CCNY at TRECVID 2014: Surveillance Event Detection

Yang Xian, Xuejian Rong, Xiaodong Yang, and Yingli Tian

Graduate Center and City College
City University of New York
yxian@gc.cuny.edu, {xrong, xyang02, ytian}@ccny.cuny.edu

Abstract. In this paper, we present two video-based event detection systems developed by City College of New York (CCNY) for the Surveillance Event Detection (SED) task of TRECVID 2014. One is a generic event detection system that is applied to all the events of the SED task except CellToEar event. In this proposed system, the detection unit is differentiated by a sliding temporal window and a set of spatio-temporal features including STIP-HOG/HOF, DT-Trajectory, and DT-MBH. Fisher Vector is adopted to encode low-level features as the representation of each sliding window. Since the surveillance data is highly imbalanced, we chop the training data into balanced small chunks, and within each data chunk a random forest classifier is learned. In the testing phase, decision-level fusion is applied to combine the prediction results by multiple random forests. The second system is specifically designed to the CellToEar event since it has distinct properties which are unsuited for traditional action based approaches but well compatible with the static gesture detections.

1. Introduction

Event detection aims at recognizing and localizing specified spatio-temporal patterns in videos [1]. There has been a demanding need for the automatic event detection of event surveillance for security and safety concerns, both within home area and public regions such as airports, supermarkets, and commercial establishment. Research of human action recognition in the past decades mainly experiments on controlled environment with clear background where explicit actions are performed with limited actors. However, in real-world surveillance videos, due to challenges of large variances of viewpoint, scaling, lighting, cluttered background, the ideal situation seldom holds. To bridge research efforts and real-world applications, TRECVID [2, 3, 16] sets the Surveillance Event Detection (SED) task to evaluate event detection in real-world surveillance settings. In TRECVID 2014, SED provides a corpus with 144-hour videos from the London Gatwick International Airport under five camera views. In this dataset, 99-hour videos can be used as the development set with annotations of temporal extents and event labels. We design two event detection systems: a specific system for CellToEar and a generic system for all the rest events, i.e., Embrace, ObjectPut, PeopleMeet, PeopleSplitUp, PersonRuns, and Pointing.

The rest of this paper is organized as follows. Section 2 introduces our generic event detection system which includes low-level feature extraction, video representation, the random forests classification, and post processing. In Section 3, we

provide the detailed descriptions regarding the CellToEar task-specific system. Experimental results and discussions are presented in Section 4. Section 5 summarizes the remarks of our systems.

2. Generic Event Detection System

In this section, we present the generic event detection system which is applied to events: Embrace, ObjectPut, PeopleMeet, PeopleSplitUp, PersonRuns, and Pointing.

2.1. System Overview

As demonstrated in Fig. 1, our system is consisted of three major components: (1) low-level feature extraction, (2) video (sliding window) representation based on Fisher Vector, and (3) event learning and prediction by Random Forests.

