Online Anomaly Detection in Surveillance Videos with Asymptotic Bounds on False Alarm Rate

Keval Doshi, Yasin Yilmaz*

University of South Florida
4202 E Fowler Ave, Tampa, FL 33620

Abstract

Anomaly detection in surveillance videos is attracting an increasing amount of attention. Despite the competitive performance of recent methods, they lack theoretical performance analysis, particularly due to the complex deep neural network architectures used in decision making. Additionally, online decision making is an important but mostly neglected factor in this domain. Much of the existing methods that claim to be online, depend on batch or offline processing in practice. Motivated by these research gaps, we propose an online anomaly detection method in surveillance videos with asymptotic bounds on the false alarm rate, which in turn provides a clear procedure for selecting a proper decision threshold that satisfies the desired false alarm rate. Our proposed algorithm consists of a multi-objective deep learning module along with a statistical anomaly detection module, and its effectiveness is demonstrated on several publicly available data sets where we outperform the state-of-the-art algorithms. All codes are available at `https://github.com/kevaldoshi17/Prediction-based-Video-Anomaly-Detection-`.

Keywords: computer vision; video surveillance; anomaly detection; asymptotic performance analysis; deep learning; online detection

1. Introduction

The rapid advancements in the technology of closed-circuit television (CCTV) cameras and their underlying infrastructural components such as network, storage, and processing hardware have led to a sheer number of surveillance cameras implemented all over the world, and estimated to go beyond 1 billion globally, by the end of the year 2021 [1]. Video surveillance is an essential tool used in law enforcement, transportation, environmental monitoring, etc. mainly for improving security and public safety. For example, it has become an inseparable part of crime deterrence and investigation, traffic violation detection, and traffic

*Corresponding author

management. However, considering the massive amounts of videos generated in real-time, manual video analysis by human operator becomes inefficient, expensive, and nearly impossible, which in turn makes a great demand for automated and intelligent methods for analyzing and retrieving important information from videos, in order to maximize the benefits of CCTV.

One of the most important, challenging and time-critical tasks in automated video surveillance is the detection of abnormal events such as traffic accidents and violations, crimes, and natural disasters. Hence, video anomaly detection has become an important research problem in the recent years. Anomaly detection in general is a vast, crucial, and challenging research topic, which deals with the identification of data instances deviating from nominal patterns. It has a wide range of applications, e.g., in medical health care[2], cyber-security [3], hardware security [4], aviation [5], and spacecraft monitoring [6].

Given the important role that video anomaly detection can play in ensuring safety, security and sometimes prevention of potential catastrophes, one of the main outcomes of a video anomaly detection system is the real-time decision making capability. Events such as traffic accidents, robbery, and fire in remote places require immediate counteractions to be taken in a timely manner, which can be facilitated by the real-time detection of anomalous events. Despite its importance, a very limited body of research has focused on online and real-time detection methods. Moreover, some of the methods that claim to be online heavily depend on batch processing of long video segments. For example, [7] performs a normalization step which requires the entire video.

A vast majority of the recent state-of-the-art video anomaly detection methods depend on complex neural network architectures [8]. Although deep neural networks provide superior performance on various machine learning and computer vision tasks, such as object detection [9], image classification [10], playing games [11], image synthesis[12], etc., where sufficiently large and inclusive data sets are available to train on, there is also a significant debate on their shortcomings in terms of interpretability, analyzability, and reliability of their decisions [13]. For example, [14, 15] propose using a nearest neighbor-based approach together with deep neural network structures to achieve robustness, interpretability for the decisions made by the model, and as defense against adversarial attack. Additionally, to the best of the our knowledge, none of the neural network-based video anomaly detection methods has been analyzed in terms of performance guarantees. On the other hand, statistical and nearest neighbor-based methods remain popular due to their appealing characteristics such as being amenable to performance analysis, computational efficiency, and robustness [16, 17].

Motivated by the aforementioned domain challenges and research gaps, we propose a hybrid use of neural networks and statistical k nearest neighbor (kNN) decision approach for finding anomalies in video in an online fashion. In summary, our contributions in this paper are as follows:

- We propose a novel framework composed of deep learning-based feature extraction from video frames, and a statistical sequential anomaly detection algorithm.

- We derive an asymptotic bound on the false alarm rate of our detection algorithm, and propose a technique for selecting a proper threshold which satisfies the desired false alarm rate.
- We extensively evaluate our proposed framework on publicly available video anomaly detection data sets.

The remainder of the paper is organized as: Related Work (Section 2), Proposed Method (Section 3), Experiments (Section 4), and Conclusion (Section 5).

2. Related Work

Semi-supervised detection of anomalies in videos, also known as outlier detection, is a commonly adopted learning technique due to the inherent limitations in availability of annotated and anomalous instances. This category of learning methods deals with learning a notion of normality from nominal training videos, and attempts to detect deviations from the learned normality notion. [18, 19]. There are also several supervised detection methods, which train on both nominal and anomalous videos. The main drawback of such methods is the difficulty in finding frame-level labeled, representative, and inclusive anomaly instances. To this end, [8] proposes using a deep multiple instance learning (MIL) approach to train on video-level annotated videos, in a weakly supervised manner. Although training on anomalous videos would enhance the detection capability on similar anomaly events, supervised methods typically suffer from unknown and novel anomaly types.

One of the key components of the video anomaly detection algorithms is the extraction of meaningful features, which can capture the difference between the nominal and anomalous events within the video. The selection of feature types has a significant impact on the identifiability of types of anomalous events in the video sequences. Many early video anomaly detection techniques and some recent ones focused on the trajectory features [20], which limits their applicability to the detection of the anomalies related to the trajectory patterns, and moving objects. For instance, [21] studied detection of abnormal vehicle trajectories such as illegal U-turn. [22] extracts human skeleton trajectory patterns, and hence is limited to only the detection of abnormalities in human behavior.

Motion and appearance features are another class of widely used features in this domain. [23] extracts motion direction and magnitudes, to detect spatio-temporal anomalies. Histogram of optical flow [24, 25], and histogram of oriented gradients [26] are some other commonly used hand-crafted feature extraction techniques used in the literature. Sparse coding based methods [27] are also applied in detection of video anomalies. They learn a dictionary of normal sparse events, and attempt to detect anomalies based on the reconstructability of video from the dictionary atoms. [28] uses sparse reconstruction to learn joint trajectory representations of multiple objects.

