CCNY-SRI @ TRECVid 2013 intED: a Human Interactive Event Detection System

Chenyang Zhang[1], Xiaodong Yang[1], Chucai Yi[2], Yingli Tian[1], Qian Yu[3], Amir Tamrakar[3], and Ajay Divakaran[3]

[1]Department of Electrical Engineering, CUNY City College
[2]Department of Computer Science, CUNY Graduate Center
[3]SRI International

Abstract

In this paper, we present a interactive Event Detection (interactiveED) system evaluated in TRECVID 2013 competition. The proposed system is evaluated on all the seven event categories. Two channels of raw features: STIP-HOG/HOF and SURF/MHI-HOG – are extracted from down-sampled surveillance video frames. A Bag Of Words model is used to generate representations of segments of surveillance videos. Segments of videos are sampled using sliding windows. Spatial priors are utilized in two channels: Hot Region and Human Tracking Masks. Spatial priors are used for filtering noisy feature responses. Feature mapping technique is employed to generate more features, which potentially increase the separability of sample set in feature space. A cascade of SVMs are learned to tackle the classification task, with respect to different camera views. Human interaction is utilized as a post-processing layer to filter out false alarms. A Graphical User Interface (GUI) is developed to help an expert user to accomplish the task efficiently and conveniently.

1 Introduction

Detection is a primary and fundamental task in computer vision. Object detection is to search a given type of object in a spatial image space. Similar to object detection, event detection is to search a given type of activity or behavior of human in spatio-temporal video space. With the dimensional increase (from 2D image space to 3D video space) in search space, the difficulty of detection task is increased exponentially. However, event detection has remained a very applicable topic despite difficulty for decades, especially in surveillance systems, traffic tracking systems, and sports programs. Many fundamental visual descriptors and representation means are also evolved, such as Space Time Interest Point [4], Motion History Image [1], *etc*.

Considering the difference and gap between pure research and needs in real life application, such as surveillance, TRECVid [] organized Interactive surveillance event detection (iSED) task to provide a platform to apply a variety of computer vision technologies. The objective of iSED is to design a human assisting system supporting the optimal division of labor between a human user and the interactive system. The corpus TRECVid provides is from London Gatwick International Airport, which is 144-hour-long and under 5 different camera views.

Our team, CCNY-SRI, have participated the retrospective event detection task (rSED) last year as "MediaCCNY" for the first time. And our system of last year is evaluated in all 7 events and achieves top 3 in 5 events in terms of DCR scores.

In this year, we participate in the interactive Surveillance Event Detection task and finish all seven events: *PersonRuns*, *CellToEar*, *ObjectPut*, *PeopleMeet*, *PeopleSplitUp*, *Embrace*, and *Pointing*. Our framework is improved from our system from year 2012 [7].

2 System Framework

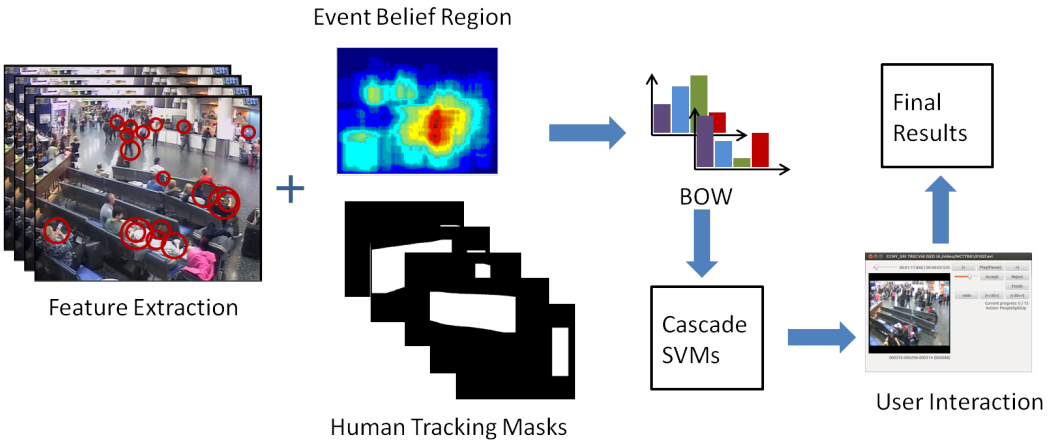

Figure 1: Framework

As shown in 1, our system has five steps: (1) Feature extraction, (2) Feature filtering based on event belief region and human tracking masks, (3) Bag-of-Words representation based on K-means Clustering, (4) Cascade Maximal Margin Classifiers Learning, (5) User interaction based on an GUI as post-processing.