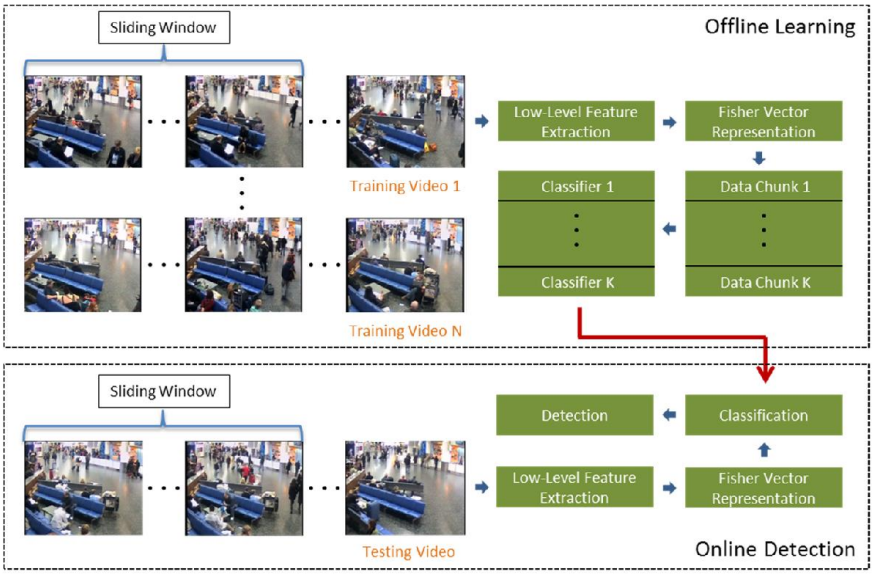


Figure 1: Overview of the CCNY generic surveillance event detection system.

Local spatio-temporal features have been demonstrated to be more robust to posture, occlusion, illumination, and cluttered background compared to global features. Detection and description are two phases in the spatio-temporal feature extraction process. A feature detector localizes interest points in a spatio-temporal space while a feature descriptor computes representations of detected points.

In the generic framework, we use the same low-level features as our previous system [3] which are STIP-HOG/HOF, DT-Trajectory, and DT-MBH. In events PersonRuns and Embrace, all three features are employed to characterize human

actions. Due to shortage of time, for the rest four events, only STIP-HOG/HOF and DT-Trajectory are extracted.

Feature encoding is commonly used to aggregate the low-level features to represent images and videos. The superiority of Fisher Vector has been demonstrated in the evaluation of recent feature encoding methods [4]. In this paper, Fisher Vector with spatial pyramids [5] are adopted to encode local spatio-temporal features. To decorrelate data and reduce the computational burden and memory consumption, we apply PCA for the dimensionality reduction by half over STIP-HOG/HOF and DT-MBH before Fisher Vector representation.

With the above video representations, the event models can be learned by Random Forests [6]. However, the surveillance data is highly imbalanced because positive events are far less frequent than negative ones (refer to Table 1 for details). Therefore, in the offline learning phase, the imbalanced data is chopped into smaller chunks which are relatively more balanced. A Random Forest classifier is learned for each data chunk. A simple post processing is performed as to combine all the prediction results by multiple Random Forests in the online detection process.

2.2. Low-Level Feature Extraction

Similar to our previous framework [3], we extract three types of low-level features including STIP-HOG/HOF, DT-Trajectory, and DT-MBH. In this subsection we briefly introduce the three features respectively. Please refer to [3] for more detailed description.

Space-Time Interest Point (STIP) [7] employs 3D Harris corner detector to detect sparse points with large gradient magnitude in both spatial and temporal domains. Histogram of Gradients (HOG) and Histogram of Optical Flow (HOF) are then computed and concatenated as descriptors based on the space-time neighborhoods of detected interest points to capture the local appearance and motion information.

STIP detector combined with HOG/HOF descriptors has been widely used in action recognition and detection tasks [8]. However, it is restrictive to have large intensity changes in both spatial and temporal dimensions. On the other hand, Dense Trajectories (DT) [9] provides an alternative to the joint space-time based interest point detectors. It densely samples interest points at multiple spatial scales. Then the sampled interest points are tracked over a dense optical flow field and reinitialized every few frames. Two local descriptors, Trajectory and Motion Boundary Histogram (MBH) are then extracted from the space-time volumes aligned with the trajectories. DT-Trajectory characterizes the shape of a trajectory that is used to capture local motion cues. For DT-MBH, the space-time volume aligned with a trajectory is used to extract local descriptors.

2.3. Video Representation

After extracting the low-level features, we perform PCA to reduce the feature dimensions of STIP-HOG/HOF and DT-MBH by half. Then Fisher Vector [10] combined with spatial pyramids [5] are employed to represent each sliding window. Fisher Vector describes each feature descriptor by its deviation with the respect to the parameters of a generative model and provides a feature aggregation scheme based on

Fisher kernel that shares the benefits of both generative and discriminative models. Then the spatial pyramids are employed to roughly incorporate the spatial layout of the video scene.