In contrary to the hand-crafted feature extraction, are the neural network based feature learning methods. [29] learns the appearance and motion features

by deep neural networks. [30] utilizes Convolutional Neural Networks (CNN), and Convolutional Long Short Term Memory (CLSTM) to learn appearance and motion features, respectively. Neural network based approaches have been recently dominating the literature. For example, [31] trains Generative Adversarial Network (GAN) on normal video frames, to generate internal scene representations (appearance and motion), based on a given frame and its optical flow, and detects deviation of the GAN output from the normal data, by AlexNet [10]. [32] trains a GAN-like adversarial network, in which a reconstruction component learns to reconstruct the normal test frames, and attempts to train a discriminator by gradually injecting anomalies to it, while concurrently the discriminator (detector) learns to detect the anomalies injected by the reconstructor. In [33, 34], a transfer learning based approach is used for continual learning for anomaly detection in surveillance videos from a few samples.

3. Proposed Method

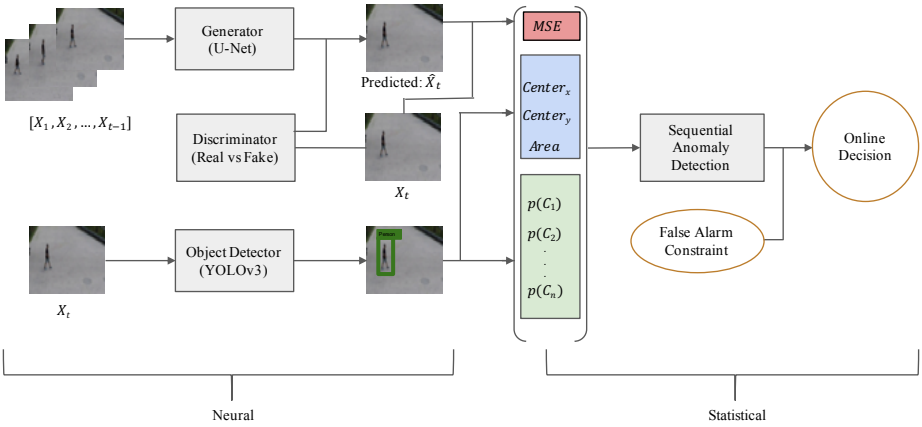

Figure 1: Proposed MONAD framework. At each time t, neural network-based feature extraction module provides motion (MSE), location (center coordinates and area of bounding box), and appearance (class probabilities) features to the statistical anomaly detection module, which automatically sets its decision threshold to satisfy a false alarm constraint and makes online decisions.

3.1. Motivation

Anomaly detection in surveillance videos is defined as the identification of unusual events which do not conform to the expectation. We base our study on two important requirements that a successful video anomaly detector should satisfy: (i) extract meaningful features which can be utilized to distinguish nominal and anomalous data; and (ii) provide a decision making strategy which can be easily tuned to satisfy a given false alarm rate. While existing works partially fulfills the first requirement by defining various constraints on spatial

and temporal video features, they typically neglect providing an analytical and amenable decision strategy. Motivated by this shortcoming, we propose a unified framework called Multi-Objective Neural Anomaly Detector (MONAD[1]). Like *monads* provide a unified functional model for programming, our proposed MONAD unifies deep learning-based feature extraction and analytical anomaly detection by incorporating two modules, as shown in Figure 1. The first module consists of a Generative Adversarial Network (GAN) based future frame predictor and a lightweight object detector (YOLOv3) to extract meaningful features. The second module consists of a nonparametric statistical algorithm which uses the extracted features for online anomaly detection. To the best of our knowledge, this is the first work to present theoretical performance analysis for a deep learning-based video anomaly detection method. Our MONAD framework is described in detail in the following sections.

3.2. Feature Selection

Most existing works focus on a certain aspect of the video such as optical flow, gradient loss or intensity loss. This in turn restrains the existing algorithms to a certain form of anomalous event which is manifested in the considered video aspect. However, in general, the type of anomaly is broad and unknown while training the algorithm. For example, an anomalous event can be justified on the basis of appearance (a person carrying a gun), motion (two people fighting) or location (a person walking on the roadway). To account for all such cases, we create a feature vector F_t^i for each object i in frame X_t at time t, where F_t^i is given by $[w_1 F_{motion}, w_2 F_{location}, w_3 F_{appearance}]$. The weights w_1, w_2, w_3 are used to adjust the relative importance of each feature category.

3.3. Frame Prediction

A heuristic approach for detecting anomalies in videos is by predicting the future video frame $\widehat{X}_t$ using previous video frames $\{X_1, X_2, \ldots, X_{t-1}\}$, and then comparing it to X_t through mean squared error (MSE). Instead of deciding directly on MSE, we use MSE of video frame prediction to obtain motion features (Section 3.5). GANs are known to be successful in generating realistic images and videos. However, regular GANs might face the vanishing gradient problem during learning as they hypothesize the discriminator as a classifier with the sigmoid cross entropy loss function. To overcome this problem, we use a modified version of GAN called Least Square GAN (LS-GAN) [35]. The GAN architecture comprises of a generator network G and a discriminator network D, where the function of G is to generate frames that would be difficult-to-classify by D. Ideally, once G is well trained, D cannot predict better than chance. Similar to [7], we employ a U-Net [36] based network for G and a patch discriminator for D.

[1] *Monad* is a philosophical term for infinitesimal unit, and also a functional programming term for minimal structure.

For training the generator G, we follow [7], and combine the constraints on intensity, gradient difference, optical flow, and adversarial training to get the following objective function

$$\begin{aligned} L_G = \gamma_{int} L_{int}(\widehat{X}, X) + \gamma_{gd} L_{gd}(\widehat{X}, X) + \\ \gamma_{of} L_{of}(\widehat{X}, X) + \gamma_{adv} L_{adv}(\widehat{X}, X) \end{aligned} \tag{1}$$

where $\gamma_{int}, \gamma_{gd}, \gamma_{of}, \gamma_{adv} \geq 0$ are the corresponding weights for the losses.

