SAVASA Project @ TRECVID 2012: Interactive Surveillance Event Detection

Suzanne Little*[1], Iveel Jargalsaikhan[1], Cem Direkoglu[1],
Noel E. O'Connor[1], Alan F. Smeaton[1],
Kathy Clawson[2], Hao Li[2], Marcos Nieto[3],
Aitor Rodriguez[4], Pedro Sanchez[4],
Karina Villarroel Paniza[5], Ana Martínez Llorens[5],
Roberto Giménez[6], Raúl Santos de la Cámara[6], Anna Mereu[6]
1. CLARITY Centre for Sensor Web Technology,
Dublin City University, Ireland
2. University of Ulster, United Kingdom
3. Vicomtech-IK4, Spain
4. IKUSI, Spain
5. RENFE, Spain
6. Hi-Iberia, Spain

Abstract

In this paper we describe our participation in the interactive surveillance event detection task at TRECVid 2012. The system we developed was comprised of individual classifiers brought together behind a simple video search interface that enabled users to select relevant segments based on down sampled animated gifs. Two types of user – 'experts' and 'end users' – performed the evaluations. Due to time constraints we focussed on three events – ObjectPut, PersonRuns and Pointing – and two of the five available cameras (1 and 3). Results from the interactive runs as well as discussion of the performance of the underlying retrospective classifiers are presented.

1 Introduction

The DCU-SAVASA team consisted of researchers and developers from three different institutions – Dublin City University (Ireland), University of Ulster (United Kingdom) and Vicomtech-IK4 (Spain) – and three partners who provided end users for the interactive component – IKUSI, RENFE and Hi-Iberia (all from Spain). The team's submission to the interactive surveillance event detection task of TRECVid 2012 [10, 12] is an example of how individual, disparate tools for video analysis can be brought together and applied to identify specific events in surveillance video. As first-time TRECVid participants, brand-new partners from multiple, multinational institutions we needed to combine the novel element of user interactivity with classical computer vision classifiers. Our work is motivated by the goals of the EU FP7 SAVASA project[1] (Standards-based Approach to Video Archive Search and Analysis), where we contribute to the semantic annotation of CCTV footage, and we were fortunate to have end-users as project partners who were willing to participate in the user-based evaluations.

*Contact author: suzanne.little@dcu.ie
[1]http://www.savasa.eu

The SAVASA project aims to develop a standards-based video archive search platform that allows authorised users to query over various remote and non-interoperable video archives of CCTV footage from geographically diverse locations. At the core of the search interface is the application of algorithms for person/object detection and tracking, activity detection and scenario recognition. The project also includes research into interoperable standards for surveillance video, discussion of the legal, ethical and privacy issues and how to effectively leverage cloud computing infrastructures in these applications. Project partners come from a number of different European countries and include technical and research institutions as well as end user, security and legal partners.

Our experiments consisted of two interactive and eight retrospective (machine-based) runs. The interactive runs combined results from two different classes of users – 'experts' who were technical people with an understanding of computing or computer vision and 'end users' who worked in organisations performing video surveillance tasks – using both raw output and the annotations from trained classifiers presented in a search interface. The classifiers used techniques including conventional person tracking, motion trajectory analysis and frame-based activity. In the interest of time and to focus the work we only looked at cameras 1 and 3 and the events ObjectPut, PersonRuns and Pointing.

2 User Interactive Search

The approach we chose for the new interactive surveillance event detection task was to combine individual approaches to video analysis and annotation and provide a dashboard style search interface that enabled the user to view results for various algorithms and filter them by factors such as confidence, level of motion, camera, number of people etc. The use of a search interface changes the problem from a pure annotation task, where the objective for each individual classifier is to maximise precision and recall for 1 or more events over all videos, to an information retrieval task, where fuzzy, difficult to predict factors such as user understanding and patience, interface asthetics, minimising false alarms and optimising for high precision in the first results, have the most influence on the success of the system.

To reduce the focus on the interface and interaction model, interface components from the AXES video archive search interface[2] were reused including the display of thumbnails in a grid and the drag-n-drop method to save video segments that matched the query target. Rather than a static thumbail we choose to generate looping, animated gifs to display a summary of the video segment. Figure 1 shows a screenshot of the interface.

