FAST DETECTION OF ABNORMAL EVENTS IN VIDEOS WITH BINARY FEATURES

Roberto Leyva [†], *Victor Sanchez* [†] *and Chang-Tsun Li* [‡]

[†] Department of Computer Science, University of Warwick, United Kingdom
[‡] School of Computing and Mathematics, Charles Sturt University, Australia

ABSTRACT

Millions of surveillance cameras are currently installed in public places around the world, making it necessary to intelligently analyse the acquired data to detect the occurrence of abnormal events. A vast number of methods to detect such events have been recently proposed; unfortunately, there is a lack of methods capable of detecting these events as frames are acquired, also known as online processing. In this paper, we present an online framework for video anomaly detection that employs binary features to encode motion information, and low-complexity probabilistic models for detection. Evaluation results on the popular UCSD dataset and on a recently introduced real-event video surveillance dataset show that our framework outperforms non-online and online methods.

***Index Terms*—** Video anomaly detection, online processing, binary features, surveillance video.

1. INTRODUCTION

Big data continues to grow exponentially and surveillance video has become one of the largest sources [1]. This is evident by the increasing number of surveillance cameras throughout our surroundings, e.g., in elevators, ATMs, and other public places. To fully exploit the data acquired by these cameras, it is important to develop automatic video surveillance methods capable of intelligently analyzing and understanding the visual information. At the core of automatic video surveillance is anomaly detection, which aims at detecting unusual events without any a priori knowledge. Such events may include riots, robberies, fights, traffic accidents and other dangerous situations [2, 3].

Despite its many advantages, video anomaly detection is still far from being practical in real-time scenarios due to its high computational complexity. Many state-of-the-art video anomaly detection methods are not capable of classifying frames as they are acquired at a specific frame rate. Consequently, the data are usually manually analyzed in real-time, hindering the benefits of current surveillance cameras [2].

The long computational times of many existing methods are mainly due to the operations required to 1) extract feature descriptors from the data, and 2) process these features descriptors by trained classifiers. Developing low-complexity feature descriptors and classifiers is a key aspect that can increase the practicality of video anomaly detection methods for online processing [2, 3]. In this paper, we then propose a video anomaly detection framework suitable for online processing. The contributions of our work are: **(I)** Two sources of motion are used to classify events; i.e., background and temporal gradients. **(II)** The background is encoded only for regions depicting foreground, thus reducing this type of features. **(III)** The temporal gradients are encoded into binary features, which are known to be fast to compute and process. **(IV)** Event detection is attained by using multiple low-complexity probabilistic models.

The proposed framework is tested on the UCSD dataset and on the LV dataset, which is a new collection of surveillance videos depicting real events. Results show that our framework attains online processing outperforming other online methods, particularly on real surveillance videos. The rest of the paper is organized as follows. Section 2 reviews common techniques used to detect abnormal events in videos. Section 3 details our proposed framework. Section 4 presents the evaluation results and Section 5 concludes the paper.

2. PREVIOUS WORK

Video anomaly detection usually relies on analyzing and modeling the motion information of several spatio-temporal support regions compacted into feature descriptors. Common sources of motion information include optical flow [4–7], temporal gradients [2, 8], and dynamic textures [9, 10]. Methods commonly used to define spatio-temporal regions include those based on dense sampling [2, 8], Hessian convolution [5, 6], and fixed-size cells [11, 12]. Modeling these spatio-temporal support regions is usually done via dictionary modeling [2, 5–8] and sparse reconstructions [4, 7, 9, 10] that evaluate the likelihood of a particular observation as a statistic inference problem. Although state-of-the-art video anomaly detection methods can attain an outstanding performance [2, 4, 11, 13], long computational times and high demands for computational resources make them unsuitable for online processing [2, 4, 11, 12]. On the other hand, methods aimed at attaining online processing, e.g., [14, 15], significantly sacrifice accuracy by reducing the complexity of their motion

Thanks to the Mexican Ministry of Education - CONACYT - 372028 and the EU Horizon 2020 project IDENTITY, Project ID: 690907