Intensity loss is the l_1 or l_2 distance between the predicted frame $\widehat{X}$ and the actual frame X, which is used to maintain similarity between pixels in the RGB space, and given by

$$L_{int}(\widehat{X}, X) = \left\| \widehat{X} - X \right\|^2 . \tag{2}$$

Gradient difference loss is used to sharpen the image prediction and is given by

$$\begin{aligned} L_{gd}(\widehat{X}, X) = \sum_{i,j} \left\| |\widehat{X}_{i,j} - \widehat{X}_{i-1,j}| - |X_{i,j} - X_{i-1,j}| \right\|_1 \\ + \left\| |\widehat{X}_{i,j} - \widehat{X}_{i,j-1}| - |X_{i,j} - X_{i,j-1}| \right\|_1 \end{aligned} \tag{3}$$

where (i, j) denotes the spatial index of a video frame.

Optical flow loss is used to improve the coherence of motion in the predicted frame, and is given by

$$L_{of}(\widehat{X}_{t+1}, X_{t+1}, X_t) = \left\| f(\widehat{X}_{t+1}, X_t) - f(X_{t+1}, X_t) \right\|_1 \tag{4}$$

where f is a pretrained CNN-based function called Flownet, and is used to estimate the optical flow.

Adversarial generator loss is minimized to confuse D as much as possible such that it cannot discriminate the generated predictions, and is given by

$$L_{adv}(\widehat{X}) = \sum_{i,j} \frac{1}{2} L_{MSE}(D(\widehat{X}_{i,j}), 1) \tag{5}$$

where $D(\widehat{X}_{i,j}) = 1$ denotes "real" decision by D for patch (i, j), $D(\widehat{X}_{i,j}) = 0$ denotes "fake" decision, and L_{MSE} is the mean squared error function.

3.4. Object Detection

We propose to detect objects using a real-time object detection system such as You Only Look Once (YOLO) [37] to obtain location and appearance features (Section 3.5). The advantage of YOLO is that it is capable of processing higher frames per second on a GPU while providing the same or even better accuracy as compared to the other state-of-the-art models such as SSD and ResNet. Speed is a critical factor for online anomaly detection, so we currently prefer YOLOv3 in

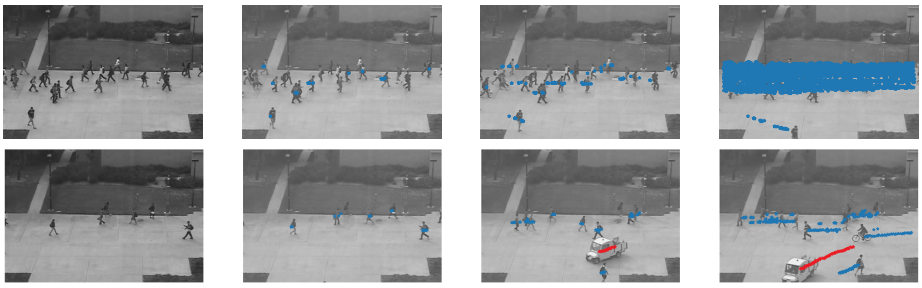

Figure 2: Example video frames from the UCSD Ped2 dataset showing the extraction of bounding box center (location) feature in nominal training data (top row) and test data (bottom row). Columns from left to right correspond to the first, 30th, 150th, and the last frame in all training videos (top row), and in a test video (bottom row). In the test video, the unusual path of golf cart, shown with red dots, together with the class probability and high prediction error (MSE) due to unusual speed of cart statistically contribute to the anomaly decision. Best viewed in color.

our implementations. For each detected object in image X_t, we get a bounding box (location) along with the class probabilities (appearance). As shown in Fig. 2, we monitor the center of the bounding boxes to track paths different objects might take in the training videos. Instead of simply using the entire bounding box, we monitor the center of the box and its area to obtain location features. This not only reduces the complexity, but also effectively avoids false positives in case the bounding box is not tight. In a testing video, objects diverging from the nominal paths and class probabilities will help us detect anomalies, as explained in Section 3.6.

3.5. Feature Vector

Finally, for each object i detected in a frame, we construct a feature vector as:

$$F_t^i = \begin{bmatrix} w_1 MSE(X_t, \widehat{X}_t) \\ w_2 Center_x \\ w_2 Center_y \\ w_2 Area \\ w_3 p(C_1) \\ w_3 p(C_2) \\ \vdots \\ w_3 p(C_n) \end{bmatrix}, \tag{6}$$

where $MSE(X_t, \widehat{X}_t)$ is the prediction error from the GAN-based frame predictor (Section 3.3); $Center_x, Center_y, Area$ denote the coordinates of the center of the bounding box and the area of the bounding box (Section 3.4); and $p(C_1), \ldots, p(C_n)$ are the class probabilities for the detected object (Section 3.4). Hence, at any given time t, with n denoting the number of possible classes, the dimensionality of F_t^i is given by $m = n + 4$.

3.6. Anomaly Detection

Our goal here is to detect anomalies in streaming videos with minimal detection delays while satisfying a desired false alarm rate. We can safely hypothesize that any anomalous event would persist for an unknown period of time. This makes the problem suitable for a sequential anomaly detection framework [38]. However, since we have no prior knowledge about the anomalous event that might occur in a video, parametric algorithms which require probabilistic model and data for both nominal and anomaly cannot be used directly. Next, we explain the training and testing of our proposed nonparametric sequential anomaly detection algorithm.

