Evaluation of Low-Level Features for Real-World Surveillance Event Detection

Yang Xian, *Student Member, IEEE,* Xuejian Rong, *Student Member, IEEE,* Xiaodong Yang, *Member, IEEE,* and Yingli Tian, *Senior Member, IEEE*

***Abstract*—Event detection targets at recognizing and localizing specified spatio-temporal patterns in videos. Most research of human activity recognition in the past decades experimented on relatively clean scenes with limited actors performing explicit actions. Recently, more efforts have been paid to the real-world surveillance videos in which the human activity recognition is more challenging due to large variations caused by factors such as scaling, resolution, viewpoint, cluttered background, and crowdedness, etc. In this paper, we systematically evaluate 7 different types of low-level spatio-temporal features in the context of surveillance event detection using a uniform experimental setup. Fisher Vector is employed to aggregate low-level features as the representation of each video clip. A set of Random Forests is then learnt as the classification models. To bridge the research efforts and real-world applications, we utilize the NIST TRECVID Surveillance Event Detection (SED) as our testbed in which 7 events are pre-defined involving different levels of human activity analysis. Strengths and limitations for each low-level feature type are analyzed and discussed.**

***Index Terms*—Surveillance event detection, Low-level feature evaluation, Fisher Vector, Random Forests, Human activity recognition.**

I. Introduction

RECOGNIZING human activities has been widely applied to a number of practical applications including surveillance event detection. Event detection aims at recognizing and localizing specified spatio-temporal patterns in videos [1]. Automatic event detection of video surveillance has a variety of security applications for both private and public areas, e.g., house, airport, bank, supermarket, etc. Along with the rapid deployment of huge amounts of surveillance cameras, security agencies are seeking intelligent solutions to assist or replace human operators for the conventional surveillance systems which heavily demand human monitors. In the past decades, most research of human activity recognition experimented on relatively simple and clean scenes where only a limited number of actors performing explicit actions [2], [3], [4], [5]. This constrained scenario however rarely holds in the real-world surveillance videos. It is of great challenge to recognize human activities from surveillance videos captured in the wild due to large variations caused by different factors such as scaling, resolution, viewpoint, occlusion, cluttered background, and imbalanced data. Fig. 1 presents several examples captured by surveillance cameras. As observed, it is even a challenging task for human experts to recognize certain human activities.

In the proposed evaluation framework, we first extract low-level features, and code these features over a visual dictionary, then pool the codes in some pre-defined space-time cells. Most recent activity recognition approaches hinge on the bag-of-visual-words (BOV) representation which consists of computing and aggregating statistics from local spatio-temporal features [6]. In the basic framework of BOV, K-means is used to learn a visual dictionary and hard-assignment is employed to quantize low-level features. A set of more robust coding methods is then proposed to reduce information loss by relaxing the restrictive cardinality constraint in encoding low-level features, e.g., soft-assignment [7], sparse coding [8], and locality-constrained linear coding [9]. A specific coding method can be coupled with either average-pooling or max-pooling. Recently, several more effective coding methods have emerged to encode low-level features by recording the differences between features and visual words, e.g., Fisher Vector [10], vector of locally aggregated descriptors [11], and super sparse coding vector [4]. These approaches usually retain higher order statistics compared to the traditional coding methods. Extensive evaluations have shown that these approaches could achieve noticeably better recognition results in both image and video classification tasks [10], [4]. A significant progress has also been made in the development of low-level spatio-temporal features. A number of papers have flourished and reported the state-of-the-art results on various benchmarks. While several evaluations of low-level features on action recognition [12] and multimedia event classification [13] have been reported in the past, the efficacy of each individual low-level feature has not been systematically evaluated on the complex and unconstrained surveillance videos. We attempt to fill the gap of missing systematic low-level feature evaluation over real-world surveillance videos in this paper.

It is well known that the performance of a visual recognition system strongly depends on all stages of the pipeline. In this paper, we aim to evaluate and compare the low-level spatio-

Manuscript received Oct. 30, 2015, revised on Mar. 22, 2016. This work was supported in part by NSF grants EFRI-1137172, IIP-1343402, IIS-1400802, and FHWA grant DTFH61-12-H-00002.

Y. Xian is with the Department of Computer Science, The Graduate Center, The City University of New York, New York, NY, 10016 USA e-mail: yxian@gradcenter.cuny.edu.

X. Rong is with the Department of Electrical Engineering, The City College, The City University of New York, New York, NY, 10031 USA e-mail: xrong@ccny.cuny.edu.

X. Yang is with NVIDIA Research, Santa Clara, CA, 95050 USA e-mail: xiaodongy@nvidia.com.

Y. Tian is with the Department of Electrical Engineering, The City College and the Department of Computer Science, The Graduate Center, The City University of New York, New York, NY, 10031 USA phone: 212-650-7046; fax: 212-650-8249; e-mail: ytian@ccny.cuny.edu.