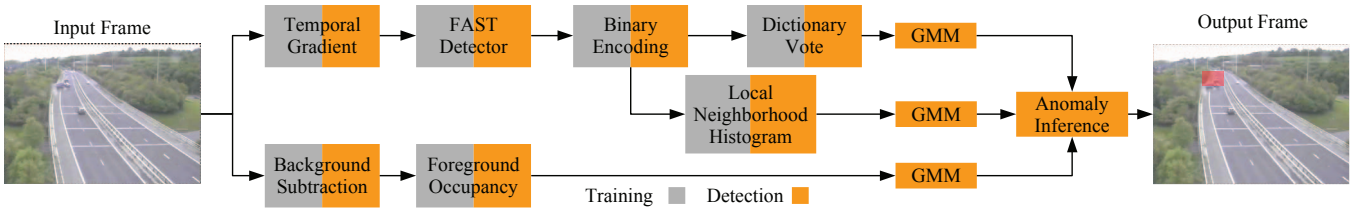


Fig. 1. Proposed framework. The temporal gradients and background of frames are calculated. Interest points are detected by using the FAST detector. Binary encoding then generates binary features, which are used to create dictionaries. GMMs are used to model all binary features and those obtained by computing the foreground occupancy. An inference mechanism that uses GMM votes detects abnormal events.

sources, feature descriptors and classifiers.

3. PROPOSED FRAMEWORK

Our framework [1] is graphically depicted in Fig. 1. It first extracts features from the scene, which are used to construct a visual dictionary and several probabilistic models. It then uses an inference mechanism to detect abnormal events.

3.1. Feature Extraction

Features are computed from two motion sources: the background and temporal gradients. For both, we define spatio-temporal support regions of different size by using a grid of cells divided into three regions, $\{R_1, R_2, R_3\}$ (see Fig. 2a). Cells of different size help to compensate for the apparent change of objects' size as they approach the camera, under the assumption that the lower part of the scene is closest to the camera [3, 12, 16]. This allows us to avoid using a multi-scale procedure, as in [2, 8, 13].

Background: We create a video volume u_q of size $n_x \times n_y \times n_t$ for each cell of the grid (see Fig. 2a). We then compute the foreground occupancy, $O_q \in \mathbb{R}^1$, of u_q as follows:

$$O_q(u_q) = \frac{1}{N} \sum_{0<n\leqslant N} u_q^{(n)}, \tag{1}$$

where N is the number of pixels in u_q. Only those video volumes with a foreground occupancy $\geq 10\%$ are further processed, which helps to reduce computational times.

Temporal gradients: To detect interest points, we use the binary detector Fast Accelerated Segmentation Test (FAST), whose computational times are in the order of milliseconds per frame [17]. As illustrated in Fig. 2b, each FAST point defines the center (x_p, y_p) of a video volume, v_p, of size $n_x \times n_y \times n_t$, where the spatial size $n_x \times n_y$ is determined by the cell in which the FAST point is detected. Encoding v_p by employing double-precision feature descriptors; e.g., HOG, HOF, and MBH [18, 19], results in very computationally expensive subsequent steps [4]. For this reason, we employ binary features to reduce processing times. Specifically, we encode v_p by using the binary descriptors Binary Wavelet Differences (BWD) and Binary Centroid Tracker (BCT) [20, 21],