Training: First, given a set of N training videos $\mathcal{V} \triangleq \{v_i : i = 1, 2, \ldots, N\}$ consisting of P frames in total, we leverage the deep learning module of our proposed detector to extract M feature vectors $\mathcal{F}^M = \{F^i\}$ for M detected objects in total such that $M \geq P$. We assume that the training data does not include any anomalies. These M vectors correspond to M points in the nominal data space, distributed according to an unknown complex probability distribution. Following a data-driven approach we would like to learn a nonparametric description of the nominal data distribution. Due to its attractive traits, such as analyzability, interpretability, and computational efficiency [16, 17], we use k nearest neighbor (kNN) distance, which captures the local interactions between nominal data points, to figure out a nominal data pattern. Given the informativeness of extracted motion, location, and appearance features, anomalous instances are expected to lie further away from the nominal manifold defined by $\mathcal{F}^M$. Consequently, the kNN distance of anomalous instances with respect to the nominal data points in $\mathcal{F}^M$ will be statistically higher as compared to the nominal data points. The training procedure of our detector is given as follows:

1. Randomly partition the nominal dataset $\mathcal{F}^M$ into two sets $\mathcal{F}^{M_1}$ and $\mathcal{F}^{M_2}$ such that $M = M_1 + M_2$.
2. Then for each point F_i in $\mathcal{F}^{M_1}$, we compute the kNN distance d_i with respect to the points in set $\mathcal{F}^{M_2}$.
3. For a significance level α, e.g., 0.05, the $(1-\alpha)$th percentile d_α of kNN distances $\{d_1, \ldots, d_{M_1}\}$ is used as a baseline statistic for computing the anomaly evidence of test instances.
4. The maximum value of kNN distances $\{d_1, \ldots, d_{M_1}\}$ is used as an upper bound (ϕ) for δ_t, given by Eq. (7), which is then used for selecting a threshold h, as explained in Section 3.7.

Testing: During the testing phase, for each object i detected at time t, the deep learning module constructs the feature vector F_t^i and computes the kNN (Euclidean) distance d_t^i with respect to the training instances in $\mathcal{F}^{M_2}$. The proposed sequential anomaly detection system then computes the instantaneous frame-level anomaly evidence δ_t:

$$\delta_t = (\max_i \{d_t^i\})^m - d_\alpha^m, \tag{7}$$

where m is the dimensionality of feature vector F_t^i. Finally, following a CUSUM-like procedure [38] we update the running decision statistic s_t as

$$s_t = \max\{s_{t-1} + \delta_t, 0\}, s_0 = 0. \tag{8}$$

For nominal data, δ_t typically gets negative values, hence the decision statistic s_t hovers around zero; whereas for anomalous data δ_t is expected to take positive values, and successive positive values of δ_t will make s_t grow. We decide that a video frame is anomalous if the decision statistic s_t exceeds the threshold h. After s_t exceeds h, we perform some fine tuning to better label video frames as nominal or anomalous. Specifically, we find the frame s_t started to grow, i.e., the last time $s_t = 0$ before detection, say τ_{start}. Then, we also determine the frame s_t stops increasing and keeps decreasing for n, e.g., 5, consecutive frames, say τ_{end}. Finally, we label the frames between τ_{start} and τ_{end} as anomalous, and continue testing for new anomalies with frame $\tau_{end} + 1$ by resetting $s_{\tau_{end}} = 0$.

3.7. Threshold Selection

If the test statistic crosses the threshold when there is no anomaly, this event is called a false alarm. Existing works consider the decision threshold as a design parameter, and do not provide any analytical procedure for choosing its value. For an anomaly detection algorithm to be implemented in a practical setting, a clear procedure is necessary for selecting the decision threshold such that it satisfies a desired false alarm rate. The reliability of an algorithm in terms of false alarm rate is crucial for minimizing human involvement. To provide such a performance guarantee for the false alarm rate, we derive an asymptotic upper bound on the average false alarm rate of the proposed algorithm.

Theorem 1. *The false alarm rate of the proposed algorithm is asymptotically (as $M_2 \to \infty$) upper bounded by*

$$FAR \leq e^{-\omega_0 h}, \tag{9}$$

where h is the decision threshold, and $\omega_0 > 0$ is given by

$$\omega_0 = v_m - \theta - \frac{1}{\phi}\mathcal{W}\left(-\phi\theta e^{-\phi\theta}\right), \tag{10}$$
$$\theta = \frac{v_m}{e^{v_m d_\alpha^m}}.$$

In (10), $\mathcal{W}(\cdot)$ is the Lambert-W function, $v_m = \frac{\pi^{m/2}}{\Gamma(m/2+1)}$ is the constant for the m-dimensional Lebesgue measure (i.e., $v_m d_\alpha^m$ is the m-dimensional volume of the hyperball with radius d_α), and ϕ is the upper bound for δ_t.

Proof. See Appendix.

Although the expression for ω_0 looks complicated, all the terms in (10) can be easily computed. Particularly, v_m is directly given by the dimensionality m, d_α comes from the training phase, ϕ is also found in training, and finally there is a

built-in Lambert-W function in popular programming languages such as Python and Matlab. Hence, given the training data, ω_0 can be easily computed, and based on Theorem 1, the threshold h can be chosen to asymptotically achieve the desired false alarm period as follows

$$h = \frac{-\log(FAR)}{\omega_0}. \tag{11}$$

4. Experiments

4.1. Datasets

We evaluate our proposed method on three publicly available video anomaly data sets, namely the CUHK avenue dataset [39], the UCSD pedestrian dataset [40], and the ShanghaiTech [41] campus dataset. Each data set presents its own set of challenges and unique characteristics such as types of anomaly, video quality, background location, etc. Hence, we treat each dataset independently and present individual results for each of them. Here, we briefly introduce each dataset that are used in our experiments.

UCSD: The UCSD pedestrian data set is composed of two parts, namely Ped1 and Ped2. Following the work of [19, 42], we exclude Ped1 from our experiments due to its significantly lower resolution of 158 x 238 and a lack of consistency in the reported results as some recent works reported their performance only on a subset of the entire data set. Hence, we present our results on the UCSD Ped2 dataset which consists of 16 training and 12 test videos, each with a resolution of 240 x 360. All the anomalous events are caused due to vehicles such as bicycles, skateboarders and wheelchairs crossing pedestrian areas.

Avenue: The CUHK avenue dataset consists of 16 training and 21 test videos with a frame resolution of 360 x 640. The anomalous behaviour is represented by people throwing objects, loitering and running.

ShanghaiTech: The ShanghaiTech Campus dataset is one of the largest and most challenging datasets available for anomaly detection in videos. It consists of 330 training and 107 test videos from 13 different scenes, which sets it apart from the other available datasets. The resolution for each video frame is 480 x 856.