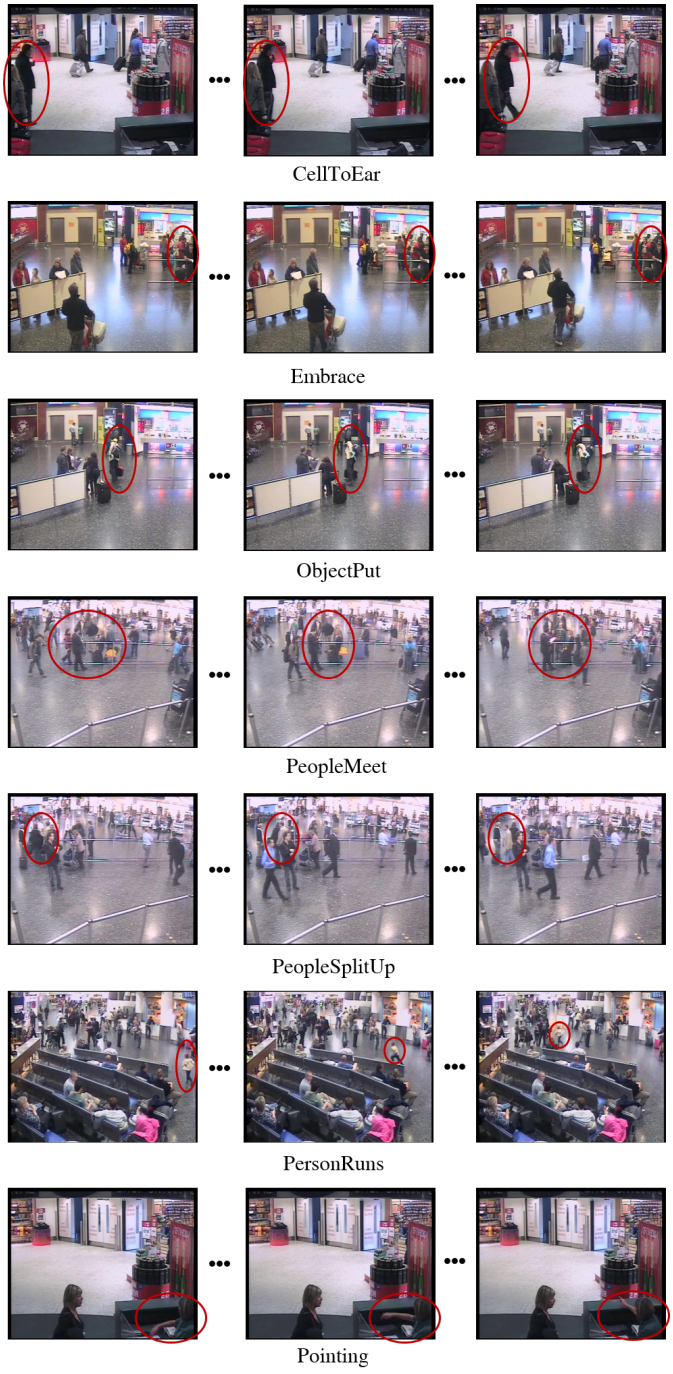


Fig. 1: Samples of the 7 events in surveillance videos from NIST TRECVID SED task. Persons involved in the identified events are circled in red. Due to the cluttered background, limited resolution, and other factors presented, it is extremely challenging to recognize certain human activities even for human experts.

temporal features in the context of surveillance event detection using a common experimental setup. In particular, we consider 7 types of low-level spatio-temporal features: the Space-Time Interest Points (STIP) [14], Motion Scale-invariant Feature Transform (MoSIFT) [15], Action Histograms of Oriented Gradients (Action-HOG) [16], as well as Trajectory (TRA), Histograms of Oriented Gradients (HOG), Histograms of Optical Flow (HOF), and Motion Boundary Histogram (MBH) from dense trajectories [17], [18][1]. Fisher Vector is employed to aggregate the low-level features as the representation of each video clip. In order to handle the imbalanced nature of the surveillance data (i.e., negative samples $\gg$ positive samples), we propose the ensemble of Random Forests [19] as the learning model, which partitions the training data into balanced chunks and learns a Random Forest within each data portion. Detection performance for each low-level feature type at a variety of pre-defined events is evaluated utilizing a common testing dataset measured in several different metrics.

To bridge the research efforts and real-world applications, we utilize the NIST TRECVID [20], [21] Surveillance Event Detection (SED) as our testbed. SED provides a corpus of 144-hour videos under 5 cameras views from the London Gatwick International Airport. In this dataset, 99-hour videos are provided with annotations of temporal spans and event labels and are divided into the development set and the testing set. Our experiments are based on all the 7 pre-defined events, i.e., PersonRuns, CellToEar, ObjectPut, PeopleMeet, PeopleSplitUp, Embrace, and Pointing. Within the testbed, we perform the evaluation over all the pre-defined events, i.e., to determine the temporal localization of each specific event, which is meaningful to the applicability to surveillance video indexing.

The remainder of this paper is organized as follows. Section II introduces the 7 types of low-level features evaluated in this paper. Section III presents the event detection system utilized to evaluate the detection performance of each low-level feature. Experimental results and discussions are presented in Section IV. Section V summarizes our observations including the future work.

II. Low-Level Features

In this paper, we extract and evaluate 7 most widely used local spatio-temporal features, namely, four types of dense trajectory based features [17], [18], including TRA, HOG, HOF, and MBH, and three other features which are STIP [14], MoSIFT [15], and Action-HOG [16]. To evaluate the performance of each low-level feature type, we first extract them from the raw video data respectively. Recent work on action recognition demonstrated that local spatio-temporal features are more effective and robust to image degradations, occlusion, illumination inconsistency, and cluttered background as compared with global features such as shape descriptors and contour representations, due to the capability of encoding both the appearance and the motion information of the objects in the continuous frames. A spatio-temporal feature extraction generally includes two phases: detection (i.e., a feature detector localizes interest points in a spatio-temporal space) and description (i.e., a feature descriptor computes representations of the detected points).

[1]In later context, for ease of simplicity, we use HOG, HOF, and MBH to represent dense trajectory based HOG, HOF, and MBH.

A. Dense Trajectory based Features

Dense trajectory based features were originally introduced by Wang *et al.* in [18] as an alternative to interest point based detectors. Due to different characteristics of the 2D spatial domain and the 1D time domain, it is more intuitive to track 2D interest points through time rather than to directly detect 3D interest points in videos. To better combine the power of the conventional 3D volume representations and the dense trajectory feature, extra descriptors are computed within a space-time volume around the trajectory, as described in the subsections below.

1) TRA: The trajectory feature is obtained by tracking densely sampled points using optical flow fields. Inspired by the improvements in the dense sampling based image classification approaches over the sparse sampling based ones, it was proposed in [18] to adopt densely sampled trajectories that are extracted in multiple spatial scales. Feature points are sampled on a grid spaced by W pixels both horizontally and vertically, and tracked in each scale separately. Each point at one frame is tracked to the next frame through median filtering in a dense optical flow, which is more robust than commonly used bilinear interpolation especially when tracked points are near motion boundaries. Once the dense optical flow field is computed, points can be tracked densely enough without additional cost. Points of subsequent frames are concatenated to form a trajectory. The shape of a trajectory with length L encodes local motion patterns, and could be described as a sequence $S = (\Delta P_t, ..., \Delta P_{t+L-1})$ of displacement vector $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. The generated vector is further normalized by the sum of magnitude and referred as the trajectory descriptor.

In the evaluation, we set $W = 5$ and adopt 8 spatial scales spaced by a factor of $\frac{1}{\sqrt{2}}$, which is the same as in [18]. Moreover, since trajectories tend to drift from their initial locations during tracking, we limit the length of a trajectory to a length of $L = 15$ to avoid drifting as suggested in [18]. In our experimental setting, the dimension of each trajectory feature is set to 30.

2) HOG: Among the existing descriptors for action recognition, HOG [6] has shown good and stable performance in representing local appearance information. The distribution of local intensity gradients or edge directions is effective in describing object appearance and shape information.

