ANOMALY DETECTION FOR VIDEO-BASED SURVEILLANCE USING COVARIANCE FEATURES AND MODELING OF SEQUENCES VIA LSTMS

by
ALI ENVER BILECEN

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
Dec 2021

ANOMALY DETECTION FOR VIDEO-BASED SURVEILLANCE USING COVARIANCE FEATURES AND MODELING OF SEQUENCES VIA LSTMS

Approved by:

Approval: Dec 17, 2021

ABSTRACT

ANOMALY DETECTION FOR VIDEO-BASED SURVEILLANCE USING COVARIANCE FEATURES AND MODELING OF SEQUENCES VIA LSTMS

ALI ENVER BILECEN

ELECTRONICS ENGINEERING M.S. THESIS, DECEMBER 2021

Thesis Supervisor: Assist. Prof. Dr. Huseyin Ozkan

Keywords: anomaly detection, covariance features, long short-term memory, autoregressive modeling, support vector regression

In this thesis, we propose three different methods for anomaly detection in surveillance videos based on autoregressive modeling of observation likelihoods. By means of the methods we propose, normal (typical) events in a scene are learned in a probabilistic framework by estimating the features of consecutive frames taken from the surveillance camera. The proposed methods are based on long short-term memory (LSTM), linear regression, and support vector regression (SVR). To decide whether an observation sequence (i.e. a small video patch) contains an anomaly or not, its likelihood under the modeled typical observation distribution is thresholded. An anomaly is decided to be present if the threshold is exceeded. Due to its effectiveness in object detection and action recognition applications, covariance features are used in this study to compactly reduce the dimensionality of the shape and motion cues of spatiotemporal patches obtained from the video segments. Our proposed methods that are based on the final state vector of LSTM and support vector regression (SVR) applied to mean covariance features, and achieve an average performance of up to 0.95 area under curve (AUC) on benchmark datasets.

ÖZET

VIDEO BAZLI GÖZETIM SISTEMLERINDE KOVARYANS ÖZNITELIKLERI KULLANIMI VE DIZILERIN LSTM MODELLENMESI ILE ANOMALI SEZIMI

ALI ENVER BILECEN

ELEKTRONİK MUHENDİSLİĞİ YÜKSEK LİSANS TEZİ, ARALIK 2021

Tez Danışmanı: Dr. Öğr. Üyesi Hüseyin Özkan

Anahtar Kelimeler: anomali sezimi, kovaryans öznitelikleri, uzun kısa-soluklu bellek, özbağlanımsal modelleme, destek vektör bağlanımcı

Bu tezde, güvenlik kamera görüntülerinde anomali sezim problemi için özbağlanımsal olasılık kestirimine dayalı üç farklı yöntem önerilmektedir. Önerdiğimiz yöntemler vasıtasıyla bir sahnedeki normal yani tipik olaylar, güvenlik kamerasından alınan ardışık çerçeve özniteliklerinin olasılıksal bir çatı altında tahmin edilmesiyle öğrenilir. Önerilen yöntemler uzun kısa-soluklu bellek, doğrusal bağlanım ve destek vektör bağlanımcısı tabanlıdır. Bir gözlem dizisinin (bir video kesitinin) anomali içerip içermediğine karar vermek için modellenmiş tipik gözlem dağılımı altındaki olabilirliği eşiklenir. Kovaryans öznitelikleri nesne tespiti ve eylem tanıma uygulamalarındaki etkinliğinden dolayı, video kesitlerinden elde edilen zaman-mekansal parçaların şekil ve hareket işaretlerini kompakt bir şekilde düşük boyuta indirgemek için kullanıldı. Ayrıca gözlem dizisinin olasılıksal formülasyonu sonucunda dizideki statik ve dinamik bilginin video anomali sezim problemine katkılarını ayrı ayrı görünür kılarak probleme yeni bir bakış açısı kazandırmaktayız. Önerdiğimiz yöntemlerden uzun kısa-soluklu bellek'in son durum vektör modellemesi ve ortalama kovaryans öznitelikleri üzerinde çalışan destek vektör bağlanımcısı, genel kullanıma açık kıyaslama veri setlerinde 0.95 ortalamaya varan eğri altındaki alan (EAA) performansları elde etmektedirler.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis advisor Assist. Prof. Dr. Hüseyin Özkan for his continuous support of my studies and the invaluable guidance he gave to me. He has been an inspiration for me starting from the time I attended his first university lecture and he selflessly taught me so much ever since.

I would like to thank my thesis defense jury members Prof. Dr. Mustafa Ünel and Assist. Prof. Dr. Furkan Kıraç for their constructive criticism and insightful comments regarding my thesis studies.