4.2. Comparison with Existing Methods

We compare our proposed algorithm in Table 1 with state-of-the-art deep learning-based methods, as well as methods based on hand-crafted features: MPPCA [43], MPPC + SFA [40], Del et al. [44], Conv-AE [45], ConvLSTM-AE [30], Growing Gas [46], Stacked RNN [41], Deep Generic [42], GANs [47], Liu et al. [7]. A popular metric used for comparison in anomaly detection literature is the Area under the Receiver Operating Characteristic (AuROC) curve. Higher AuROC values indicate better performance for an anomaly detection system. For performance evaluation, following the existing works [48, 19, 7], we consider frame level AuROC.

4.3. Implementation Details

In the prediction pipeline, the U-NET based generator and the patch discriminator are implemented in Tensorflow. Each frame is resized to 256 x 256 and normalized to [-1,1]. The window size t is set to 4. Similar to [7], we use the Adam optimizer for training and set the learning rate to 0.0001 and 0.00001 for the generator and discriminator, respectively. The object detector used is YOLOv3 which is based on the Darknet architecture and is pretrained on the MS-COCO dataset. During training, we extract the bounds which have a confidence level greater than 0.6, and for testing we consider confidence levels greater than or equal to 0.4. The weights w_1, w_2 and w_3 are set to 1, 0.4 and 0.9 respectively. The sequential anomaly detection algorithm is implemented in Python.

4.4. Impact of Sequential Anomaly Detection

To demonstrate the importance of sequential anomaly detection in videos, we implement a nonsequential version of our algorithm by applying a threshold to the instantaneous anomaly evidence δ_t, given in (7), which is similar to the approach employed by many recent works [7, 8, 19]. As Figure 3 shows, instantaneous anomaly evidence is more prone to false alarms than the sequential MONAD statistic since it only considers the noisy evidence available at the current time to decide. Whereas, the proposed sequential statistic handles noisy evidence by integrating recent evidence over time.

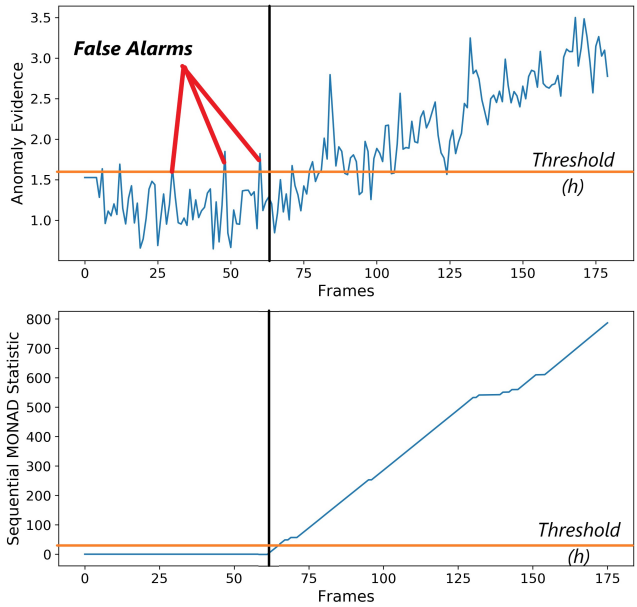

Figure 3: The advantage of sequential anomaly detection over single-shot detection in terms of controlling false alarms.

Methodology	CUHK Avenue	UCSD Ped 2	ShanghaiTech
MPPCA [43]	-	69.3	-
MPPC + SFA [40]	-	61.3	-
Del et al. [44]	78.3	-	-
Conv-AE [45]	80.0	85.0	60.9
ConvLSTM-AE [30]	77.0	88.1	-
Growing Gas [46]	-	93.5	-
Stacked RNN [41]	81.7	92.2	68.0
Deep Generic [42]	-	92.2	-
GANs [31]	-	88.4	-
Liu et al. [7]	85.1	95.4	**72.8**
Ours	**86.4**	**97.2**	70.9

Table 1: AuROC result comparison on three datasets.

4.5. Results

We compare our results to a wide range of methods in Table 1. Recently, [19] showed significant gains over the rest of the methods. However, their methodology of computing the AuROC gives them an unfair advantage as they calculate the AuROC for each video in a dataset, and then average them as the AuROC of the dataset, as opposed to the other works which concatenate all the videos first and then determine the AuROC as the dataset's score.

As shown in Table 1 we are able to outperform the existing results in the avenue and UCSD dataset, and achieve competitive performance in the ShanghaiTech dataset. We should note here that our reported result in the ShanghaiTech dataset is based on online decision making without seeing future video frames. A common technique used by several recent works [7, 19] is to normalize the computed statistic for each test video independently, including the ShanghaiTech dataset. However, this methodology cannot be implemented in an online (real-time) system as it requires prior knowledge about the minimum and maximum values the statistic might take.

Hence, we also compare our online method with the online version of state-of-the-art method [7]. In that version, the minimum and maximum values of decision statistic is obtained from the training data and used for all videos in the test data to normalize the decision statistic, instead of the minimum and maximum values in each test video separately. AuROC value, which is the most common performance metric in the literature, considers the entire range $(0, 1)$ of false alarm rates. However, in practice, false alarm rate must satisfy an acceptable level (e.g., up to 10%). In Figure 4, on the UCSD and ShanghaiTech data sets, we compare our algorithm with the online version of [7] within a practical range of false alarm in terms of the ROC curve (true positive rate vs. false positive rate). As clearly seen in the figures, the proposed MONAD algorithm achieves much higher true alarm rates than [7] in both datasets while satisfying practical false alarm rates.