Our expectation was that the inclusion of a user in the task would greatly reduce the number of false alarms in the results submitted for evaluation as a human would provide the best authority on the relevance of a segment. The total number of found results would also be much smaller as rather than the user having all 15 hours of video to choose from only the video segments that had been annotated by one or more classifiers would be available for selection as a match.

2.1 Process

1. User introducted to TRECVid and SAVASA project aims
2. Demonstration of search interface using the EVAL08 portion of the training dataset (so results were all 'correct'). Discussion with users about the different types of ObjectPut, PersonRuns, Pointing events and how difficult some could be to detect.
3. User 'trains' on the interface using the EVAL08 dataset.
4. User is instructed to 'be generous' and save any segment that they think might be showing the event they are currently searching for. User is also told that the time is a 'limit' and not an instruction to spend the full amount if they feel they are finished sooner.
5. User searches for 'PersonRuns', 'Pointing', 'ObjectPut' events with the search interface.
6. Results lists from each of the 'expert' or 'end user' users were merged by a simple vote and detection scores normalised to produce the final submission.

[2] http://wwww.axes-project.eu

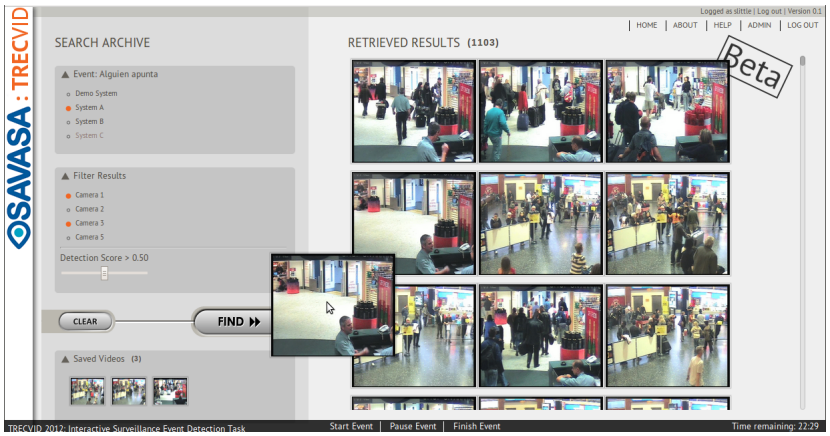

Figure 1: User interface screenshot

Table 1: summary of results for interactive runs

run, event	#Targ	#Sys	#CorDet	RFA	PMiss	DCR	minDCR
end users, ObjectPut	621	48	3	2.95136	0.995	1.0099	1.0003
experts, ObjectPut	621	64	3	4.00073	0.994	1.0152	1.0003
end users, PersonRuns	107	10	2	0.52469	0.981	0.9839	0.9836
experts, PersonRuns	107	14	2	0.78703	0.981	0.9852	0.9843
end users, Pointing	1063	25	12	0.85261	0.989	0.9930	0.9926
experts, Pointing	1063	100	25	4.91893	0.976	1.0011	0.9995

Table 2: Average search time (seconds) per event for each class of user

	'expert'	'end user'	all
ObjectPut	1109	238	673
PersonRuns	187	95	141
Pointing	1020	184	602

2.2 Results

A summary of results for interactive runs can be found in Table 1. It can be seen that the Detection Cost Rate (DCR) ranges from 0.9839 (end users, Person Runs) to 1.0152 (experts, object put). Across all experiments, mean DCR is 0.99805. Our evaluation considers only a subset of the complete evaluation data — the estimated maximum number of correct event classifications achievable in our case (for cameras 1 and 3) was between 39–50 percent of the total, based on event occurrence statistics in the training dataset. Therefore the true PMiss values are slightly smaller while the RFA will be slightly larger and the overall effect on the DCR is negligible.

Table 2 illustrates the average search time in seconds per event for each class of user. It can be seen that our 'expert' users' search times were significantly longer than 'end users'. Across all experiments, users spent the least time searching for PersonRun. Correspondingly, this was the event group for which our classifiers returned least annotations.