We derive HOG along the dense trajectories, and quantize the whole range of orientations $[0, 360^\circ)$ into 8 bins. The descriptor is further normalized with the L_2 norm, and the size of the space-time volume around the trajectory is $N \times N$ pixels and endures L frames. To embed structure information in the representation, the volume is subdivided into a spatio-temporal grid of size $n_\sigma \times n_\sigma \times n_\tau$. The parameters in our experiments are $N = 32, L = 15, n_\sigma = 2, n_\tau = 3$. Therefore, in our evaluation the final feature dimension of each interest point is 96.

3) HOF: Compared with HOG, HOF descriptor [6] tends to capture the local motion information instead of the static appearance information. Optical flow [22] measures the pattern of apparent motion of image objects between two consecutive frames caused by the movements of the objects or the camera. It is a 2D vector field in which each entry is a displacement vector representing the movement of points between the two consecutive frames. In the HOF descriptor generation, the dense optical flow that has already been computed to extract dense trajectories, is reused to boost the efficiency of the computation process.

In our evaluation, the orientations of HOF are quantized into 8 bins similar to HOG, but with an additional zero bin (i.e., 9 bins in total). The descriptor normalization is also based on the L_2 norm. Routine of the volume subdivision is the same as in HOG and the feature dimension is 108.

4) MBH: Since optical flow computes the absolute motion and will inevitably introduce the camera motion, Dalal *et al.* [23] proposed the MBH descriptor for human detection where derivatives are computed separately for the horizontal and vertical components of the optical flow. Hereby MBH is considered another top-performance descriptor besides HOG and HOF to further describe the dense trajectories. Since it also represents the gradient of the optical flow, the constant motion information mainly caused by the camera motion is suppressed and only the clue for variations in the flow field (i.e., motion boundaries) is kept for further analysis.

The MBH descriptor separates the optical flow field $I_\omega = (I_x, I_y)$ into its x and y components. Spatial derivatives are computed for each of them and the orientation information is quantized into histograms, similarly to the HOF descriptor. In the experiments, the 8-bin histogram is derived for each component, and then is normalized separately with L_2 norm. The final feature dimension for each interest point is 192.

B. STIP

The STIP detector was proposed by Laptev and Lindeberg in [14], [24], which extends the Harris detector [25] in the space-time domain. Firstly the points with large gradient magnitude are extracted by the 3D Harris corner detector in both the spatial and the temporal domains, and afterwards, a spatio-temporal second-moment matrix is computed at each interest point. The descriptors used to describe STIP interest points are HOG/HOF descriptors. To characterize the local motion and appearance information, histograms of spatial gradient and optical flow accumulated in space-time neighborhoods of detected interest points are computed by the detector in STIP volumes. The features are extracted by searching interest points with significant variations in both spatial and temporal domains.

The descriptor size of HOG/HOF is defined by $\Delta_x(\sigma) = \Delta_y(\sigma) = 18\sigma$ and $\Delta_t(\tau) = 8\tau$, with scales fixed as $\sigma^2 = 4$ and $\tau^2 = 2$. Each STIP volume is subdivided into $3 \times 3 \times 2$ cuboids as suggested in [6]. A 4-bin histogram of gradient orientation and a 5-bin histogram of optical flow are further generated on each cuboid, and concatenated as the final HOG/HOF descriptor after normalization. The final feature dimension for each interest point is $3 \times 3 \times 2 \times 9 = 162$.

C. Action-HOG

It is quite restrictive to assume the existence of large intensity changes in both spatial and temporal domains since

the detected points are usually very sparse and insufficient in the action recognition tasks. Therefore, instead of utilizing the spatio-temporal volumes, in Action-HOG [16], spatial and temporal information are extracted separately, i.e., first apply the Speed Up Robust Features (SURF) detector and then followed by the Motion History Image (MHI) filtering and HOG. The SURF detector [26] is firstly exploited to extract visually distinctive points in the spatial domain. Compared with the 2D Harris detector, SURF detector performs better by generating additional scale information while maintaining computational efficiency. The dominant orientations of interest points which serve as motion directions also provide important clues for action recognition. These extracted SURF points are then filtered by MHI [16] generated by differentiating adjacent frames, i.e., only SURF points with the most recent motions or large MHI intensities are selected as the interest points. In particular, MHI is a real-time motion template generated by stacking consecutive frame differences [27]. The brighter pixels on MHI correspond to the more recent motions. MHI gradients also reflect directional information of human actions. Therefore, the pixels with relatively larger intensities in MHI represent moving objects with more recent motion. To characterize the shape and motion information, HOG features are then computed for each interest point on both the image channel and the MHI channel. In addition, due to the specific camera views and scenes, occurrence of the specific events is usually biased to a certain range of locations. We further take advantage of this spatial prior to eliminate a large amount of interest points from the background.

In the experimental settings, each support region associated with an interest point on the image, MHI, and optical flow channels is subdivided into 3×3 grids. Image and MHI gradients are evenly sampled in 8 orientation bins. Therefore, each SURF/MHI-HOG point generates a feature vector of $3 \times (3 \times 3 \times 8) = 216$ dimensions.

D. MoSIFT

Scale-invariant Feature Transform (SIFT) feature was originally proposed by Lowe in [28] as a local image feature based on the appearance of the image at particular interest points. SIFT features are invariant to image scale and rotation. Moreover, they are robust to noise, blur, illumination variations, and minor changes in viewpoint. It performs well in static single image related tasks but still suffers from lacking of continuous inter-related temporal information. By contrast, the MoSIFT feature [15] introduces the multiple-scale optical flows which are calculated according to the SIFT scales, which effectively extends the SIFT descriptor with extra temporal information. MoSIFT is able to encode both the appearance and motion information of the objects and scenes in a video. The local appearance part is the same as the original SIFT. The local motion is computed through an optical flow pyramid constructed over two Gaussian pyramids. Optical flow is computed at multiple scales in concert with the SIFT scales. As long as a candidate SIFT interest point contains a small amount of movement, the algorithm will extract it as a MoSIFT interest point.

In the evaluation, to compensate the sheer volume of video data in the dataset and improve the processing efficiency, we resample all the videos to a standard size of 320×240 for dynamic feature extraction. The aggregated grids generated are further concatenated to form a 256-dimension vector via the public implementation [29] with default parameter settings.

III. Event Detection Evaluation System

In this section, we present the event detection system designed for the evaluation of the 7 low-level feature types introduced in Section II over all the 7 pre-defined events in TRECVID SED (examples shown in Fig. 1), i.e., PersonRuns, CellToEar, ObjectPut, PeopleMeet, PeopleSplitUp, Embrace, and Pointing.