[1] End-to-end implementation available at: https://cvrleyva.wordpress.com/

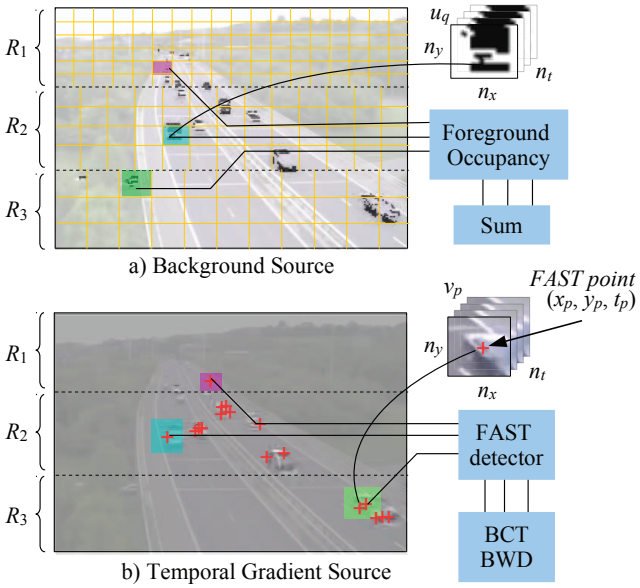


Fig. 2. Motion sources and feature extraction. a) Foreground Occupancy is extracted from the background. b) The FAST detector is applied to the temporal gradient and the spatio-temporal support regions are encoded with BWD and BCT.

which can encode video volumes in the order of microseconds.

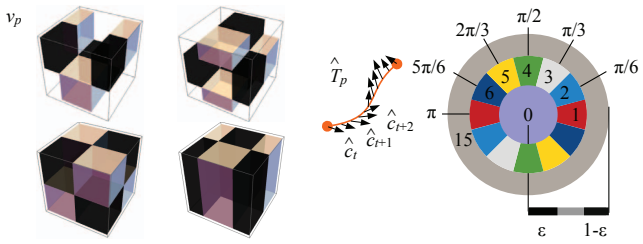
To capture the vertical and horizontal motion components of v_p, we first apply two Prewitt differential operators to v_p to generate video volumes v_p^x and v_p^y. We then encode v_p^x and v_p^y by using BWD, which generates a binary string, G_p^n, by comparing the values of $M = 32$ pairs of regions defined within v_p^n, with $n \in \{x, y\}$. Each pair P_m groups pixel locations into one of two regions according to a wavelet pattern (see Fig. 3a). Each bit of string G_p^n is computed as follows:

$$C(P_m) = \begin{cases} 1, & \text{if } \Sigma(P_m^1) > \Sigma(P_m^2) \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where $\Sigma(P_m^r)$ represents the sum of values of region $r \in \{1, 2\}$. G_p^n is then the concatenation of the M comparisons:

$$G_p^n = \sum_{0\leqslant m<M} 2^m C(P_m). \tag{3}$$

It is important to note that the displacements in time of the centroid of video volume v_p generate a trajectory. We



a) Binary Wavelet Differences (BWD) b) Binary Centroid Tracker (BCT)

Fig. 3. a) Sample patterns used in BWD to define two regions within a video volume. b) Bins used in BCT to encode trajectory $\hat{T}_p$.

use BCT to encode this trajectory into a binary string. The centroid of v_p at time t, denoted by c_t, is first computed as:

$$c_t = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}), \quad m_{ab} = \sum_{\substack{1 \leqslant x \leqslant n_x \\ 1 \leqslant y \leqslant n_y}} x^a y^b v_p(x, y, t), \tag{4}$$

and the normalized trajectory $\hat{T}_p$, comprising L displacement vectors, $\Delta c_t ... \Delta c_{t+L-1}$, is computed as follows:

$$\hat{T}_p = \frac{(\Delta c_t, ..., \Delta c_{t+L-1})}{\max \|\Delta c_t\|}. \tag{5}$$

BCT encodes each normalized displacement vector of $\hat{T}_p$, i.e., each $\Delta \hat{c}_t = \Delta c_t / \max \|\Delta c\|$, by using a binary binning scheme:

$$B(\Delta \hat{c}_t) = \begin{cases} k : \arg\min_k \parallel \Delta \hat{c}_t - b_k \parallel & : |\Delta \hat{c}_t| \in (\epsilon, 1-\epsilon) \\ 15 & : |\Delta \hat{c}_t| \geq 1 - \epsilon \\ 0 & : |\Delta \hat{c}_t| \leq \epsilon \end{cases} \tag{6}$$

