[1]the result attributes to CMU, [2]the result attributes to PKUNEC, [3]the result attributes to TokyoTech-Canon, [4]the result attributes to BJTU-SED, [5]the result attributes to BUPT-MCPRL.

Table 3: Comparisons between MediaCCNY and TRECVID SED best systems in 2012.

Event	Rank	Best 2012 System ADCR	MediaCCNY Primary Run				
			ADCR	MDCR	#CorDet	#FA	#Miss
CellToEar	3	1.0007[1]	1.0086	1.0003	1	42	193
Embrace	4	0.8000[1]	0.9552	0.9351	20	212	155
ObjectPut	3	0.9983[2]	1.0158	1.0003	1	53	620
PeopleMeet	2	0.9799[2]	1.0082	0.9885	14	120	435
PeopleSplitUp	3	0.8433[1]	0.9843	0.9787	6	50	181
PersonRuns	2	0.8346[1]	0.9702	0.9623	6	80	101
Pointing	5	0.9813[3]	1.0895	0.9987	29	356	1034

[1]the result attributes to CMU-IBM, [2]the result attributes to PKUNEC, [3]the result attributes to BJTU-SED.

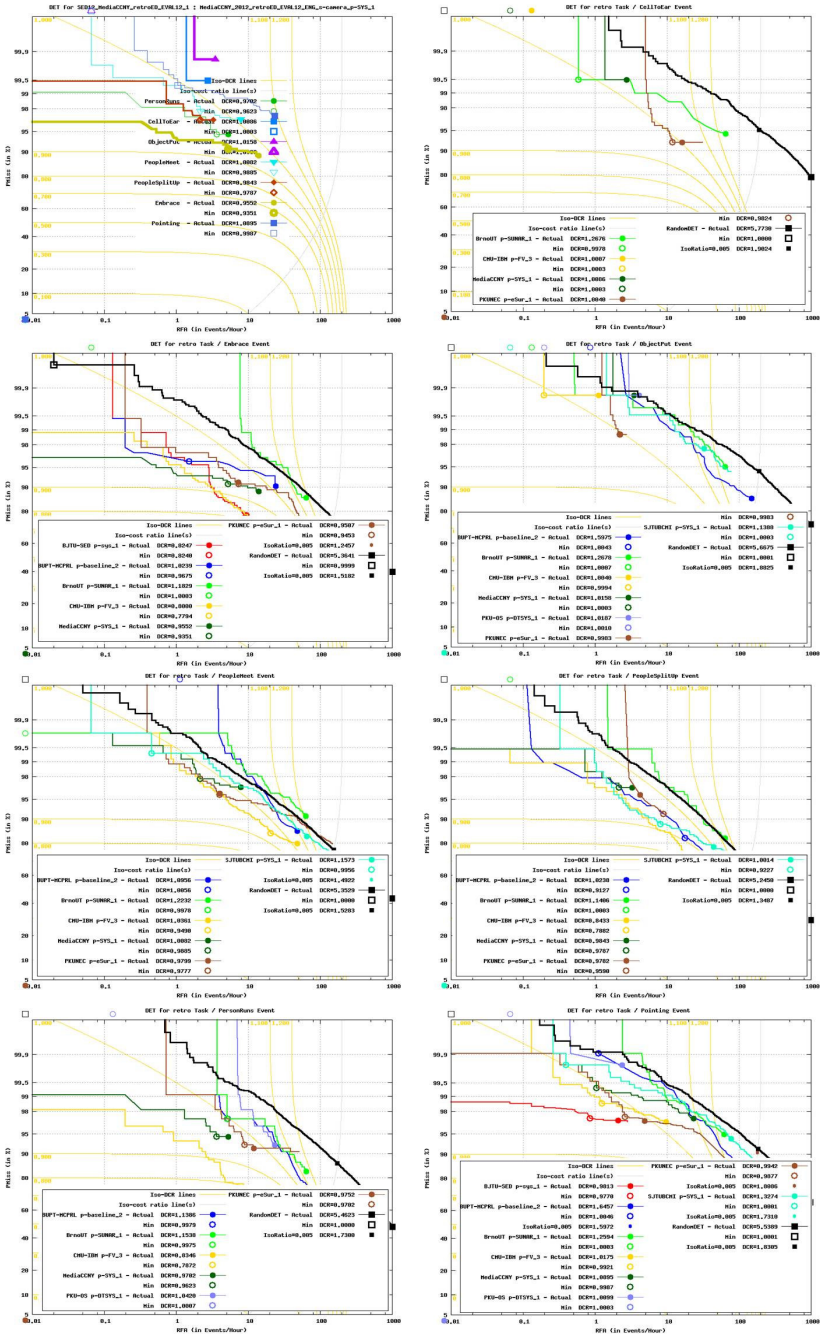

Figure 5: The Detection Error Tradeoff (DET) curves of our system and each event.

The Detection Error Tradeoff (DET) curves of our system and other participants in all events are demonstrated in Fig. 5. These curves represent event-averaged miss detection probabilities vs. false alarm rates through varying a detection threshold.

7 Conclusion

In this paper we have presented detailed implementations of the MediaCCNY system in TRECVID 2012 SED. Our system starts from extracting STIP-HOG/HOF and SURF/MHI-HOG features of each sliding window. The local soft assignment coding and max pooling are then employed to aggregate low-level features. To make use of large scale linear SVMs, we further apply explicit feature maps to approximate non-linear kernels. A sliding window is classified by event models that are learned using our proposed cascade SVMs algorithm. The prediction results are in the end post processed to localize temporal extents of detected events. Our system is evaluated in all 7 events and achieves top 3 performances in 5 events.

Acknowledgments. This work was supported in part by ARO W911NF-09-1-0565.

References

1. H. Bay, A. Ess, and L. Gool. SURF: Speed Up Robust Features. *Computer Vision and Image Understanding*, 2008.
2. A. Bobick and J. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2001.
3. K. Chatfield, V. Lemtexpitsky, A. Vedaldi, and A. Zisserman. The Devil Is in the Details: An Evaluation of Recent Feature Encoding Methods. *British Machine Vision Conference*, 2011.
4. R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 2008.
5. I. Laptev. On Space-Time Interest Points. *International Journal of Computer Vision*, 2005.
6. L. Liu, L. Wang, and X. Liu. In Defense of Soft-Assignment Coding. *International Conference on Computer Vision*, 2011.
7. S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
8. M. Michel, J. Fiscus, and P. Over. TRECVID 2011 Video Surveillance Event Detection Task. *NIST TRECVID Workshop*, 2011.
9. A. Smeaton, P. Over, and W. Kraaij. Evaluation Campaigns and TRECVID. *ACM Workshop on Multimedia Information Retrieval*, 2006.
10. Y. Tian, L. Cao, Z. Liu, and Z. Zhang. Hierarchical Filtered Motion for Action Recognition in Crowded Videos. *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 2011.
11. A. Vedaldi and A. Zisserman. Efficient Additive Kernels via Explicit Feature Maps. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012.