Table 3: sumary of results for retrospective runs

run, event	#Targ	#Sys	#CorDet	RFA	PMiss	DCR	minDCR
dcu1, ObjectPut	621	9	0	0.59027	1.000	1.0030	1.0003
dcu1, PersonRuns	107	20	3	1.11496	0.972	0.9775	0.9769
dcu1, Pointing	1063	136	16	7.87029	0.985	1.0243	1.0003
uu1a, ObjectPut	621	457	11	29.25123	0.982	1.1285	1.0003
uu1a, Pointing	1063	981	50	61.06030	0.953	1.2583	1.0003
uu1b, ObjectPut	621	308	3	20.00364	0.995	1.0952	1.0003
uu1b, Pointing	1063	950	57	58.56805	0.946	1.2392	1.0003
uu1c, ObjectPut	621	730	24	46.30352	0.961	1.1929	1.0010
uu1c, Pointing	1063	2174	96	136.28712	0.910	1.5911	1.0007
uu1d, ObjectPut	621	876	22	56.0102	0.965	1.2246	1.0007
uu1d, Pointing	1063	1286	56	80.67043	0.947	1.3507	1.0013
uu2, PersonRuns	107	39	2	2.42667	0.981	0.9934	0.9820

2.3 Discussion

The most surprising part of the interactive results was the high number of false alarms. Our prediction was that having a user act as a result filter would greatly reduce the false alarms. One possibility is that the high RFA is due to the sensitivity of the event alignment method used in the evaluation. It is likely that the event segments found by the classifiers have a shorter duration or inexact overlap with the groundtruth event segments. This would be interesting to explore further, as for surveillance video retrieval tasks it is less critical to find the exact boundaries of the event but rather just to identify that an event has occured. Particularly if the task is interactive.

The users from our evaluations were divided into two classes. We expected that the ‘experts‘ would produce better results due to a better understanding of the task and spending more time searching for relevant segments. Contrary to our expectations the ‘end users’ results were slightly better. This is most likely due to the smaller number of correct results in the segments found by the classifiers and available to the users while searching. The ‘experts‘ were also more likely to find more segments and hence have a higher false alarm rate.

The most beneficial outcome of our user evaluations was the opportunity to meet with end users in the surveillance field and discuss their requirements for event retrieval. Many of the users we met with were unaware of TRECVid and information retrieval research in general. The particular activities annotated in the dataset were not of interest to our users who were often confused as to why we had annotated events such as Pointing.

3 Event Detection

This section outlines the classifiers used to detect segments that were displayed to the user through the search interface. Six retrospective runs were also submitted based on the raw output from the classifiers.

As part of our work we began a shared, manual annotation effort to identify the region of interest in frames where events had been annotated in the training video set, similar to that mentioned by the IRDS-CASIA team in the SED task 2011 [11]. While time and resources prevented extensive coverage, we were able to generate a few matrices showing the probability of a pixel belonging to an event region. These were converted into ‘heat maps’ showing the areas in different cameras where specific events tended to take place (Figure 2). This probability was applied in two of the runs described in section 3.2.5 to adjust the detection score of classifiers.

3.1 Action Recogntion Using Motion Trajectory (DCU1)

3.1.1 Feature Extraction

To represent motion, we have used salience point trajectory as a low-level feature and further described it using four different descriptors. First, in order to extract motion trajectory, we applied a background subtraction algorithm [7] to detect foreground regions. This processing helps to reduce computationally complexity and increases the accuracy of point tracking by reducing the regions of search area. Then salience points are located within the foreground regions by Harris Corner Detector and are tracked over video sequences using Kanade-Lucas-Tomasi (KLT) algorithm. In the experiments, we have observed that longer salience points' trajectories are likely to be erroneous. Thus we empirically set the maximum trajectory length to be 15 frames.