where b_k is the kth bin representing a direction of $\Delta \hat{c}_t$ (see Fig. 3b). Displacements with magnitudes $\in (\epsilon, 1-\epsilon)$ are encoded into one of six 4-bit strings according to their direction, where the index of b_k is the decimal representation of such 4-bit string; while those displacements with magnitudes outside this range are encoded as 1111 or 0000, as they represent displacements with unstable directions. The binary string, F_p, representing trajectory $\hat{T}_p$ is then the concatenation of the L encoded displacements:

$$F_p = \sum_{0 \leqslant t < L} 2^{4t} B(\Delta \hat{c}_t). \tag{7}$$

The final binary string, E_p, representing video volume v_p is the concatenation of F_p with two BWD strings, G_p^x and G_p^y:

$$E_p = F_p + G_p^x + G_p^y. \tag{8}$$

3.2. Probabilistic Models

We generate three probabilistic models for the extracted features: one dictionary model, one local neighbourhood histogram model and one foreground occupancy model.

Dictionary model: we create independent visual dictionaries of the binary features, E_p. To this end, we define a region $S : x_p - 10 : x_p + 10$ and $y_p - 10 : y_p + 10$ for each FAST point represented by E_p, and cluster all FAST points within S by k-means. Each cluster's centroid, z_i, is:

$$z_i : \arg\min_S \sum_{i=1}^{k} \sum_{E_p \in S} E_p \otimes z_i. \tag{9}$$

Dictionary voting is evaluated via a Gaussian Mixture Model (GMM) with parameters $\theta = \{\pi_k, \mu_k, \sigma_k\}$, representing, respectively, the weight, mean and standard deviation of the k^{th} component:

$$p_{DIC}(d_{E_p} \mid \theta) = \sum_k \pi_k \mathcal{N}(d_{E_p} \mid \mu_k, \sigma_k), \tag{10}$$

where d_{E_p} is the Hamming distance of the word $E_p \in S$ to the closest dictionary centroid. It is important to notice that binary features make this procedure remarkably fast.

Local neighborhood histogram model: We capture local word compositions via histograms. We first compute the frequency of the label l_p, which defines the matching of E_p to the closest centroid within a dictionary. For each local neighborhood S, a histogram H_S is then generated representing the frequency of all matching labels. All histograms generated for all the local neighborhoods are then clustered via k-means, and their distance d_{H_S} to the closest centroid is modeled via a GMM with k components:

$$p_H(d_{H_S} \mid \theta) = \sum_k \pi_k \mathcal{N}(d_{H_S} \mid \mu_k, \sigma_k). \tag{11}$$

Foreground occupancy model: For each region $\{R_1, R_2, R_3\}$ (see Fig. 2a), we generate a GMM with k components to model foreground occupancy features:

$$p_O(O_q \mid \theta) = \sum_k \pi_k \mathcal{N}(O_q \mid \mu_k, \sigma_k). \tag{12}$$

For all GMMs, the best number of components is obtained by iterating over the Akaike Information Criterion [3].

3.3. Inference Mechanism

The inference mechanism is based on the GMM votes (Eq. 10 - 12) given the newly observed features. The models used for the temporal gradients (Eq. 10, 11) and that used for the background (Eq. 12) generate two masks, M_{TG} and M_{BG}, respectively. If the values in these masks are greater than thresholds γ_{TG} and γ_{BG}, respectively, the corresponding regions are deemed to be abnormal:

$$M_{TG} = -\log\left(\prod (p_{DIC})\,(p_H)\right) > \gamma_{TG}, \tag{13}$$

$$M_{BG} = -\log(p_O) > \gamma_{BG}. \tag{14}$$

The corresponding frame is labeled by using a joint criterion:

$$A = M_{TG} \vee M_{BG}. \tag{15}$$

The binary mask A depicts the abnormal regions in the labelled frame. Note that either the temporal gradients or the

Fig. 4. Events detected (in green). 1st row: UCSD Peds1 (frames 1-3) and Peds2 (frames 4-6). 2nd row: LV dataset. The zoomed-in RoI is shown next to the corresponding labeled frame. Events are (left to rigth): a car accident, a motorcycle theft and vandalism outside a store.