A. System Overview

As illustrated in Fig. 2, the evaluation system consists of three main components: (1) low-level feature extraction, (2) video (sliding window) representation based on Fisher Vector, and (3) event learning and prediction by Random Forests.

In the event detection system, we evaluate the 7 low-level feature types as mentioned in Section II which are TRA, HOG, HOF, MBH, STIP, Action-HOG, and MoSIFT. 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 [30]. In this paper, Fisher Vector is employed to encode local spatial-temporal features. To accomplish data decorrelation and to reduce the computational complexity and memory consumption, we apply PCA to reduce the feature dimension by half over all listed low-level features but keeping the dimension of TRA before Fisher Vector representation.

Based on the above video representations, Random Forests [19] are utilized to learn the event-based models. However, the surveillance data is highly imbalanced for all events because positive events occur far less frequently than negative ones (statistics listed in Fig. 3). CellToEar and PeopleSplitUp are the least and most frequent events that occupy only 0.31% and 4.37% of the training video sequences, respectively. To overcome this extreme imbalance, in the offline learning phase, the original training data is disassembled into smaller chunks which are relatively more 'balanced'. A Random Forest classifier is learnt for each data segment. A straightforward weighted average process is performed to combine all the prediction results generated by multiple Random Forests in the online detection process.

B. Video Representation

After extracting the low-level features, we perform PCA for the dimensionality reduction of all features by half except TRA. Fisher Vector [31] is then employed to represent each sliding window. Fisher Vector encodes each feature descriptor by its deviation with respect to the parameters of a generative model and provides a feature aggregation scheme based on Fisher kernel that bounds the benefits of both generative and discriminative models.

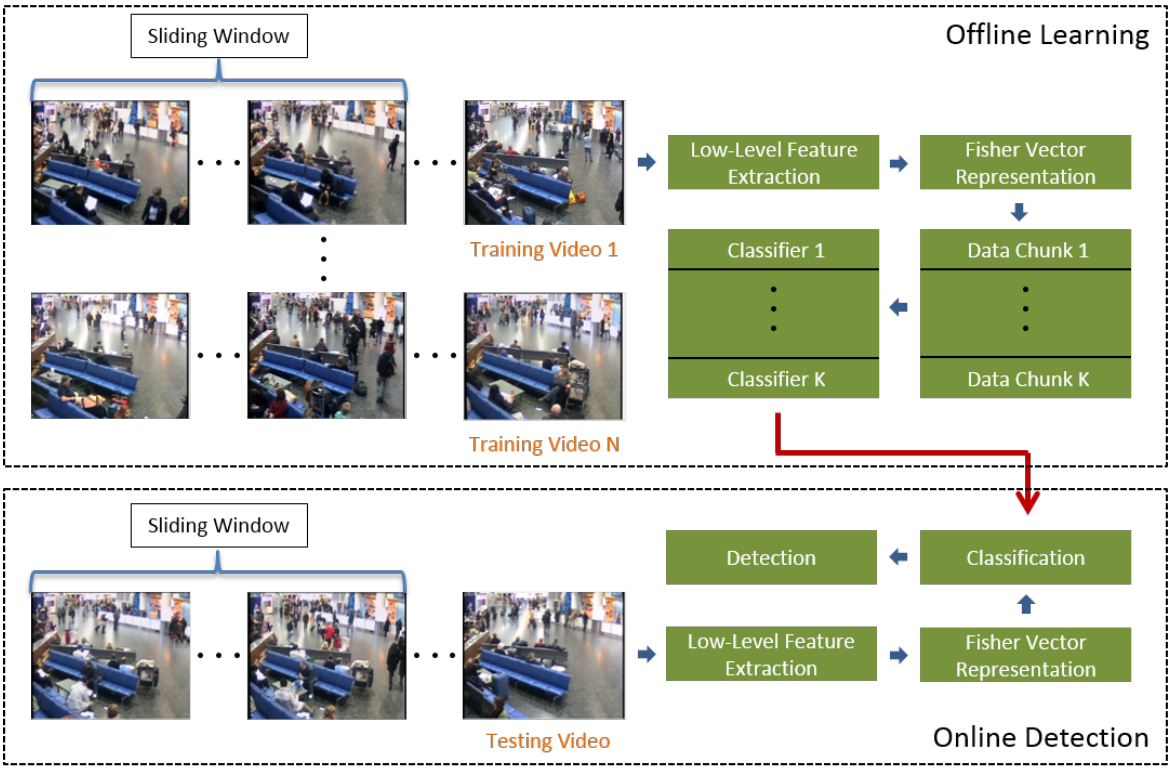


Fig. 2: Surveillance event detection system overview.

In the proposed framework, Gaussian mixture models (GMM) $G_\lambda(x) = \sum_{k=1}^{K} \pi_k g_k(x)$ are adopted as the generative model for the Fisher Vector in which g_k represents the kth Gaussian component:

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

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

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

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

The Fisher Vector of X is represented as $F(x) = \{\alpha_1, \beta_1, \ldots, \alpha_K, \beta_K\}$ where α_k and $\beta_k (k = 1, \ldots, K)$ are D-dimensional gradients with respect to the mean vector μ_k and the standard deviation σ_k of component k:

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

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

This system follows the scheme introduced in [31] to normalize the Fisher Vector, i.e., first the power normalization and followed with L_2 normalization. The L_2 normalization is adopted to remove the dependence on the proportion of event-specific information contained in a video, in other words, to cancel the effect of different amount of foreground and background information contained in different video segments. The power normalization is motivated by the observation that, with the increment of Gaussian component numbers, Fisher Vector becomes sparser which negatively impacts the dot-product in the following steps. Therefore, the power normalization $f(z) = \text{sign}(z)|z|^a$ with $0 < a \leq 1$ is applied to each dimension z in the Fisher Vector. We set $a = 0.5$ (i.e., the Hellinger kernel) to perform the signed square-rooting operation.

Compared with the BOV based approaches, Fisher Vector holds the following advantages:

- BOV is a particular case of Fisher Vector since only the gradient with respect to the mixture weights of GMM is utilized. The additional gradients with respect to the mean

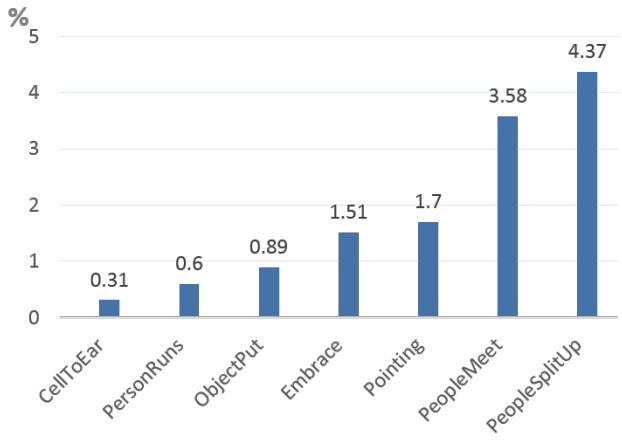


Fig. 3: Proportions of video sequences containing positive events in the training set.