In our system, we use the same channels of detectors and descriptors as we used in last year's TRECVID rSED task: STIP-HOG/HOF and SURF/MHI-HOG. However, since we observed that the original frame resolution may be a bottleneck of computation, therefore we down-sample the resolution into its one fourth. To further eliminate the number of features, we apply two types of filtering techniques, one is Event Belief Region, which is based on event-wise location confidence statistics; and Pedestrian Tracking masks. The features are then quantized to a certain vector length using K-means clustering and soft assignment to build a Bag-of-Words (BOW) representation of each sliding window.

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 [5] to code low-level features. The local soft assignment coding is able to achieve comparable performance but with acceptable computational cost. After feature coding, we choose the max pooling to aggregate coded features. Before learning event models, we first apply the explicit feature maps [6] 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 automatic 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.

An expert user is then involved with the aid of a Graphical User Interface to filter out more False Alarm detections (FA). The UI is implemented with a cross-platform toolbox name PyQt4. The user can easily use the interface to view the videos segment-by-segment and reject FAs instantly. The experimental results demonstrated that within 25 minutes, the user can remove a significant amount of FAs.

3 Feature Extraction and Representation

In this section, we introduce the features we extracted and the techniques and priors we used for feature filtering. The original video frame size is 720×576 and the frame rate is 25 frame per second, which contains over 10 million pixels per second. To reduce the number of features extracted and to accelerate computation, we down-sample each frame in scale of 2, which makes the resolution to 360×288 before we extract features.

3.1 Feature Channels

The two feature channels we utilized this year is the same as we did in last years rSED: STIP-HOG/HOF and SURF/MHI-HOG. The name of each feature channel follows the format as "detector-name(s)"-"descriptor-name(s)". Space-Time Interest Points (STIP) [4] is a spatio-temporal interest point detection built in Harris and Förstner interest point operators. We extract Histogram of Gradient (HOG) [3] and Histogram of Optical Flow (HOF) [2] descriptors from image regions around detected interest points. The other channel is built upon SURF and MHI, which combination is used as interest point detector. In this feature channel, HOG are used as descriptors around the regions detected by SURF. We believe that the two channels are complementary so that the combination can be more informative and robust to noise. Some examples of our feature extraction is as shown in Figure 2.

Figure 2: Interest points detected by different channels.

3.2 Feature Filters

The original number of features are numerous and it is also not necessary and practical to use all of them. In this section, we introduce two ways to fulfill feature filtering in our system. We first introduce an online human tracking technique using robust human appearance modeling and cascade particle filter. Human tracking results provided a local and precise location prior while the other filtering way, event belief region, provides a location prior based on the whole statistics with respect to a certain event under a given camera view.

3.2.1 Appearance based Pedestrian Tracking Mask

Human tracking provides useful information about the exact coordinates where event may occur. In our system, we deploy a descriptive appearance model for robust human tracking, which utilizes

encoded color histogram for human appearance representation [9]. The body of human can be separated into a set of rectangles as shown in Figure 3, which enables a part-based spatially encoded color histogram. We also maintain an online updated appearance model [8] to adapt to the appearance changes during tracking. Human tracking also needs to maintain the identities and states of objects at each time instant. However, missing detections, false alarms and erroneous detections are inevitable. Thus, the "detect and associate" types of approaches cannot make full use if the spatio-temporal contextual cues available in the video data. Our tracking system does not rely solely on the human detections after track initialization. In order to achieve real-time performance, we also adopt the cascade particle filter [10]. A cascade particle filter spreads the evaluation of likelihood functions into difference stages. It is also shown that cascade particle filter can achieve 40 $\times$ speed up over the conventional particle filter thus making real-time tracking feasible.