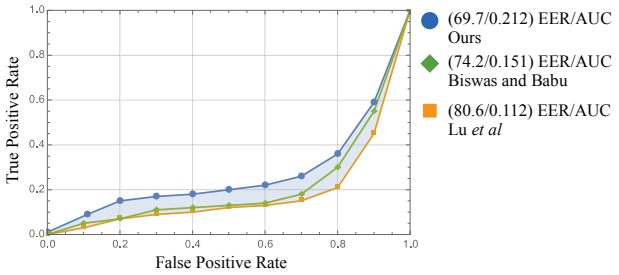


Fig. 5. RoI-level ROC performance for the LV dataset.

Table 1. Equal Error Rate (EER) for UCSD Peds1/Peds2.

Authors	EER Frame level	EER Pixel level	Frame processing time	On-line performance
Bertini *et al.*[13]	31/32	–	125 ms	
Reddy *et al.* [12]	22.5/20	32/–	140 ms	
Hu *et al.* [10]	18/15	36/–	200 ms	
Javan and Levine [2]	15/13	27/26	220 ms	
Cheng *et al.* [6]	19.9	38.8	1100 ms	
Cong *et al.* [4]	23/–	51.2/–	3800 ms	
Lu *et al.* [14]	**15**/22.3	59.1/49.8	**6 ms**	✓
Biswas and Babu [15]	24.66/29.6	50.95/42.3	14 ms	✓
Proposed	25.34/**21.2**	**48.1/38.4**	26 ms	✓

background may trigger the inference mechanism. To improve performance, we set the thresholds in Eq. 13, 14 on a per-region basis, where each region R_n with $n \in \{1, 2, 3\}$ is assigned a pair of thresholds, γ_{TG} and γ_{BG}. This is based on the significance vote variance observed across the regions. We set the thresholds for a region to the lowest votes given by the individual GMM models for that region during training. During testing, we multiply the lowest probability observed during training by a constant < 1 to detect abnormal events.

4. EXPERIMENTS AND DISCUSSIONS

To rank our framework and compare it with existing methods, we first employ the UCSD Peds1 and Peds2 [22] datasets. We compare our framework against some of the best-performing non-online methods [2, 4, 6, 10, 12, 13]; and two state-of-the-art online methods [14, 15]. Results for this dataset are reported in terms of the Equal Error Rate (EER). The EER describes the rate of misclassified frames, i.e., it becomes smaller as the system correctly identifies abnormal frames. If the EER increases, the system is not detecting abnormal frames (False Negatives) or is incorrectly detecting abnormal frames (False Positives). Table 1 shows that our framework achieves competitive performance compared to non-online methods and outperforms online methods, particularly at the pixel level. Our framework detects the non-pedestrian entities (cars, bikes and trollies) accurately, as Fig. 4 shows, first row.

We also rank our framework on the real events of the LV dataset [23], which consist of 28 real surveillance videos capturing a variety of events including robberies, car accidents, and kidnappings. A frame is deemed to be correctly detected as abnormal if at least 20% of the region of interest (RoI) is detected. For this dataset, we compare our framework against the online methods in [14, 15]. The Receiver Operating Characteristic (ROC) curves for these methods are depicted in Fig. 5. We observe that our framework achieves the greatest Area Under the Curve (AUC) and the smallest EER, which demonstrates significantly better detection rates than the other online methods. Our framework is capable of detecting the majority of events with competitive accuracy (see Fig. 4, second row). Note how our framework can detect very small RoIs very accurately thanks to defining video volumes of different size by using a grid of variable-size cells. Even though our framework is slower than the other methods, it achieves better EER/AUC performance while attaining online processing. This aspect reveals our framework's good compromise between detection accuracy and frame processing times.