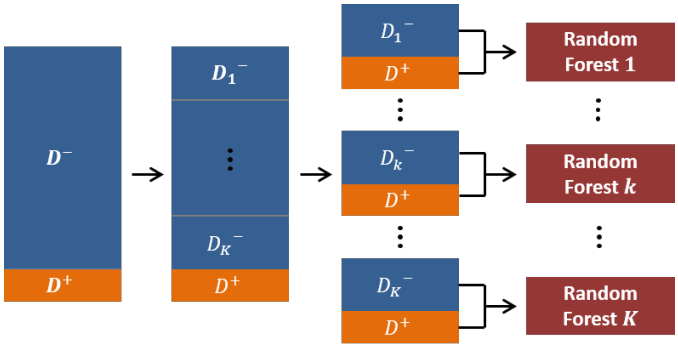


Fig. 4: Illustration of the data segmentation where within each data chunk a Random Forest is learnt.

vectors and standard deviations in Fisher Vector provide extra distribution information of descriptors.

- Fisher Vector can be computed upon a much smaller vocabulary compared with that used in BOV which facilitates a lower computational cost.
- Fisher Vector performs well with simple linear classifiers and is efficient in terms of both training and testing.

C. Model Learning and Post-Processing

In the event detection system, we adopt a 60-frame sliding window size strides in every 15 frames. This sliding window scheme generates highly imbalanced data. As shown in Fig. 3, among the evaluated events, 5 out of the 7 events cover less than 1.7% of the entire video sequences.

Event-dependent models are learnt to reduce intra-class variance and memory consumption in the training phase, namely, a set of Random Forests is trained for each of the 7 events utilizing training data combined from all 5 camera views. In order to handle the imbalanced data and make full usage of the valuable positive samples, we propose the following data segmentation scheme as illustrated in Fig. 4. For event i, the training data under all the camera views are combined and then separated according to the positive and negative labels. They are denoted as $D^i = \{D^{i^+}, D^{i^-}\}$. For simplicity concern, notation $D = \{D^+, D^-\}$ is utilized in later context. The negative data set is further segmented into a series of non-overlapped partitions $D_m^-, m = 1, \ldots, M$ with triple size of $|D^+|$. The whole training set is therefore chopped into a group of data segments in which each data chuck is composed of a portion of the negative samples and the whole positive data. A Random Forest is then trained within each data chunk. Therefore, for each low-level feature type under every event, a set of Random Forests is generated through learning upon segmented data chunks.

Given a testing video, all the 7 types of low-level features are extracted utilizing the same scheme as in the training step. Each low-level feature representation generates a corresponding Fisher Vector. Afterwards, all Fisher Vectors are fed into a group of pre-learnt Random Forests and a simple averaging is adopted to combine the prediction results from all classifiers where each prediction measures the probability of this window span contains this specific event detected. Generally, an event spans several different windows due to the fixed sliding window scheme utilized in our system and the time-lasting for different events varies. Therefore, after the classifier prediction, we employ a post-processing step to group continuous positive windows as to decide the final temporal interval of a detected event. To be more specific, two positive predictions which have overlaps in their sliding windows are merged together.

IV. Experimental Results

In this section, details of the dataset used for the evaluation is listed. Moreover, we present the detection performance over the 7 evaluated events for all the 7 low-level feature types utilizing the pipeline described in Section III. Strengths and limitations of each low-level feature are discussed.

A. Dataset and Parameter Settings

As mentioned in Section I, we utilize NIST TRECVID SED as our testbed. SED provides a corpus of 144-hour videos under 5 fixed camera views from the London Gatwick International Airport. It contains 7 pre-defined events, i.e., PersonRuns, CellToEar, ObjectPut, PeopleMeet, PeopleSplitUp, Embrace, and Pointing. These events represent three levels of human activity analysis: single person action (Pointing, PersonRuns), person-object interaction (CellToEar, ObjectPut), and multiple people activity (Embrace, PeopleMeet, PeopleSplitUp). As observed from samples of the events presented in Fig. 1, it is an extremely challenging task to detect a specific event with subtle movements and short durations in such crowded environment. Certain events take place far from the camera which causes a very limited resolution for the target people involved.

All videos provided by TRECVID SED are captured with the frame resolution 720×576 at 25 fps. Within the video corpus, 99-hour videos are provided with annotations of temporal extents and event labels. In our experiments, we further divide the annotated videos into two parts where half of the

data forms the training set and the rest half is utilized as the testing set to evaluate the detection performance over a specific event. 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. In the low-level feature extraction process, we downsample all videos to half of the original size in both horizontal and vertical directions. After performing PCA of all the 7 types of low-level features other than TRA to further reduce the feature dimension, we train GMM with 128 Gaussian components. Table I illustrates the dimensions of Fisher Vectors for all the 7 feature types. Within each Random Forest, the maximum depth of each tree is set to 5 and the maximum number of trees in the forest is 20.

B. Evaluation Methodology and Results

For each feature type i, under the event j, a set of Random Forests RF_i^j are learnt utilizing the annotated training data in NIST TRECVID SED dataset. Provided with a testing video, after generating Fisher Vector representations for feature i utilizing the sliding window scheme, Random Forests RF_i^j are employed to produce outputs for each sliding window under event j. The output of a certain sliding window measures the probability that these frames contain the detected event.

Top N_i^j sliding windows with the largest Random Forests scores are considered the positive predictions for feature type i under event j. Based on the empirical observations during the training, we adopt a universal setting of $2,000$ for all $N_i^j (i, j = 1 \dots 7)$. As mentioned in Section III.C, generally an event spans several sliding windows and therefore, among the $2,000$ selected instances, two positive predictions which have overlaps in their sliding windows are merged together. Therefore, the actual number of the positive predictions (i.e., #SysInp shown in Table III) varies for each case and normally are less than 2000 (refer to Table III for details).