Finally, in Figure 5, we analyze the bound for false alarm rate derived in Theorem 1. For the clarity of visualization, the figure shows the logarithm of false alarm period, which is the inverse of the false alarm rate. In this case,

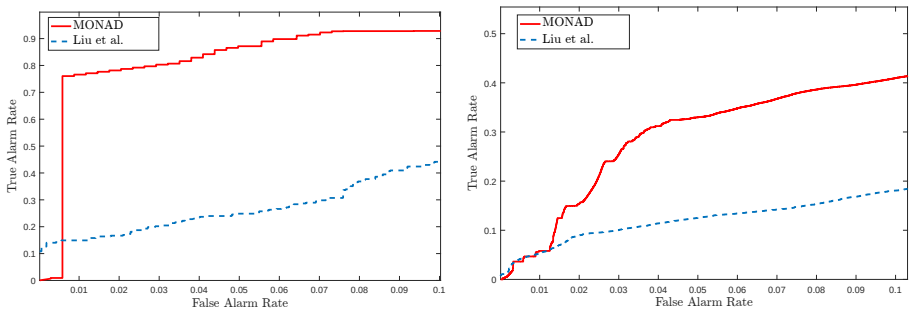

Figure 4: The ROC curves of the proposed MONAD algorithm and the online version of Liu et al. [7] for a practical range of false alarm rate in the UCSD Ped 2 (left) and ShanghaiTech (right) data sets.

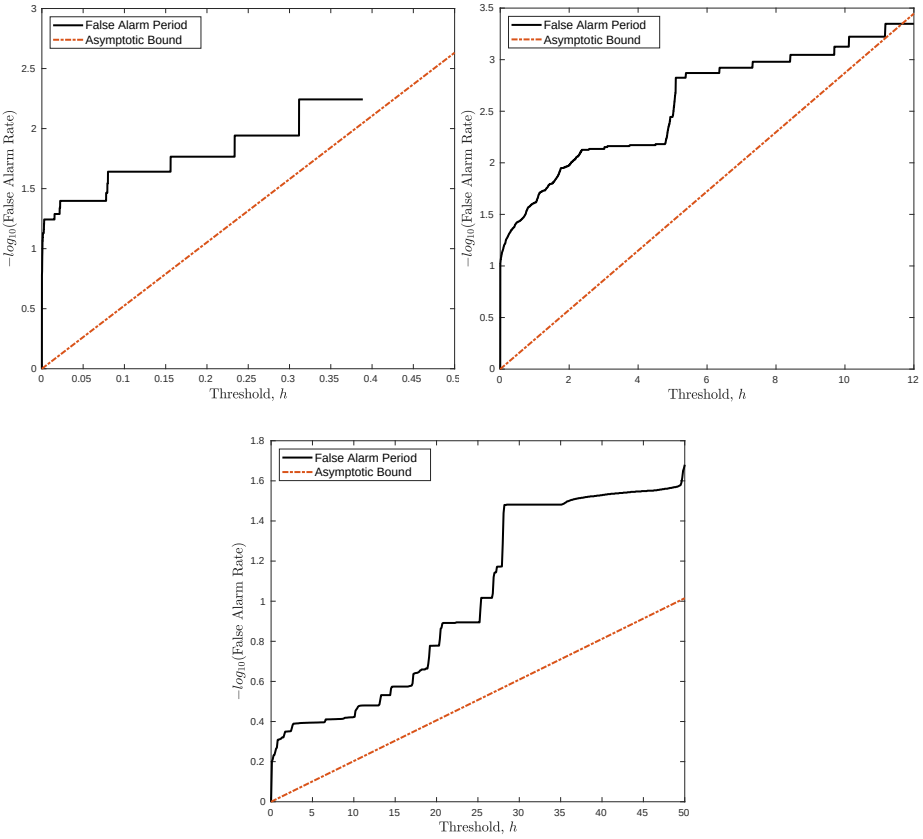

Figure 5: Actual false alarm periods vs. derived lower bounds for the UCSD Ped.2 (top left), ShanghaiTech (top right), and Avenue (bottom) data sets.

the upper bound on false alarm rate becomes a lower bound on the false alarm period. The experimental results corroborate the theoretical bound and the procedure presented in Section 3.7 for obtaining the decision threshold h.

4.6. Computational Complexity

In this section we analyze the computational complexity of the sequential anomaly detection module, as well as the average running time of the deep learning module.

Sequential Anomaly Detection: The training phase of the proposed anomaly detection algorithm requires computation of kNN distances for each point in $\mathcal{F}^{M_1}$ to each point in $\mathcal{F}^{M_2}$. Therefore, the time complexity of training phase is given by $\mathcal{O}(M_1 M_2 m)$. The space complexity of the training phase is $\mathcal{O}(M_2 m)$ since M_2 data instances need to be saved for the testing phase. In the testing phase, since we compute the kNN distances of a single point to all data points in $\mathcal{F}^{M_2}$, the time complexity is $\mathcal{O}(M_2 m)$.

Deep Learning Module: The average running time for the GAN-based video frame prediction is 22 frames per second. The YOLO object detector requires about 12 milliseconds to process a single image. This translates to about 83.33 frames per second. The running time can be further improved by using a faster object detector such as YOLOv3-Tiny or a better GPU system. All tests are performed on NVIDIA GeForce RTX 2070 with 8 GB RAM and Intel i7-8700k CPU.

5. Conclusion

For video anomaly detection, we presented an online algorithm, called MONAD, which consists of a deep learning-based feature extraction module and a statistical decision making module. The first module is a novel feature extraction technique that combines GAN-based frame prediction and a lightweight object detector. The second module is a sequential anomaly detector, which enables performance analysis. The asymptotic false alarm rate of MONAD is analyzed, and a practical procedure is provided for selecting its detection threshold to satisfy a desired false alarm rate. Through real data experiments, MONAD is shown to outperform the state-of-the-art methods, and yield false alarm rates consistent with the derived asymptotic bounds. For future work, we plan to focus on the importance of timely detection in video [49] by proposing a new metric based on the average delay and precision.

6. Acknowledgements

This research is funded in part by the Florida Center for Cybersecurity and in part by the U.S. National Science Foundation under the grant #2029875.

Appendix A. Proof of Theorem 1

In [38][page 177], for CUSUM-like algorithms with independent increments, such as MONAD with independent δ_t, a lower bound on the average false alarm period is given as follows

$$E_\infty[T] \geq e^{\omega_0 h},$$

where h is the detection threshold, and $\omega_0 \geq 0$ is the solution to $E[e^{\omega_0 \delta_t}] = 1$.