5. CONCLUSIONS

We have proposed an online framework to detect abnormal events in videos with competitive accuracy and short processing times. Our framework uses binary features to encode temporal gradients, in conjunction with foreground occupancy features, to accurately classify events by relying on low-complexity probabilistic models. Evaluations on the UCSD dataset show that our framework outperforms other online methods, while achieving competitive results compared to non-online methods. Evaluations on the LV dataset, which comprises real surveillance videos, also show that our framework outperforms online methods.

References

[1] T. Huang, "Surveillance video: The biggest big data," *Computing Now*, vol. 7, no. 2, pp. 82–91, 2014.

[2] M. J. Roshtkhari and M. D. Levine, "An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1436 – 1452, 2013.

[3] R. Leyva, V. Sanchez, and C.-T. Li, "Video anomaly detection with compact feature sets for online performance," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3463–3478, July 2017.

[4] Y. Cong, J. Yuan, and Y. Tang, "Video anomaly search in crowded scenes via spatio-temporal motion context," *IEEE Transactions on Information Forensics and Security, 2013*, vol. 8, no. 10, pp. 1590–1599, Oct 2013.

[5] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and gaussian process regression," in *IEEE International Conference on Computer Vision*, June 2015, pp. 2909–2917.

[6] K. Cheng, Y. Chen, and W. Fang, "Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5288–5301, Dec 2015.

[7] X. Zhu, J. Liu, J. Wang, C. Li, and H. Lu, "Sparse representation for robust abnormality detection in crowded scenes," *Pattern Recognition*, vol. 47, no. 5, pp. 1791 – 1799, 2014.

[8] M. Roshtkhari and M. Levine, "Online dominant and anomalous behavior detection in videos," in *IEEE International Conference on Computer Vision*, June 2013, pp. 2611–2618.

[9] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 18–32, Jan 2014.

[10] Y. Hu, Y. Zhang, and L. Davis, "Unsupervised abnormal crowd activity detection using semiparametric scan statistic," in *IEEE International Conference on Computer Vision*, June 2013, pp. 767–774.

[11] M. Bertini, A. Del Bimbo, and L. Seidenari, "Scene and crowd behaviour analysis with local space-time descriptors," in *International Symposium on Communications Control and Signal Processing (ISCCSP), 2012*, May 2012, pp. 1–6.

[12] V. Reddy, C. Sanderson, and B. Lovell, "Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture," in *IEEE International Conference on Computer Vision*, June 2011, pp. 55–61.

[13] M. Bertini, A. D. Bimbo, and L. Seidenari, "Multi-scale and real-time non-parametric approach for anomaly detection and localization," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 320 – 329, 2012, special issue on Semantic Understanding of Human Behaviors in Image Sequences.

[14] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *IEEE International Conference on Computer Vision*, Dec 2013, pp. 2720–2727.

[15] S. Biswas and R. Babu, "Real time anomaly detection in h.264 compressed videos," in *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013*, Dec 2013, pp. 1–4.

[16] R. Leyva, V. Sanchez, and C.-T. Li, "Video anomaly detection based on wake motion descriptors and perspective grids," in *IEEE International Workshop on Information Forensics and Security (WIFS), 2014*, Dec 2014, pp. 209–214.

[17] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *IEEE International Conference on Computer Vision*, vol. 2, Oct 2005, pp. 1508–1515 Vol. 2.

[18] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE International Conference on Computer Vision*, June 2008, pp. 1–8.

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

[20] R. Leyva, V. Sanchez, and C.-T. Li, "A fast binary pair-based video descriptor for action recognition," in *IEEE International Conference on Image Processing*, Sept 2016, pp. 4185–4189.

[21] R. Leyva, V. Sanchez, and C. T. Li, "Fast binary-based video descriptors for action recognition," in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Nov 2016, pp. 1–8.

[22] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *IEEE International Conference on Computer Vision*, June 2010, pp. 1975–1981.

[23] R. Leyva, V. Sanchez, and C.-T. Li, "The lv dataset: a realistic surveillance video dataset for abnormal event detection," in *2017 International Workshop on Biometrics and Forensics*, Mar 2017.