3.1.2 Motion Trajectory Description

We adopted Heng *et al.*'s [13] approach to describe the trajectory features. For each trajectory, we calculated four descriptors to capture the different aspects of motion trajectory. Among the existing descriptors, HOGHOF [8] has shown to give excellent results on a variety of datasets [14]. Therefore we computed HOGHOF along our trajectories. HOG (histograms of oriented gradient)[2] captures the local appearance around the trajectories whereas HOF (histograms of optical flow) captures the local motion. Additionally, MBH (motion boundary histogram) which is proposed by Dalal *et al.* [3] and TD (trajectory descriptor) [13] are computed in order to represent the relative motion and trajectory shape.

In order to represent the video scene, we have built the Bag-of-Feature (BoF) model based on our four descriptors. This requires the construction of a visual vocabulary. In our experiments, we cluster a subset of 250k descriptors sampled from the training video with the k-means algorithm for each descriptor. The number of clusters is set to $k = 4000$, which has shown empirically to give good results in [8]. The BoF representation then assigns each descriptor to the closest vocabulary word in Euclidean distance and computes the co-occurrence histogram over the sub-sequence video.

3.1.3 Classification

For classification, we use a non-linear support vector machine (SVM) with RBF kernel. Using the cross-validation technique, we have empirically found the parameters of cost (32) and gamma (1e-05) of the kernel. In order to represent the video frame, we have utilized the temporal sliding window approach. In the experiments, we set the window size to 25 frames and sliding step size to 8 frames.

3.1.4 Discussion

Here we have implemented action detection algorithm which is based on sparse motion trajectory. Since trajectory is suitable for representing gesture like movement, we mainly focused on building a classifier for Pointing events using this method. Although we have not assigned any spatial association between the extracted trajectories' descriptors, this approach performed reasonably well on the challenging TRECVid SED dataset. In the future, we would like to explore an alternative way to represent the video scene rather than Bag-of-Feature (BoF) approach which ignores the spatial information.

3.2 Comparing Frame-based Methodologies - (UU1)

We compare two frame-based methodologies for event recognition, namely

- Hidden Markov Model classification using Optical Flow features
- SVM classification using dense SIFT features with Bag of Words

After classification, we threshold to retain only the top *n* events (ranked by confidence), and link these events temporally to derive start and end times. Final confidence score is the mean confidence across the

linked time period. We set *n* purposefully high to allow high false positive rates, in the hope that this will allow a higher proportion of true positives to be captured.

3.2.1 HMM Model with Optical Flow Features

The following features are generated and applied to the action recognition task:

Histograms of Oriented Optical Flow For each individual frame we compute a normalised histogram of oriented optical flow magnitude (90 bins, spaced equally apart). The magnitude of each bin corresponds to the sum of magnitude of optical flow.

2D Zernike Moments of Efros Descriptor images For each frame, calculate the optical flow vector field F. Split F into two scalar fields, Fx and Fy, corresponding to the horizontal and vertical components of the flow. Then half-wave rectify Fx and Fy into 4 non-negative channels: Fx+ Fx- Fy+ and Fy-. Finally, blur with a Gaussian to remove spurious motions. These channels, known as Efros descriptors [6] may be regarded as distinct images. As features, we calculate 2D Zernike moments of each of the new channels (16 features per channel, per frame), resulting in a 64*1 feature vector per frame.

These features are concatenated into a single vector (154*1) and the feature space is reduced to 16*1 using principal components decomposition.

3.2.2 SVM Model with Dense SIFT features

During training, we calculate dense SIFT features [1] around spatial interest points and cluster these features to create a visual bag of words. For classification model learning and test set evaluation, histograms of visual words are used as input features to an SVM with radial basis function.

3.2.3 Training Paradigm

To generate classification models, we train our HMM and SVM using ground truth events from the TRECVid training dataset. Specifically, we utilised an in-house annotation tool to manually identify within each frame the region of interest containing an event. Each ROI is regarded as a unique event instance.

3.2.4 Test Paradigm

In order to apply frame-based classification to TRECVid test data, we adopt a grid based approach. Specifically, each frame is divided into 36 equally-sized regions of interest and each region is evaluated separately.