I would like to thank my friends from my undergraduate and graduate years, especially Kaan and Berker for our deep and stimulating discussions, and also for their encouragement and presence through the challenges. Alp and M.Sami spent a tremendous amount of time assisting me and worked diligently for a joint publication. My sincere appreciation goes to my other lab partners Kutay, Bulut, Osman Berke, Deniz, Mehmet and Alaattin for making the time I spent at Sabanci University worthwhile and meaningful.

Most of all, I am grateful to my beloved family members, Hülya, Rasim and Gökçen, who were always there to support me.

This thesis study was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under Contract 118E268.

Dedicated to my family and friends

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Building a model that can robustly capture the normal modes of an environment is a long-studied research problem in machine learning and computer vision, enabling the detection of novelties/outliers in variety of applications such as industrial inspection, network security, healthcare, and automated surveillance [1–5]. In recent years, extensive research has been carried out on the anomaly detection problem for computer vision based surveillance purposes [6–11]. With the rapid deployment of surveillance cameras, the amount of data produced by these cameras far exceeds the amount that can be manually inspected by the security personnel. This indicates that anomaly detection has an immense practical value, and robust algorithms designed to this end can direct the attention of the security personnel with a minimum delay to unusual events such as violent acts, traffic accidents, and violation of public rules [12].

Anomalous events can be defined as observations that deviate significantly from the statistical patterns of normal, i.e. typical, observations [13] that are observed frequently. For example, in the context of surveillance-based anomaly detection, typical observations are pedestrians acting in ordinary ways, e.g. walking, whereas some example anomalous events are fighting, and illegal crossing, as can be seen in Fig. 1.1. Such events are generally low probable observations, hence it is usually infeasible to collect a sufficient amount of data from the set of such events. For this reasons, in the anomaly detection literature, generally the training set only consists of samples from the normal class and the problem is recast as the estimation of the shape of the distribution of these normal samples. New observations at test time are classified as normal or abnormal based on the degree of agreement with the modeled typical data distribution. On the other hand, since anomalies occur very rarely and in unprecedented ways, their distribution cannot be estimated reliably, and in the literature, anomalies are generally assumed to be uniformly distributed [13]. Under this assumption, thresholding the negative log-likelihood of samples under the typical data distribution yields the classifier which maximizes the detection power while minimizing the false positive rate [14–17].

(a) (b)

(c) (d)

(e) (f)

Figure 1.1 Examples from the ShanghaiTech Campus dataset containing both normal and abnormal activity. In the figures located at the left column, normal scene behavior is observed. Right column of figures contain abnormal scene activity, namely, a bicycle and a motorcycle illegally crossing the pedestrian walkway, and instance of a violent pedestrian action.

1.1 Extended Summary of the Thesis

In this thesis we propose methods to detect and localize anomalies in a given surveillance video. To this end, a video is partitioned into spatiotemporal patches (sequences of frame patches). Each patch is represented by a sequence of covariance features (each one extracted from a frame). The developed approaches operate on the level of spatiotemporal patches for modeling the normal events in the scene. LSTM-based approaches work under a nonlinear sequential prediction framework to

learn temporal evolution of covariance features, whereas the linear regression based method discards the temporal dynamics of the sequence. At test time, to decide whether a spatiotemporal patch is anomalous or not, the deviation from the learned model is examined. We evaluate our methods on two benchmark datasets, and also report our findings on the dataset we gathered on Sabanci University campus.

The rest of the thesis is organized as follows. In Chapter 2 we provide background information on covariance features and anomaly detection, and discuss relevant approaches. In Chapter 3, we provide the problem description, and explain how a video is represented as spatiotemporal parts and modeled for anomaly localization. After we explain the process of extracting the covariance features and propose variants of them in Chapter 4, we introduce our proposed methods based on linear and nonlinear regression in Chapter 5. We present our performance evaluation results on two benchmark datasets in Chapter 6, introduce the surveillance dataset we gathered on campus in Chapter 7 and finally conclude with Chapter 8.

2. RELATED WORK

In this chapter, relevant methods in the literature are discussed. Covariance features provide a way of compactly representing a region in the image by computing a sample covariance matrix of several low-level feature descriptors. In most object detection scenarios, the essence of a classes' distribution is captured by its covariance [18]. We start with an overview of methods that employ covariance features, and then we continue by discussing recent approaches to anomaly detection problem.