2.3.1. Fisher Vector

The Gaussian mixture model (GMM) $G_\lambda(x) = \sum_{k=1}^{K} \pi_k g_k(x)$ is adopted as the generative model for the Fisher Vector in which g_k denotes the kth Gaussian component:

$$g_k(x) = \frac{1}{2\pi^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_k)' \Sigma_k^{-1} (x-\mu_k)\right\}, \quad \forall k : \pi_k \geq 0, \ \sum_{k=1}^{K} \pi_k = 1, \tag{1}$$

where $x \in \mathbb{R}^D$ represents the feature descriptor; K is the number of Gaussian components; π_k, μ_k, and Σ_k stand for the mixture weight, mean vector, and covariance matrix, respectively. The covariance matrix Σ_k is assumed to be diagonal with the variance vector σ_k^2. We use the Expectation-Maximization (EM) algorithm to optimize Maximum Likelihood (ML) to estimate the GMM parameters $\lambda = \{\pi_k, \mu_k, \Sigma_k, \ k = 1, \dots, K\}$ based upon a large set of training descriptors.

Let $X = \{x_1, \dots, x_N\}$ denote a set of descriptors extracted from a sliding window. The soft assignment of descriptor x_i with respect to kth component is defined as:

$$w_i^k = \frac{\pi_k g_k(x_i)}{\sum_{j=1}^{K} \pi_j g_j(x_i)}. \tag{2}$$

Then the Fisher Vector of X is represented as $F(X) = \{\alpha_1, \beta_1, \dots, \alpha_K, \beta_K\}$, where α_k and β_k are D-dimensional gradients with respect to mean vector μ_k and standard deviation σ_k of component k:

$$\alpha_k = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} w_i^k \left(\frac{x_i - \mu_k}{\sigma_k}\right), \tag{3}$$

$$\beta_k = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} w_i^k \left[\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1\right]. \tag{4}$$

This system follows the scheme introduced in [10] to normalize Fisher Vector, i.e., firstly the power normalization and then L2 normalization. Please refer to [3] for the detailed description and parameter settings.

2.3.2. Spatial Pyramid

We spatially subdivide a video scene into a set of regions where low-level descriptors are pooled. To be more specific, the generic system adopts the three level

spatial pyramids [5] which are 1×1, 3×1, and 2×2 grids. For each grid, the Fisher Vector is computed and concatenated as the video representation.

2.4. Model Learning and Post Processing

In the generic framework, we adopt a 60-frame sliding window size that strides in every 15 frames. This sliding window scheme generates highly imbalanced data. As shown in Table 1, among all the evaluated events, even the most frequent event PeopleSplitUp only covers 4.37% of the entire video sequences.

Table 1: Proportions of video sequences containing positive events in the training set.

CellToEar	PersonRuns	ObjectPut	Embrace	Pointing	PeopleMeet	PeopleSplitUp
0.31%	0.60%	0.89%	1.51%	1.70%	3.58%	4.37%

The camera and event dependent models are learned to reduce intra-class variance and memory consumption in training phase. Therefore in our generic system, we train a group of Random Forests [6] for each of the six events under each camera view. In order to handle the imbalanced data and make full usage of the valuable positive data, we propose the following data segmentation scheme as illustrated in Fig. 2. For event i under camera view j, we denote the training set to be $D_{ij} = \{D_{ij}{}^{+}, D_{ij}{}^{-}\}$. We use notation $D = \{D^{+}, D^{-}\}$ in later context for simplicity. The negative data set is divided into a series of partitions $D_k^{-}, k = 1, \ldots, K$ with triple size of $|D^{+}|$. The whole training set is chopped into a group of data chunks where each data chuck is consisted of a portion of the negative samples and the whole positive set. Within each data chunk, a Random Forest is trained with 30 decision trees and the maximum depth for each tree is set to be 8.