3.2.5 Additional Evaluation: Integration of Prior Probability into Final Confidence Measure

Using prior knowledge of event occurrence from manually annotated training data, we can update our confidence scores in a probabilistic manner. Specifically, we perform the following tasks:

- Using ground truth training data generate a heat map (sum of event occurrence per pixel across all training frames) which shows where each event tends to occur within each camera scene (Figure 2).
- Represent each heat map as a pixel-level probability distribution whose sum is 1.
- Generate a quantized probability distribution map- Represent the probability of event occurrence within a single grid region as the sum of probabilities within that grid (Figure 2).
- The result of this is a 36*1 vector (or 6*6 matrix) whose sum is 1. These are our update weights. To get the final classification score, we update each frame's event likelihood using the corresponding weight.

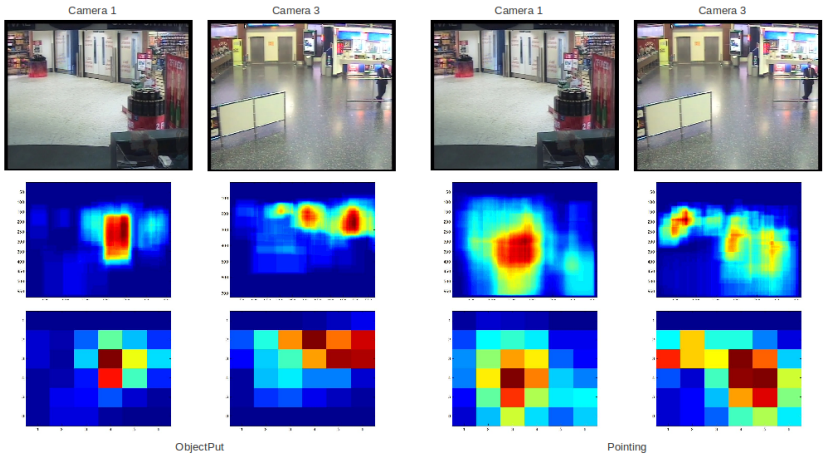

Figure 2: Heat map showing pixel frequency per event per camera

3.2.6 Discussion

The purpose of experiments described in Section 3.2 was to evaluate and compare two frame-based supervised-learning techniques for event classification (HMM with optical flow features and SVM with BoW), considering whether or not a priori information could increase accuracy and reliability of event classification. Across all experiments, the most successful classifier was the SVM incorporating a priori information. DCR was 1.0952 and 1.2392 for Object Put and Pointing events, respectively. For both events, min DCR was 1.0003.

One factor worth mentioning is the high RFA achieved across all experiments. During temporal linking and thresholding, our threshold was set intentionally to overestimate event occurrence. Future objectives include the reduction of RFA, by modification of threshold values and generation of enhanced / improved event models for classification. It is envisaged that we calculate non motion-based descriptors for event classification, and combine these with our existing optical low feature sets.

3.3 Person Runs Detection using Optical Flow (UU2)

For detection of person runs events, we propose an algorithm based on background subtraction and grid-based motion summarization. Specifically, we perform the following:

1. Background subtraction via mproved Gaussian mixture model for motion detection [9]
2. Grid based (18*18 pixels) evaluation of optical flow.

After background subtraction, we calculate the magnitude of optical flow from 25 fixed sampling points within each grid and then obtain the median magnitude as a statistical measure of action within that region. To reduce noise, we apply a temporal sliding window (size 5) within each grid. The statistical action / motion feature is collected over all training data, and the model is trained per camera. Subsequently, we apply a histogram based automatic thresholding algorithm to obtain a threshold for the magnitude of optical flow in each grid.

To classify person runs events in the test data, we apply the above algorithm and identify potential regions of interest as any significant large region (e.g., more than 5 connected grids) with magnitude of optical flow exceeding the corresponding threshold. To reduce false detection, we consider temporal consistency. The grids occupied by fast moving objects should have neighbours which also are also characterised as fast motion regions.