To analyze the false alarm period, we need to consider the nominal case. In that case, since there is no anomalous object at each time t, the selection of object with maximum kNN distance in $\delta_t = (\max_i\{d_t^i\})^m - d_\alpha^m$ does not necessarily depend on the previous selections due to lack of an anomaly which could correlate the selections. Hence, in the nominal case, it is safe to assume that δ_t is independent over time.

We firstly derive the asymptotic distribution of the frame-level anomaly evidence δ_t in the absence of anomalies. Its cumulative distribution function is given by

$$P(\delta_t \leq y) = P((\max_i\{d_t^i\})^m \leq d_\alpha^m + y).$$

It is sufficient to find the probability distribution of $(\max_i\{d_t^i\})^m$, the mth power of the maximum kNN distance among objects detected at time t. As discussed above, choosing the object with maximum distance in the absence of anomaly yields independent m-dimensional instances $\{F_t\}$ over time, which form a Poisson point process. The nearest neighbor ($k = 1$) distribution for a Poisson point process is given by

$$P(\max_i\{d_t^i\} \leq r) = 1 - \exp(-\Lambda(b(F_t, r)))$$

where $\Lambda(b(F_t, r))$ is the arrival intensity (i.e., Poisson rate measure) in the m-dimensional hypersphere $b(F_t, r)$ centered at F_t with radius r [50]. Asymptotically, for a large number of training instances as $M_2 \to \infty$, under the null (nominal) hypothesis, the nearest neighbor distance $\max_i\{d_t^i\}$ of F_t takes small values, defining an infinitesimal hyperball with homogeneous intensity $\lambda = 1$ around F_t. Since for a homogeneous Poisson process the intensity is written as $\Lambda(b(F_t, r)) = \lambda|b(F_t, r)|$ [50], where $|b(F_t, r)| = \frac{\pi^{m/2}}{\Gamma(m/2+1)} r^m = v_m r^m$ is the Lebesgue measure (i.e., m-dimensional volume) of the hyperball $b(F_t, r)$, we rewrite the nearest neighbor distribution as

$$P(\max_i\{d_t^i\} \leq r) = 1 - \exp\left(-v_m r^m\right),$$

where $v_m = \frac{\pi^{m/2}}{\Gamma(m/2+1)}$ is the constant for the m-dimensional Lebesgue measure.

Now, applying a change of variables we can write the probability density of $(\max_i\{d_t^i\})^m$ and δ_t as

$$f_{(\max_i\{d_t^i\})^m}(y) = \frac{\partial}{\partial y}\left[1 - \exp\left(-v_m y\right)\right], \tag{A.1}$$

$$= v_m \exp(-v_m y), \tag{A.2}$$

$$f_{\delta_t}(y) = v_m \exp(-v_m d_\alpha^m) \exp(-v_m y) \tag{A.3}$$

Using the probability density derived in (A.3), $E[e^{\omega_0 \delta_t}] = 1$ can be written as

$$1 = \int_{-d_\alpha^m}^{\phi} e^{\omega_0 y} v_m e^{-v_m d_\alpha^m} e^{-v_m y} dy, \tag{A.4}$$

$$\frac{e^{v_m d_\alpha^m}}{v_m} = \int_{-d_\alpha^m}^{\phi} e^{(\omega_0 - v_m) y} dy, \tag{A.5}$$

$$= \frac{e^{(\omega_0 - v_m) y}}{\omega_0 - v_m} \Bigg|_{-d_\alpha^m}^{\phi}, \tag{A.6}$$

$$= \frac{e^{(\omega_0 - v_m)\phi} - e^{(\omega_0 - v_m)(-d_\alpha^m)}}{\omega_0 - v_m}, \tag{A.7}$$

where $-d_\alpha^m$ and ϕ are the lower and upper bounds for $\delta_t = (\max_i\{d_t^i\})^m - d_\alpha^m$. The upper bound ϕ is obtained from the training set.

As $M_2 \to \infty$, since the mth power of $(1-\alpha)$th percentile of nearest neighbor distances in training set goes to zero, i.e., $d_\alpha^m \to 0$, we have

$$e^{(\omega_0 - v_m)\phi} = \frac{e^{v_m d_\alpha^m}}{v_m}(\omega_0 - v_m) + 1. \tag{A.8}$$

We next rearrange the terms to obtain the form of $e^{\phi x} = a_0(x + \theta)$ where $x = \omega_0 - v_m$, $a_0 = \frac{e^{v_m d_\alpha^m}}{v_m}$, and $\theta = \frac{v_m}{e^{v_m d_\alpha^m}}$. The solution for x is given by the Lambert-W function [51] as $x = -\theta - \frac{1}{\phi}\mathcal{W}(-\phi e^{-\phi\theta}/a_0)$, hence

$$\omega_0 = v_m - \theta - \frac{1}{\phi}\mathcal{W}\left(-\phi\theta e^{-\phi\theta}\right). \tag{A.9}$$

Finally, since the false alarm rate (i.e., frequency) is the inverse of false alarm period $E_\infty[T]$, we have

$$FAR \leq e^{-\omega_0 h},$$

where h is the detection threshold, and ω_0 is given above.

References

[1] L. Lin, N. Purnell, A world with a billion cameras watching you is just around the corner, The Wall Street Journal, https://www.wsj.com/articles/a-billion-surveillance-cameras-forecast-to-be-watching-within-two-years-11575565402.

[2] H. Zhang, J. Liu, N. Kato, Threshold tuning-based wearable sensor fault detection for reliable medical monitoring using bayesian network model, IEEE Systems Journal 12 (2) (2018) 1886–1896.

[3] Y. Xiang, K. Li, W. Zhou, Low-rate ddos attacks detection and traceback by using new information metrics, IEEE transactions on information forensics and security 6 (2) (2011) 426–437.

[4] R. Elnaggar, K. Chakrabarty, M. B. Tahoori, Hardware trojan detection using changepoint-based anomaly detection techniques, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27 (12) (2019) 2706–2719.

[5] B. Matthews, Automatic Anomaly Detection with Machine Learning, `https://ntrs.nasa.gov/citations/20190030491` (2019).

[6] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 387–395.

[7] W. Liu, W. Luo, D. Lian, S. Gao, Future frame prediction for anomaly detection–a new baseline, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6536–6545.