2.1 Covariance Features

Covariance features were first introduced in [18] for object tracking, since then they have proved to be effective in other computer vision tasks such as texture classification [19], pedestrian detection [20, 21], and action recognition [22, 23]. Work in [24] extends covariance features to model spatiotemporal patches by also calculating temporal gradients, for fire and flame detection. However, use of covariance features on video based anomaly detection applications has remained limited. [25] computes covariance features from whole spatiotemporal patches extracted from a video by using optical flow maps and RGB values as feature descriptors, and one-class SVM (OCSVM) is adopted for anomaly detection. Ergezer et al. [26] proposes a object tracking module based on covariance features that can be plugged into any model-free anomaly detection algorithm, such as [27–29]. In this thesis, covariance features are computed to summarize spatiotemporal information extracted from video patches. More specifically, a temporal sequence of feature vectors (each one corresponding to a frame) is computed to represent a video segment. We propose a novel approach to the anomaly detection problem by modeling the probability distribution of these sequences using long short-term memory (LSTM) and linear regression based methods.

2.2 Anomaly Detection

Anomaly detection refers to the identification of out-of-distribution data and is critical for safety and security applications.

Early work in vision-based anomaly detection considered tracking based methods [30–34]. Generally in these methods, objects that follow an unexpected trajectory are labeled as anomalies. However, robust tracking of objects is computationally challenging as the number of objects gets large or the objects get occluded. For this reason, histogram based features of low level descriptors (HOG, HOF) [35–37] are used to summarize patches extracted from frames to detect rare shape and/or motion patterns. In [38], different from previously proposed approaches, videos are represented from two views by two distinct and partially independent feature descriptors. First of these feature descriptors is used for detecting anomalies in the whole video frame, whereas the second one is used for anomaly localization. These features are learned using denoising autoencoders. Also, a new approach is presented for integrating the two views to perform both anomaly detection and localization in the testing phase in real-time. In a similar work [39], authors propose to train one autoencoder by minimizing the reconstruction error, while the other one is trained by sparsity constraints for obtaining sparse representations. These two autoencoders are combined as a cascade classifier for fast detection. Heuristically, a spatiotemporal cuboid is considered as a potential anomaly if it has a sparsity value higher than a sparsity threshold. These patches are resized to larger dimensions and analyzed for reconstruction errors by the first autoencoder. An anomaly is decided to be present if an error threshold is exceeded. With a slight deviation from the unsupervised learning paradigm, and as the first time in the literature, [40] trains a neural network model composed of alternating 3D and 2D convolutional layers, using both abnormal and normal samples, treating the anomaly detection as a binary classification problem.

Deep Autoencoding Methods More recently, deep learning methods have been extensively used to discover representations from typical data that better explain the underlying structure of normality than hand-crafted, problem specific feature descriptors. In [7], a convolutional autoencoder is trained to learn regular motion patterns by minimizing the reconstruction error of a temporal stack of frames. In [41], authors build up on [7] by further processing the temporal stack of feature maps at the bottleneck of the autoencoder by a ConvLSTM [42] layer while preserving the spatial information for anomaly localization. Abati et al. [11] propose

to train an autoencoder by jointly minimizing the reconstruction error of samples and the latent variables' entropy using autoregressive models as a way of forcing the autoencoder to capture regular patterns. [6] follows a patch-based modeling approach and trains separate Stacked Denoising Autoencoders(SDAE) using patches extracted at multiple scales and their corresponding optical flows, as well as their combination. However, the final anomaly score is obtained from one-class SVMs trained by the latent representations extracted from the bottleneck layer of SDAEs. Since these features are not learned in an end-to-end manner, they are suboptimal. Building upon this work, authors in [43] propose to learn the temporal regularity of the features extracted by the SDAEs via LSTMs. Also, a graph manifold ranking algorithm is proposed to reduce the false alarm rate. Similarly, motivated by the capability of sparse coding based anomaly detection algorithms, [44] proposes a temporally-coherent sparse coding algorithm and employ stacked-LSTMs for modeling temporal evolution of sparse coefficients. Ionescu et al. [10] encode both motion and appearance information of region proposals extracted from video frames using a convolutional autoencoder and treats the anomaly detection problem as a one-versus-rest classification problem by clustering the latent representations of samples into normality clusters, leveraging the advantages of supervised methods. Sabokrou et al. [8] extract features from intermediate feature maps of a pretrained CNN model for the statistical modeling of temporal and spatial regularities. A Gaussian distribution is fitted to each location in the intermediate feature map, and a local observation inside a particular feature map that do not conform to the corresponding distribution is rolled back into the original input image for anomaly localization. Work in [45] aims to detect anomalies and also to provide evidence for the occurence of these anomalies through the use of a kernel density estimator, which can be beneficial in real world applications. It's also the first step step towards producing explainable models in anomaly detection literature. Sultani et al. [46] works in a weakly-supervised anomaly detection framework. In the UCF-Crime dataset published along with the proposed detection algorithm, video-level ground truths are available, and the task is to localize the sources of anomalies in time. This work is unique, in the sense that it tackles the detection of abnormal activities in very long and untrimmed surveillance videos. For this, C3D features [47] which are originally proposed for action recognition tasks are used to drastically reduce the dimensionality of the lengthy video data. Building on this work, [48] employs graph convolutional networks to extract frame-level labels using video-level ground truths, and proposes an EM-like training schedule for cleaning the label noise, an inherent problem in weakly-supervised learning. These progressively cleaned labels at each iteration of the training phase are used for training neural network based anomaly detectors at the same time, which further boosts the performance. Sabokrou et al. [49]