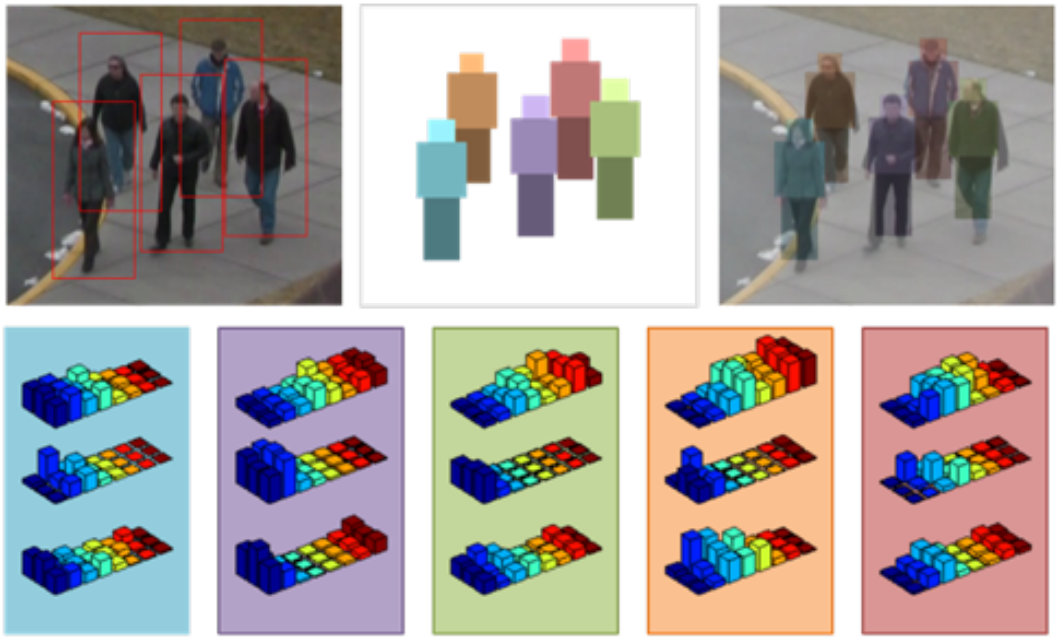

Figure 3: (Top Left) Human detectins. (Top Middle) Appearance model masks. (Top Right) Masks applied to the image to compute color histograms. (Bottom) The appearance features for the five persons detected.

3.2.2 Event Belief Region

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. 4. The bounding boxes of people performing actions are annotated to construct these Event Belief Region.

4 Cascade Maximal Margin Classifier Learning

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.

Suppose we have a training for each event under each camera view. The cascade SVMs algorithm adaptively divides the negative set into a series of partitions with the same size as costive set ac-

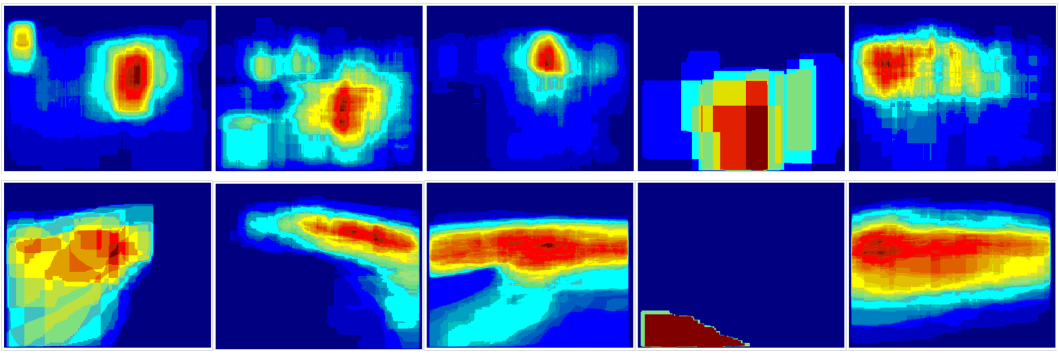

Figure 4: Examples of hot regions of event ObjectPut (top) and event PersonRuns (bottom) corresponding to camera views of 1-5 (from left to right).

cording to the ranked prediction scores and iteratively learns a group of binary SVMs classifiers that favors to positive samples. These classifiers are cascaded as the event model. For more details, please refer to our last year's report [7].

5 Human Interaction as Post-processing and Design of GUI

Figure 5: Our Graphical User Interface and functional panels implemented by PyQt4, a cross-paltform framework. (a) Video playing and information panel. (b) Control and event information panel.

As illustrated in Figure 5, our GUI provides the user a basic video watching and control platform. The video playing panel as shown in blue dashed box shows the content of the video and the progress of the whole video (top sliding bar) and local progress of current detection clip (bottom numbers). The control panel on the right, bounded by red dashed box provides the user several useful operations

such as accept or reject the current clip, redo last decision, left or right shifting the clip by a fixed number of frames and information about current task: what is the event detected and how many are left. Noted that the sliding bar on the upper left in the control panel gives the user a instant progress control function where user can use the sliding bar to view any contend inside the current clip, which is very useful when the event occurring in the current clip is so ambiguous that the user need to repeatedly and quickly watch the clip to confirm.