[8] W. Sultani, C. Chen, M. Shah, Real-world anomaly detection in surveillance videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6479–6488.

[9] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Advances in neural information processing systems, 2016, pp. 379–387.

[10] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[11] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of go without human knowledge, Nature 550 (7676) (2017) 354–359.

[12] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text to image synthesis, arXiv preprint arXiv:1605.05396.

[13] H. Jiang, B. Kim, M. Guan, M. Gupta, To trust or not to trust a classifier, in: Advances in neural information processing systems, 2018, pp. 5541–5552.

[14] N. Papernot, P. McDaniel, Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning, arXiv preprint arXiv:1803.04765.

[15] C. Sitawarin, D. Wagner, Defending against adversarial examples with k-nearest neighbor, arXiv preprint arXiv:1906.09525.

[16] G. H. Chen, D. Shah, et al., Explaining the success of nearest neighbor methods in prediction, Foundations and Trends® in Machine Learning 10 (5-6) (2018) 337–588.

[17] X. Gu, L. Akoglu, A. Rinaldo, Statistical analysis of nearest neighbor methods for anomaly detection, in: Advances in Neural Information Processing Systems, 2019, pp. 10921–10931.

[18] K.-W. Cheng, Y.-T. Chen, W.-H. Fang, Video anomaly detection and localization using hierarchical feature representation and gaussian process regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2909–2917.

[19] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, L. Shao, Object-centric auto-encoders and dummy anomalies for abnormal event detection in video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7842–7851.

[20] N. Anjum, A. Cavallaro, Multifeature object trajectory clustering for video analysis, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1555–1564.

[21] Z. Fu, W. Hu, T. Tan, Similarity based vehicle trajectory clustering and anomaly detection, in: IEEE International Conference on Image Processing 2005, Vol. 2, IEEE, 2005, pp. II–602.

[22] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, S. Venkatesh, Learning regularity in skeleton trajectories for anomaly detection in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11996–12004.

[23] V. Saligrama, Z. Chen, Video anomaly detection based on local statistical aggregates, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2112–2119.

[24] R. Chaudhry, A. Ravichandran, G. Hager, R. Vidal, Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1932–1939.

[25] R. V. H. M. Colque, C. Caetano, M. T. L. de Andrade, W. R. Schwartz, Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos, IEEE Transactions on Circuits and Systems for Video Technology 27 (3) (2016) 673–682.

[26] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 1, IEEE, 2005, pp. 886–893.

[27] B. Zhao, L. Fei-Fei, E. P. Xing, Online detection of unusual events in videos via dynamic sparse coding, in: CVPR 2011, IEEE, 2011, pp. 3313–3320.

[28] X. Mo, V. Monga, R. Bala, Z. Fan, Adaptive sparse representations for video anomaly detection, IEEE Transactions on Circuits and Systems for Video Technology 24 (4) (2013) 631–645.

[29] D. Xu, E. Ricci, Y. Yan, J. Song, N. Sebe, Learning deep representations of appearance and motion for anomalous event detection, arXiv preprint arXiv:1510.01553.

[30] W. Luo, W. Liu, S. Gao, Remembering history with convolutional lstm for anomaly detection, in: 2017 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2017, pp. 439–444.

[31] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, N. Sebe, Abnormal event detection in videos using generative adversarial nets, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017, pp. 1577–1581.

[32] M. Sabokrou, M. Khalooei, M. Fathy, E. Adeli, Adversarially learned one-class classifier for novelty detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3379–3388.

[33] K. Doshi, Y. Yilmaz, Continual learning for anomaly detection in surveillance videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 254–255.

[34] K. Doshi, Y. Yilmaz, Any-shot sequential anomaly detection in surveillance videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 934–935.

[35] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, S. Paul Smolley, Least squares generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2794–2802.

[36] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[37] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[38] M. Basseville, I. V. Nikiforov, et al., Detection of abrupt changes: theory and application, Vol. 104, prentice Hall Englewood Cliffs, 1993.

[39] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 fps in matlab, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 2720–2727.

[40] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 1975–1981.

[41] W. Luo, W. Liu, S. Gao, A revisit of sparse coding based anomaly detection in stacked rnn framework, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 341–349.

[42] R. Hinami, T. Mei, S. Satoh, Joint detection and recounting of abnormal events by learning deep generic knowledge, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3619–3627.

[43] J. Kim, K. Grauman, Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 2921–2928.

[44] A. Del Giorno, J. A. Bagnell, M. Hebert, A discriminative framework for anomaly detection in large videos, in: European Conference on Computer Vision, Springer, 2016, pp. 334–349.

[45] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, L. S. Davis, Learning temporal regularity in video sequences, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 733–742.

[46] Q. Sun, H. Liu, T. Harada, Online growing neural gas for anomaly detection in changing surveillance scenes, Pattern Recognition 64 (2017) 187–201.

[47] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, N. Sebe, Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 1689–1698.

[48] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection, in: CVPR 2011, IEEE, 2011, pp. 3449–3456.

[49] H. Mao, X. Yang, W. J. Dally, A delay metric for video object detection: What average precision fails to tell, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 573–582.

[50] S. N. Chiu, D. Stoyan, W. S. Kendall, J. Mecke, Stochastic geometry and its applications, John Wiley & Sons, 2013.

[51] T. C. Scott, G. Fee, J. Grotendorst, Asymptotic series of generalized lambert w function, ACM Communications in Computer Algebra 47 (3/4) (2014) 75–83.

Keval Doshi received the B.Sc. degree in Electronics and Communications Engineering from Gujarat Technological University, India, in 2017. He is currently a Ph.D. student at the Electrical Engineering Department at the University of South Florida, Tampa. His research interests include computer vision, machine learning and cybersecurity.

Yasin Yilmaz received the Ph.D. degree in Electrical Engineering from Columbia University, New York, NY, in 2014. He is currently an Assistant Professor of Electrical Engineering at the University of South Florida, Tampa. He received the Collaborative Research Award from Columbia University in 2015. His research interests include machine learning, statistical signal processing, and their applications to computer vision, cybersecurity, IoT networks, energy systems, transportation systems, environmental systems, and communication systems.