Table II represents the numbers of True Positives, False Positives, and False Negatives for all feature types under all 7 events. To further evaluate the sliding window based performance, error rates are provided to illustrate how many sliding windows in the testing data are misclassified. As observed, since majority of the sliding windows do not contain any pre-defined events, the sliding window based error rates are small (less than 5% for most cases).

Table III presents the detection performance for all the evaluated low-level feature types of each event in which #CorDec, #SysInp, and #GT denote the number of correct detections, the number of actual system inputs after merging the top $2,000$ positive predictions, and the number of positive ground-truths. The F-score is adopted to evaluate the detection performance for each feature. F-score is widely employed to measure a test's accuracy which reaches its best value at 1 and the worst at 0 and is calculated utilizing precision and recall:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{5}$$

in which the precision and recall are computed with:

TABLE II: Comparison of the detection performance for different types of low-level features in the 7 events measured in the numbers of True Positives, False Positives, False Negatives, and the sliding window based error rates.

Event1: PersonRuns	#TruePos	#FalsePos	#FalseNeg	Error Rate
Action-HOG	42	1341	294	0.0196
HOF	54	1584	282	0.0162
HOG	67	1525	269	0.0161
MBH	66	1435	270	0.0162
MoSIFT	25	1571	311	0.0169
STIP	54	1521	282	0.0166
TRA	67	1653	269	0.0160
Event2: CellToEar	#TruePos	#FalsePos	#FalseNeg	Error Rate
Action-HOG	58	1198	329	0.0151
HOF	60	1467	327	0.0124
HOG	75	1562	312	0.0124
MBH	70	1496	317	0.0124
MoSIFT	43	1712	344	0.0129
STIP	68	1618	319	0.0127
TRA	42	1687	345	0.0125
Event3: ObjectPut	#TruePos	#FalsePos	#FalseNeg	Error Rate
Action-HOG	252	1427	1757	0.0392
HOF	283	1425	1726	0.0320
HOG	291	1368	1718	0.0320
MBH	308	1340	1701	0.0320
MoSIFT	236	1431	1773	0.0334
STIP	297	1383	1712	0.0327
TRA	267	1473	1742	0.0320
Event4: PeopleMeet	#TruePos	#FalsePos	#FalseNeg	Error Rate
Action-HOG	275	1390	1013	0.0697
HOF	341	1367	947	0.0565
HOG	319	1470	969	0.0568
MBH	348	1479	940	0.0565
MoSIFT	263	1330	1025	0.0592
STIP	292	1348	996	0.0581
TRA	348	1367	940	0.0564
Event5: PeopleSplitUp	#TruePos	#FalsePos	#FalseNeg	Error Rate
Action-HOG	135	1450	559	0.0726
HOF	197	1581	497	0.0593
HOG	225	1616	469	0.0592
MBH	216	1603	478	0.0590
MoSIFT	180	1611	514	0.0617
STIP	180	1416	514	0.0606
TRA	72	1297	622	0.0597
Event6: Embrace	#TruePos	#FalsePos	#FalseNeg	Error Rate
Action-HOG	58	1338	366	0.0293
HOF	97	1601	327	0.0237
HOG	62	1664	362	0.0240
MBH	103	1547	321	0.0236
MoSIFT	16	1523	408	0.0250
STIP	128	1554	296	0.0242
TRA	38	1653	386	0.0241
Event7: Pointing	#TruePos	#FalsePos	#FalseNeg	Error Rate
Action-HOG	376	1292	2040	0.0500
HOF	382	1371	2034	0.0411
HOG	390	1363	2026	0.0412
MBH	402	1300	2008	0.0411
MoSIFT	300	1408	2116	0.0428
STIP	403	1283	2013	0.0420
TRA	358	1355	2058	0.0411

$$\begin{aligned} \text{precision} &= \frac{\text{true positive}}{\text{true positive} + \text{false positive}}, \\ \text{recall} &= \frac{\text{true positive}}{\text{true positive} + \text{false negative}}. \end{aligned} \tag{6}$$

In Table III, for each event, the two low-level feature types with the top two largest F-scores are marked in bold for a clearer illustration.

To provide a more comprehensive comparison of all the evaluated low-level features, a breakdown of the time and

TABLE I: Dimensions of the Fisher Vector representations for all the 7 feature types: Action-HOG, HOF, HOG, MBH, MoSIFT, STIP, and TRA.

Feature	AHOG	HOF	HOG	MBH	MoSIFT	STIP	TRA
FisherVector Dim	27,648	13,824	12,288	24,576	32,768	20,746	7,680

TABLE III: Comparison of the detection performance for different types of low-level features in the 7 events. For each event, the two feature types with the largest F-Scores are marked in bold.