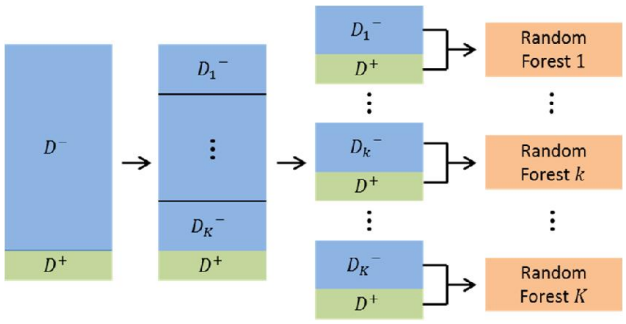


Figure 2: Illustration of data segmentation where within each data chunk a Random Forest is learned.

For a testing video, two or three low-level features are extracted from each sliding window. Each low-level feature generates a corresponding Fisher Vector. As shown in Fig. 3, each Fisher Vector is fed into a group of learned Random Forests and we perform the decision-level fusion after the classification. The decision-level fusion

combines outputs of multiple classifiers to make the final prediction. Minimum, maximum, median, majority voting, weighted sum, and geometric mean are all popular decision-level fusion methods [11]. Weighted sum is employed for late fusion in our generic system.

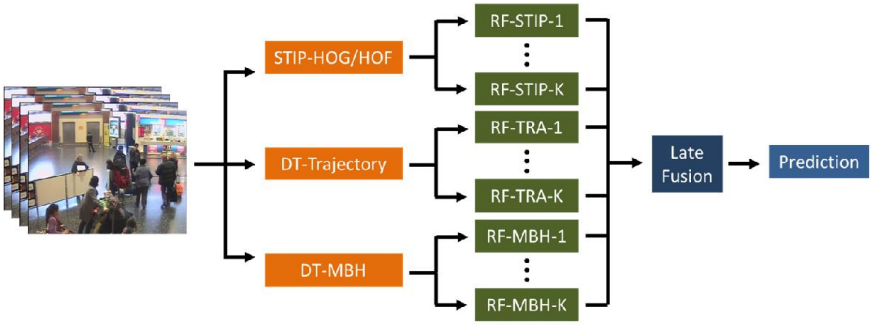


Figure 3: Illustration of late fusion in combining multiple low-level features.

An event might span several different windows due to the sliding window scheme adopted in our system. Therefore, after the classifier prediction, we employ a straightforward post processing to group continuous positive windows as to decide the final temporal interval of a detected event. In the post process, two positive predictions which have overlaps in their sliding windows can be merged together.

3. CellToEar Task Specific System

3.1. Motivation

Since the CellToEar task is commonly considered as the most difficult event among all the tasks as it has distinct properties and relatively less occurrences and more outliers, we design a different specific framework to solve this problem.

The conventional event detection methods often extract the low-level features in the temporal sliding windows first, and then design descriptors by encoding these features, finally determine the correct action with trained classifiers. The commonly used low-level features are generally categorized as global and local representations [12], both of which do not perform well enough on CellToEar task.

Recently numerous top-performance action recognition and event detection methods rely on the above framework and effectively solve large scale SED tasks, e.g., PersonRuns, Embrace and PeopleSplitUp. However, the performance is limited on small scale events including CellToEar and Pointing. For example, the Actual Detection Cost Rate (ADCR) of the best result for event PeopleSplitUp and the second best result for CellToEar task are 0.7781 and 0.9908 respectively in 2013, which means the latter one almost cannot produce any effective output in real world tasks.