6 Results

The results are still pending by the time the notebook paper is written. We showed an example from our development experiments instead. The results discussion is based on the evaluation on

Figure 6: Visualization of our detection results before and after user interaction. Ground truth and correct detections are labeled as green arrows. False Alarms are shown in red and blue arrows show missed detections. Ground truth are illustrated on the top, retrospective result are shown in the middle and result after user interaction is shown in the bottom. The figure is based on result of "Embrace" on video LGW_20071130_E1_CAM1. X-axis shows the frame index.

the data of LGW_20071130_E1 of Event "Embrace". As can be seen in Figure 6, false alarms are successfully discarded by a human user.

A more quantities comparison is as shown in Figure 7. In fact, the retrospective detection has given 193 event alarms, where 21 of them are correct and 172 of them are false alarms. Base on the result, an user interaction session has decreased the number of alarms to 21 (89% less) and decrease the number of false alarms to 7 (96%) with sacrificing 7 correct detections (33%).

7 Conclusion

In this report, we have presented a detailed description of the CCNY-SRI system in TRECVID 2013 interactive SED task. Our system starts from extracting raw features of consecutive sampled sliding windows. The local soft assignment coding and max pooling are used to generate compact representations of the sliding windows. To avoid using computational expensive kernel tricks with the SVMs, we apply explicit feature mappings to make the training data more linearly separable. Despite the automatic post-processing based on our observation, we develop a easy-to-use Graphical

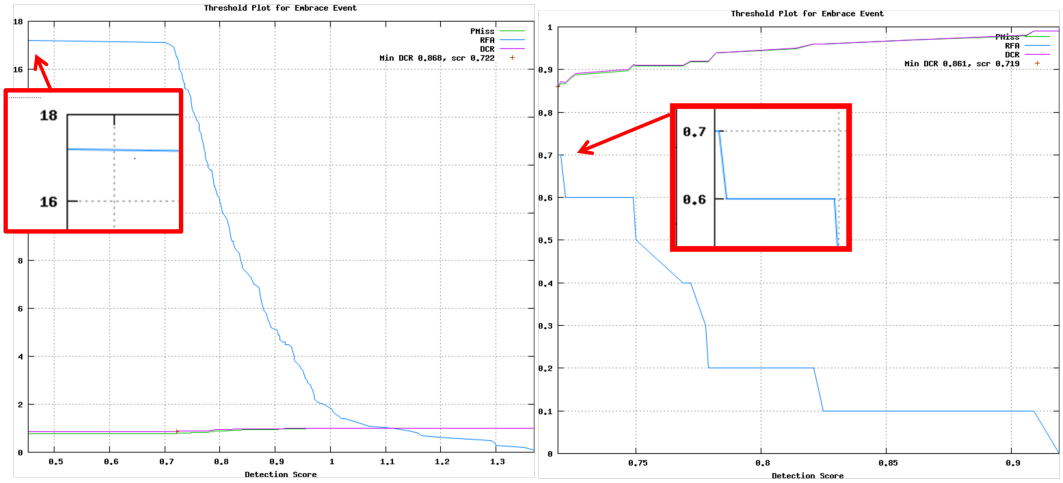

Figure 7: Threshold Plot Result of Embrace Event. Left: retrospective detection. Right: After user interaction. Noted that the enlarged figure, the Ratio of False Alarm (RFA, cyan curve) are decreased a lot.

User Interface to enable an expert user to instantly watch the video and make decisions. Although the final results are still pending by the time we write this report, we have shown that the user interaction has reduce the number of false alarms by a large ratio.

References

[1] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23(3):257–267, 2001.

[2] Rizwan Chaudry, Gregory Hager Avinash Ravichandran, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2009.

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2005.

[4] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision (IJCV)*, 64(2/3):107–123, 2005.

[5] L. Liu, L. Wnag, and X. Liu. In defense of soft-assignment coding. In *ICCV*, 2011.

[6] A. Vedaldi and A. Zizzerman. Multiple-target tracking by spatiotemporal monte carlo markov chain data association. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2012.

[7] Xiaodong Yang, Chucai Yi, Yingli Tian, and Liangliang Cao. Mediaccny at trecvid 2012: Surveillance event detection. Technical report, NIST TRECVid SED 2012.

[8] Qian Yu, Thang Dinh, and Gérard Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *ECCV*, 2008.

[9] Qian Yu and Gérard Medioni. Multiple-target tracking by spatiotemporal monte carlo markov chain data association. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 31:2196–2210, 2009.

[10] Takayoshi YAMASHITA Shihong LAO Masato KAWADE Yuan Li, Haizhou AI. Tracking in low frame rate video / tracking abrupt motion. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2007.