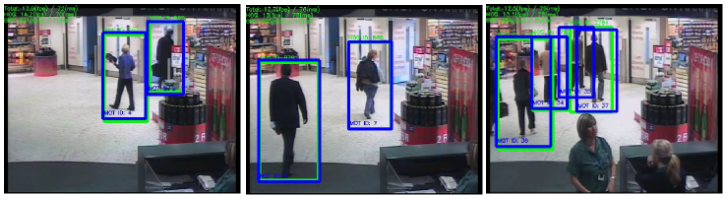

Figure 3: Examples of person detection and tracking: (green) HOG detections, (blue) tracked detections

3.4 Person Detection and Tracking (Vicomtech)

For the detection of persons, we have used HOG descriptors [2] and a pre-trained person detector which yields a "sparse" set of detections in time, i.e. there are a lot of miss-detections. False negatives can be solved using tracking approaches, which are anyway needed to provide time coherence to detections, so that we can reconstruct the trajectory of objects.

For the tracking, we have implemented a Rao-Blackwellized Data Association Particle Filter (RBDAPF) [5]. This type of filter has been proven to provide good multiple object tracking results for even in the presence of "sparse" detections as the ones we have in these sequences, and can be tuned to handle occlusions. The Rao-Blackwellization can be understood as splitting the problem into linear/Gaussian and non-linear/non-Gaussian parts. The linear part can be solved with Kalman Filters, while the non-linear one must be solved with approximation methods like particle filters. I our case, the linear part is the position and size of a bounding box that models the persons. The non-linear part refers to the data association which is the process of generating a matrix that links detections (the HOG ones, for instance), with objects or clutter. The association process can be strongly non-linear so that sampling approaches can be used. In our case we have implemented ancestral sampling [4].

Preliminary results have shown that this approach is able to detect and track persons whose full body is clearly seen in the scene, up to 4 or 5 simultaneous persons. When more than 5 persons, we have found that in these types of images, multiple occlusions happen and the full-body detector does not provide good detection results.

Besides, the control of input/output of new persons is handled thanks to the use of the data association filter, which classifies detections according to the existing objects, remove objects that have no detection for a too long period of time, and creates new objects when detections not associated to previous objects appear repeatedly.

The most time consuming part of the algorithm is the detection stage, which takes about 70 ms in a Core2 Quad @ 2.5 Ghz, using GPU capabilities. The reason is that we have focused on the tracking stage, and thus we have not optimized the use of the HOG-based detector, which scans the whole image at different scales in order to detect persons at different distances. The RBDAPF method consumes only 4 ms, since its main effort is in the generation of the data association matrix and template matching steps.

4 Conclusions

The aim of our participation in the interactive surveillance event detection task was to bring together the different computer vision techniques from our different groups, to explore options for identifying events in surveillance footage and to gather information from our end user partners about their needs. The process of exchanging video processing and feature extraction suggestions and sharing ideas for improving the classification was very valuable.

The performance of the interactive systems was surprising but the user evaluations fulfilled their purpose of starting discussions with our end user partners. The performance of the individual classifiers show some promise and we have a number of ideas for advancing and improving the work we have begun here.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 285621, project titled SAVASA. With thanks to our project partners who assisted in hosting and conducting user evaluations.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, pages 401–408. ACM, 2007.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893. IEEE, 2005.

[3] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision (ECCV)*, pages 428–441. Springer, 2006.

[4] C. R. del Blanco, F. Jaureguizar, and N. Garcia. An advanced Bayesian model for the visual tracking of multiple interacting objects. *EURASIP Journal on Advances in Signal Processing*, 130, 2011.

[5] A. Doucet, N.J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2001.

[6] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 726–733, 2003.

[7] P. Kelly, C.O. Conaire, C. Kim, and N.E. O'Connor. Automatic camera selection for activity monitoring in a multi-camera system for tennis. In *Third ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–8, 2009.

[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[9] H. Li, A. Achim, and D.R. Bull. GMM-based efficient foreground detection with adaptive region update. In *16th IEEE International Conference on Image Processing (ICIP)*, pages 3181–3184, 2009.

[10] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Barbara Shaw, Wessel Kraaij, Alan F. Smeaton, and Georges Quéenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[11] Y. Shan, Z. Zhang, S. Wang, K. Huang, and T. Tan. IRDS-CASIA at TRECVid 2011: Surveillance Event Detection. In *TRECVID Benchmarking Activity*, 2011.

[12] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[13] H. Wang, A. Klaser, C. Schmid, and C.L. Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176, 2011.

[14] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, and C. et al. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, 2009.