Event1: PersonRuns	#CorDet	#SysInp	#GT	Precision	Recall	F-Score
Action-HOG	42	1343	336	0.0313	0.1250	0.0500
HOF	54	1638	336	0.0330	0.1607	0.0547
HOG	67	1592	336	0.0421	0.1994	**0.0695**
MBH	66	1501	336	0.0440	0.1964	**0.0719**
MoSIFT	25	1596	336	0.0157	0.0744	0.0259
STIP	54	1575	336	0.0343	0.1607	0.0565
TRA	67	1720	336	0.0390	0.1994	0.0651
Event2: CellToEar	#CorDet	#SysInp	#GT	Precision	Recall	F-Score
Action-HOG	58	1256	387	0.0462	0.1499	0.0706
HOF	60	1527	387	0.0393	0.1550	0.0627
HOG	75	1637	387	0.0458	0.1938	**0.0741**
MBH	70	1566	387	0.0447	0.1809	**0.0717**
MoSIFT	43	1755	387	0.0245	0.1111	0.0401
STIP	68	1686	387	0.0403	0.1757	0.0656
TRA	42	1729	387	0.0243	0.1085	0.0397
Event3: ObjectPut	#CorDet	#SysInp	#GT	Precision	Recall	F-Score
Action-HOG	252	1679	2009	0.1501	0.1254	0.1367
HOF	283	1708	2009	0.1657	0.1409	0.1523
HOG	291	1659	2009	0.1754	0.1448	0.1587
MBH	308	1648	2009	0.1869	0.1533	**0.1684**
MoSIFT	236	1667	2009	0.1416	0.1175	0.1284
STIP	297	1680	2009	0.1768	0.1478	**0.1610**
TRA	267	1740	2009	0.1534	0.1329	0.1424
Event4: PeopleMeet	#CorDet	#SysInp	#GT	Precision	Recall	F-Score
Action-HOG	275	1665	1288	0.1652	0.2135	0.1863
HOF	341	1708	1288	0.1996	0.2648	**0.2276**
HOG	319	1789	1288	0.1783	0.2477	0.2073
MBH	348	1827	1288	0.1905	0.2702	0.2234
MoSIFT	263	1593	1288	0.1651	0.2042	0.1826
STIP	292	1640	1288	0.1780	0.2267	0.1995
TRA	348	1715	1288	0.2029	0.2702	**0.2318**
Event5: PeopleSplitUp	#CorDet	#SysInp	#GT	Precision	Recall	F-Score
Action-HOG	135	1585	694	0.0852	0.1945	0.1185
HOF	197	1778	694	0.1108	0.2839	0.1594
HOG	225	1841	694	0.1222	0.3242	**0.1775**
MBH	216	1819	694	0.1187	0.3112	**0.1719**
MoSIFT	180	1791	694	0.1005	0.2594	0.1449
STIP	180	1596	694	0.1128	0.2594	0.1572
TRA	72	1369	694	0.0526	0.1037	0.0698
Event6: Embrace	#CorDet	#SysInp	#GT	Precision	Recall	F-Score
Action-HOG	58	1396	424	0.0415	0.1368	0.0637
HOF	97	1698	424	0.0571	0.2288	0.0914
HOG	62	1726	424	0.0359	0.1462	0.0577
MBH	103	1650	424	0.0624	0.2429	**0.0993**
MoSIFT	16	1539	424	0.0104	0.0377	0.0163
STIP	128	1682	424	0.0761	0.3019	**0.1216**
TRA	38	1691	424	0.0225	0.0896	0.0359
Event7: Pointing	#CorDet	#SysInp	#GT	Precision	Recall	F-Score
Action-HOG	376	1668	2416	0.2254	0.1556	0.1841
HOF	382	1753	2416	0.2179	0.1581	0.1832
HOG	390	1753	2416	0.2225	0.1614	0.1871
MBH	402	1702	2416	0.2362	0.1664	**0.1952**
MoSIFT	300	1708	2416	0.1756	0.1242	0.1455
STIP	403	1689	2416	0.2386	0.1668	**0.1963**
TRA	358	1713	2416	0.2090	0.1482	0.1734

TABLE IV: Comparison of the time and space complexity for all low-level features in feature extraction, Fisher Vector representation, and Random Forests prediction steps measured based on $1,000$ frames.

	Feature Extraction		FV Generation		RF Prediction
	Time(sec)	Space(MB)	Time(sec)	Space(MB)	Time(sec)
Action-HOG	30.16	4.88	0.84	4.16	11.45
HOF	106.53	31.70	4.52	3.51	32.17
HOG	112.03	28.45	5.37	3.12	33.79
MBH	133.09	54.45	6.03	6.23	36.64
MoSIFT	801.37	4.89	4.26	8.25	39.02
STIP	359.04	12.96	6.52	5.22	32.51
TRA	93.60	10.32	4.87	1.95	24.94

space complexity for the key components in the evaluation pipeline is listed in Table IV. To be more specific, for a video consists of $1,000$ frames, the time cost of extracting each low-level feature[2] along with the space occupied are recorded. The time and space of the Fisher Vector representation generation process are also listed. Finally, we measure that during the testing phase, the time utilized for the Random Forests prediction process[3]. Generally speaking, the time consumed in the Random Forests prediction step is proportional to the dimensions of the Fisher Vector representations of each low-level feature. However, since Action-HOG does not extract features for certain frames in the testing video without sufficient motions, the time cost is significantly smaller compared with peer features. Moreover, based on the fact that the label files generated after the Random Forests predictions take the same amount of space for all the features, the space used in this step is not listed in Table IV.

C. Detection Performance Evaluation and Discussions

Generally speaking, as observed from Table III, group activities (e.g., PeopleMeet, PeopleSplitUp) are with higher F-scores due to a relatively higher ratio of the foreground objects in the scene. Since we employ a fixed number of system inputs (i.e., $2,000$ before merging post-processing), the mismatch between the actual number of system inputs and the number of the ground-truths is another factor that would affect the F-score values.

CellToEar is commonly considered the most challenging task among the 7 events evaluated [32]. It is a very subtle and short activity which begins with some person starting to move the phone to the ear and ends when the phone reaches the ear. Compared with ObjectPut, size of the object (i.e., cellphone) is very small and even unrecognizable if the resolution is limited. Therefore, false positives arise when the detected person tries to reach his/her head or ear with empty hand. As observed from the provided training data, even ground-truths contain mislabels and ambiguities due to chaotic surroundings in which various event-unrelated human behaviors occur.

Among the evaluated 7 types of features, TRA, HOG, HOF, and MBH utilize dense sampling while the rest three (i.e., Action-HOG, MoSIFT, and STIP) sample on sparse points. Generally speaking, for image and video event detection purpose, dense sampling works better compared with sparse sampling. However, as demonstrated in Table IV, it brings in higher storage consumption and computation costs.

As revealed from the detection performance shown in Table III, STIP outperforms MoSIFT and Action-HOG measured in F-score in 6 events: PersonRuns, ObjectPut, PeopleMeet, PeopleSplitUp, Embrace, and Pointing. Strong constraints to spatio-temporal interest points are not posed by the MoSIFT detector during detection. A MoSIFT interest point is extracted as long as a candidate SIFT interest point contains a minimal amount of movement. Similarly, spatial and temporal information is separated during Action-HOG detection utilizing SURF and MHI-HOG respectively. However, on the other hand, STIP detector computes a spatio-temporal second-moment matrix at each point which successfully bounds both spatial and temporal information. Moreover, when computing descriptors, MoSIFT encodes the appearance of the objects/scenes and the motion information of only one frame while STIP accumulates more appearance and motion information in the temporal scale.

Although both MoSIFT and Action-HOG fail to pose strong constraints to the bounding of spatial and temporal information, Action-HOG reveals superiority over MoSIFT measured in time and space complexity. Our experimental results demonstrate that Action-HOG (i.e., SURF/MHI-HOG) runs beyond 10 times faster in terms of processing each frame and around 20 times faster in terms of computing each interest point than MoSIFT (i.e., STIP-HOG/HOF).

Dense trajectory based MBH appears to be the feature with the best performance considering the detection results in all the 7 events. This is probably due to the fact that MBH represents the gradients of the optical flow and reserves the motion clues mostly in the motion boundary which leads to a more accurate and discriminative motion estimation. However, as demonstrated in Table IV, compared with other low-level features, MBH consumes significant larger amount of memory in the feature extraction step. In terms of describing the motion, MBH and HOF provide more explicit information compared with HOG. On the other hand, in events that involve objects (i.e., CellToEar and ObjectPut), HOG conveys strong appearance cues and yields relatively better detection performance compared with other events.