Given the limitations of general event detection methods on small scale event detections, [12] attempts to introduce the mid-level discriminative representation to enhance the detection performance on CellToEar task, which obviously outperforms all other detection methods and achieve the best performance on CellToEar task (with ADCR = 0.9057) in 2013. Instead of relying on the video fragments containing the whole scene as the conventional methods, they train classifiers on mid-level discriminative patches and shots which are more intuitive to users compared with the abstract low-level features and better describe the patterns of interested events. The underlying patches which contain the target events are then sorted based on the classifier scores.

The methods of [12] reply more on the post-processing of human interaction. In our CellToEar task specific system, we focus on automatic detection and classification. We introduce more upper body properties and static features to enhance the performance.

(a) Original CellToEar event scenes.

(b) Initial detection bounding boxes including part models.

(c) Finalized detection bounding box.

Figure 4: Gesture detection for CellToEar task. (a) The original CellToEar event scenes. (b) The CellToEar specific Deformable Part Models (C-DPM) detection results (red bounding box) with part models (blue bounding boxes). (c) The final prediction bounding box determined by the initial detection results.

Figure 5: User interface of training data annotation.

3.2. CellToEar Gesture Detector

Here we identify the three properties of CellToEar event:

- generally lasts for a very short time, which can be well represented by several specific keyframes.
- usually occurs in a very local location with respect to the human body, which can be well described by part-based descriptors.
- often consists of a short-time arm waving and a long-time static calling gesture (even when the caller is moving).

Based upon these specific properties, we implement our CellToEar Gesture Detector based on the top-performance discriminatively trained deformed part-based models [13]. Part-based model has been successfully used in many object detection and recognition areas and achieves state-of-the-art results on the PASCAL VOC benchmarks [14] and INRIA Person dataset [15]. It represents highly variable objects using mixtures of multi-scale deformable part models. These models are trained using a discriminative procedure which only requires the annotation bounding boxes for the objects.

The deformable part models (DPM) are highly compatible with CellToEar task due to the following reasons: 1) DPM builds on a pictorial structures framework, which represents objects by a bunch of parts arranged in a deformable configuration. The

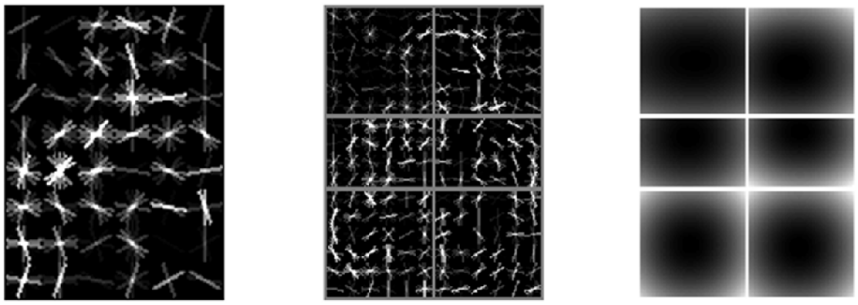
visual model provides an intuitive guidance for parameter tuning. 2) DPM can well handle the variations of human pose and appearance in the cluttered environment (as illustrated in Fig. 6.) since it relies on expressively enough mixtures models. 3) DPM can significantly reduce the difficulty of training process while boosting the efficiency as it introduces the part based latent (hidden, as the part locations have not been labeled) variables during training.

To make the DPM more effective in solving the specific CellToEar problems, we add more distinct features to distinguish the arm waving and calling gesture with other upper body gestures such as pointing, hand shaking and object put. The experiment results demonstrate the effectiveness of our method.

3.3. Details of Implementation

To obtain high performance using discriminative training methods, it is often crucial to use large training data sets. The whole TRECVID SED dataset contains approximately 100-hour videos as training data and 45-hour videos as evaluation data. We manually annotate all the bounding boxes for all the CellToEar events in the training videos based on the provided Ground Truth (time spanning). These labeled data are considered weakly labeled since the bounding boxes do not specify part locations or component labels. Fig. 5 indicates the graphical user interface which is used to label the training data.

After we learn all the parameters of target mixture models in DPM by constructing a latent SVM (LSVM), we carefully tune the parameters and number of part models to achieve the best performance since in CellToEar event the arm gesture is the unique property which shows the distinction with other gestures. Trained visual models are shown in Fig. 6. In the labeling process, we noticed that several unexpected Ground Truth video clips which actually do not contain any underlying CellToEar events at all. In this case we manually filter out those false Ground Truth clips. From the visual model we could intuitively figure out the general appearance of arm waving and phone handling gestures.



(a) Coarse Root filter (b) Higher resolution part filters (c) Spatial models

Figure 6: Trained visual DPM models for CellToEar event.

4. Experimental Results

All videos provided by TRECVID SED 2014 are captured by 5 fixed cameras with the frame resolution 720×576 at 25fps. In the generic system, the experiments reported in this paper are performed on an Intel Xeon computation server that comprises 24 cores (2.0GHz), 256GB memory, and 12TB hard disk. We downsample all videos to half of the original size for the low-level feature extraction process. After performing PCA to further reduce the feature dimension of STIP and DT-MBH by half, we train the GMM with 128 Gaussian components and adopt the 8-grid spatial pyramid in this system. Therefore, the dimensions of Fisher Vectors are 165888, 61440, and 196608 for STIP, DT-Trajectory, and DT-MBH, respectively.

In the CellToEar specific system, the experiments are performed in the same server. The training time for CellToEar specific model is about 10 hours on a 2.3 GHz 8-core and 48 GB ram Intel Xeon Computer. The detection framework works by traversing all the testing videos by performing detection every five frames. The detection process takes six to seven seconds each frame. After capturing all the detection scores, we firstly concatenate continuous underlying frames based on the time stamp (simultaneously average the scores), and then sort all the time spans based on the detection scores.

As shown in Table 2, we compare our systems to other participant systems in TRECVID 2014 in the primary metric Actual Detection Cost Rate (ADCR) and the secondary metric Minimum Detection Cost Rate (MDCR). The rank column denotes our rankings among all participants in terms of ADCR. Our system achieves the best performance in event PersonRuns and the second place in event PeopleSplitUp. The detailed performances of our system on all the events are shown in Fig. 7.

Table 2: Comparison of our system and other best systems in TRECVID SED 2014.

Event	Rank	ADCR of Other Best Systems	CCNY Primary Run				
			ADCR	MDCR	#CorDet	#FA	#Miss
CellToEar	3	0.9921	1.0257	1.0005	0	56	54
Embrace	4	0.8113	0.9611	0.9510	14	136	124
ObjectPut	3	0.9713	1.0177	1.0005	1	46	289
PeopleMeet	3	0.8587	0.9966	0.9901	11	86	245
PeopleSplitUp	2	0.8353	0.8698	0.8594	36	232	116
PersonRuns	1	0.8301	**0.8256**	0.8122	13	175	38
Pointing	4	0.9998	1.0547	1.0005	19	171	776

5. Conclusion

In this paper we have presented the detailed implementation of two SED systems participated in TRECVID 2014. The task specific system is applied to event CellToEar and the generic system is evaluated in all the rest six events. The generic system firstly extracts low-level features of STIP-HOG/HOF, DT-Trajectory, and DT-MBH from each sliding window. Fisher Vector is then employed to aggregate the low-level features. A group of Random Forests is utilized to learn the detection

models corresponding to the each specific event and camera view. We have further applied the decision-level fusions to capture the detection results. In CellToEar task, we have employed the top-performance deformable part models while integrating more specific features and prior knowledge to represent the distinct arm waving and phone handling gestures and achieved comparable performance. In the evaluations of 7 event detection tasks, our system achieves top 3 performances in 5 events.