Densely sampled trajectories are extracted in multiple spatial scales which are simple descriptions of motion patterns. Among the 4 evaluated dense trajectory based features, TRA consumes the least amount of time and space for almost all the steps. Compared with activities that involving subtle and short movements, trajectories are more effective and complementary in the events with large and consistent motions (i.e., PersonsRuns, PeopleMeet) as demonstrated by the detection performance.

[2]Please note that in our experiments to evaluate the performance of the low-level features, we use https://lear.inrialpes.fr/people/wang/dense_trajectories to extract dense trajectory based features at the same time. However, in order to measure the time and space complexity of extracting each feature as shown in Table IV, we modify the original source code and compute the 4 dense trajectory based features separately.

[3]For each low-level feature, 36 pre-trained Random Forests under Event 1 are utilized to measure the time complexity in the Random Forests prediction step shown in Table IV.

V. Conclusions and Future Work

In this paper, we have systematically evaluated the detection performance of 7 different low-level feature types utilizing NIST TRECVID Surveillance Event Detection dataset which contains a variety of challenging events.

A set of Random Forests is learnt through a uniformed pipeline for each low-level feature type in each event. The event detection performance is then evaluated utilizing the trained Random Forests over the testing videos measured with F-scores of the positive predictions.

As observed from the detection results for all the 7 events, dense sampling works better compared with sparse sampling. STIP outperforms MoSIFT and Action-HOG in 6 out of the 7 events since it bounds both spatial and temporal information in detection and accumulates more appearance and motion information in computing descriptors. Generally speaking, among all the events, MBH appears to be the best performing feature type. HOG conveys strong appearance cues in events involving objects. TRA can be effective in events involving large and consistent motions.

Deep models have gained a growing interest in action detection and recognition. In the future, we will investigate more in deep learning based classification methods in real-world surveillance event detection with more complicated events involved.

References

[1] Y. Ke, R. Sukthankar, and M. Herbert, "Event detection in crowded videos," in *ICCV*, 2007.

[2] G. Cheron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN features for action recognition," in *ICCV*, 2015.

[3] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.

[4] X. Yang and Y. Tian, "Action recognition using super sparse coding vector with spatio-temporal awareness," in *ECCV*, 2014.

[5] Y. Ye, X. Yang, and Y. Tian, "Exploring pooling strategies based on idiosyncrasies of spatio-temporal interest points," in *ICMR*, 2015.

[6] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[7] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *ICCV*, 2011.

[8] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *ICML*, 2009.

[9] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[10] F. Perronnin, J. Sanchez, and T. Mensink, "Improving the Fisher kernel for largescale image classification," in *ECCV*, 2010.

[11] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010.

[12] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[13] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. Sawhney, "Evaluation of low-level features and their combinations for complex event detection in open source videos," in *CVPR*, 2012.

[14] I. Laptev and T. Lindeberg, "Space-time interest points," in *ICCV*, 2003.

[15] M. Chen and A. Hauptmann, "MoSIFT: Recognizing human actions in surveillance videos," in *CMU-CS-09-161*, 2009.

[16] X. Yang, Z. Liu, E. Zavesky, D. Gibbon, B. Shahraray, and Y. Tian, "AT&T research at TRECVID 2013: Surveillance event detection," in *NIST TRECVID Workshop*, 2013.

[17] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[18] H. Wang, A. Klser, C. Schmid, and C. L. Liu, "Action recognition by dense trajectory," in *CVPR*, 2011.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] A. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *ACM Workshop on Multimedia Information Retrieval*, 2006.

[21] Y. Xian, X. Rong, X. Yang, and Y. Tian, "CCNY at TRECVID 2014: Surveillance event detection," in *NIST TRECVID Workshop*, 2014.

[22] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills, "Recovering motion fields: An evaluation of eight optical flow algorithms," in *BMVC*, 1998.

[23] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *ECCV*, 2006.

[24] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, 2005.

[25] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[26] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *ECCV*, 2006.

[27] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

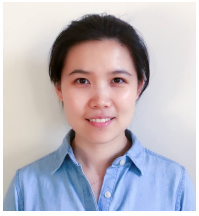
[28] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999.

[29] http://lastlaugh.inf.cs.cmu.edu/libscom/downloads.htm, 2009.

[30] K. Chatfield, V. Lemtexpitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods," in *BMVC*, 2011.

[31] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[32] 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," in *ICMR*, 2014.



Yang Xian received the B.E. degree from Southeast University, Nanjing, China, in 2009 and the M.S. degree from New York University, New York, USA, in 2012. She is currently pursuing the Ph.D degree in Computer Science at the Graduate Center, the City University of New York, New York, USA. Her research interests are in the synergic areas of computer vision, computational photography, and machine learning. She has been working on large-scale surveillance event detection and designing quality enhancement algorithms, i.e., super-resolution, completion, for both RGB images and depth maps.

Xuejian Rong is a third year Ph.D. Candidate in the Department of Electrical Engineering at the City College, City University of New York, New York, NY, USA. He received the B.E. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. For now, he is working with Prof. Yingli Tian in the intersection of Machine Learning and Computer Vision. His current research interests mainly focus on inference and learning for scene text detection and recognition in the wild, in the presence of image degradations like the blur, distortion, noise, etc.

Xiaodong Yang received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 2009, and the Ph.D. degree from the Department of Electrical Engineering at City College, City University of New York, New York, NY, USA, in 2015. He joined NVIDIA Research, Santa Clara, CA, USA, as a research scientist, in 2015. His current research interests include computer vision, machine learning, deep learning, and multimedia analytics. He has been working on large-scale image and video classification, hand gesture and human action recognition, video surveillance event detection, multimedia search, and computer vision based assistive technology.

YingLi Tian received the B.S. and M.S. degrees from Tianjin University, China, in 1987 and 1990, and the Ph.D. degree from Chinese University of Hong Kong, Hong Kong, in 1996. After holding a faculty position at National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, she joined Carnegie Mellon University in 1998, where she was a postdoctoral fellow at the Robotics Institute. She then worked as a research staff member in IBM T. J. Watson Research Center from 2001 to 2008. She is one of the inventors of the IBM Smart Surveillance Solutions. She is currently a professor in the Department of Electrical Engineering at City College and Graduate Center, City University of New York. Her current research focuses on a wide range of computer vision problems from motion detection and analysis, assistive technology, to human identification, facial expression analysis, and video surveillance.