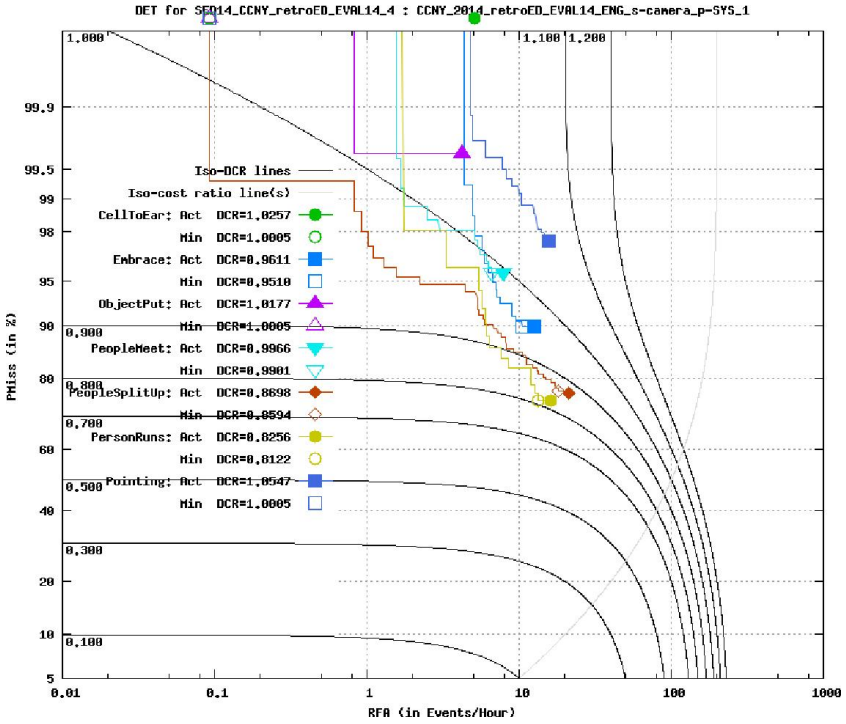


Figure 7: Detection Error Tradeoff (DET) curves of each event.

Acknowledgments. This work was supported in part by NSF Grant IIS-1400802.

References

1. Y. Ke, R. Sukthankar, and M. Herbert. Event Detection in Crowded Videos. *International Conference on Computer Vision*, 2007.
2. A. Smeaton, P. Over, and W. Kraaij. Evaluation Campaigns and TRECVID. *ACM Workshop on Multimedia Information Retrieval*, 2006.
3. X. Yang, Z. Liu, E. Zavesky, D. Gibbon, B. Shahraray, and Y. Tian. AT&T Research at TRECVID 2013: Surveillance Event Detection. *NIST TRECVID Workshop*, 2013

4. 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.
5. 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.
6. L. Breiman. Random Forests. *Machine Learning* 45(1): 5-32, 2001.
7. I. Laptev. On Space-Time Interest Points. *International Journal of Computer Vision*, 2005.
8. I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
9. H. Wang, A. Klaser, C. Schmid, and C. Liu. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *International Journal of Computer Vision*, 2013.
10. J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, 2013.
11. P. Atrey, M. Hossain, A. Saddik, and M. Kankanhalli. Multimodal Fusion for Multimedia Analysis: A Survey. *Multimedia System*, 2010.
12. C. Gao, D. Meng, W. Tong, Y. Yang, Y. Cai, H. Shen, G. Liu, S. Xu and A. Hauptmann. Interactive Surveillance Event Detection through Mid-level Discriminative Representation. *ACM International Conference on Multimedia Retrieval*, 2014.
13. P.F. Felzenszwalb, R.B. Girshick, D. McAllester and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2010.
14. M. Everingham, L.V. Gool, C. Williams, J. Winn and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 2010.
15. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
16. X. Yang, C. Yi, L. Cao, and Y. Tian. MediaCCNY at TRECVID 2012: Surveillance Event Detection. *NIST TRECVID Workshop*, 2012