proposes a cubic patch-based anomaly detection framework. A cascade of classifiers is used to improve accuracy and reduce the computational cost. Representing different normal events by just one set of features leads to inaccurate results and high computational costs. As there are events with varying levels of complexity (simple non-events such as background, and more complex events are dynamic anomalies involving motion), the provided solution works by weeding out less complex patches in earlier layers, using weak Gaussian classifiers. Between each layer of the network, classifiers are used to differentiate between normal patches and abnormal patches, as the patches get more complex in deeper layers of the network. As the first stage of the network, a deep but light 3D autoencoder is used for early identification of many normal patches. At the second stage of the network, remaining candidate patches are analyzed using a deeper 3D CNN model. A study of deep convolutional autoencoders in [50] examines the effects of different input combinations (such as optical flows, image gradients, and dynamic images) on the detection performance. The paper also proposes to measure the spatial complexity of a frame based on Kolmogorov complexity. In [51] 3D gradient features are calculated using PCANet [52], an unsupervised deep learning algorithm which uses principal component analysis to compute filters in its layers in a hierarchical manner. By computing 3D gradients, gradients in spatial domain encode appearance information while the gradients in temporal domain encode motion information. High-level gradient features obtained from PCANet are used to model events using deep Gaussian mixture models. At test time, if the likelihood of a patch is below some threshold value, that patch is labeled as an anomaly. Similar to [7], Tran et al. [53] train winner-take-all [54] convolutional autoencoders only on whole optical flow frames, but rather than using reconstruction error to decide for anomalies, latent features are applied to OC-SVM to obtain an anomaly score. This variant of autoencoders are trained by imposing sparsity in the convolution kernels, which enforces the discovery of distinct features.

Pretrained Plug-and-Play Methods In [55] authors propose to use any pretrained plug-and-play fully convolutional network without further training. Semantic features extracted from the last convolutional layer on a frame by frame basis, are used to compute a binary map with same spatial size using Iterative Quantization Hashing [56] (ITQ). ITQ is a method for projecting high-dimensional feature vectors into a binary space, and the parameters are learned in an unsupervised manner. The goal of the training is to produce similar binary maps for frames that indicates normal behaviour, such that when encountered with abnormal behaviors, they can be detected by observing abrupt changes of appearance in the binary maps. To localize anomalies in frames, corresponding locations in the binary map are rolled back to their receptive fields. The work in [57] proposes an abnormal event detection

framework based on unmasking, a technique previously used for authorship verification in text documents, without needing training data. The method works with short sliding-window and detect anomalies based on the sudden changes in extracted features' values.

Adversarial Learning Methods Recently, GANs have obtained outstanding results on video anomaly detection. GANs enable end-to-end training of two networks in such a way that both networks learn the underlying input distribution. More specifically, a GAN module is trained by conditioning the generation process on the input image to reconstruct/predict a semantically related image. Based on the anomaly detection algorithm, a combination of two networks(generator and/or discriminator) is used for producing the anomaly score. Building on the idea that abnormal frames cannot be predicted ahead of time under typical data distribution, [9] proposes to detect anomalies in a future frame prediction framework, and trains a GAN module to predict the next frame given a history of frames and the optical flow of the current frame, by regularizing the outputs of the generator network's intensity and gradients to be closer to the ground truth frames. A predicted future frame that deviates significantly from its corresponding ground truth frame is labeled as anomaly. In [58], two sets of GAN modules are trained to reconstruct the optical flow frame from the original frame and vice versa. These two studies assign anomaly scores based on deviations of the predictions/reconstructions from the ground truth. Similarly in [59], two sets of GAN modules are trained for cross-channel reconstruction. After the training phase, generators of the modules are discarded and the discriminator networks are used for locally detecting anomalies in a patch based manner. Wu et al. [60] also train a two-stream module, but different from other methods, they propose to use VAE/GAN module, where the latent spaces of the generator and the encoder of the VAE networks are connected. As the authors argue, this combination helps achieve better modeling the normal modes of behavior in the scene, and anomalies are decided based on the reconstruction errors. The studies [9, 58–60] show that the incorporation of the optical flow information boosts the performance of the anomaly detection algorithms drastically. The work in [61] proposes to train a "refiner" network (analogous to generator) to reconstruct the uncorrupted version of the input while the discriminator tries to discern between the real and reconstructed images. Both networks learn the normal data distribution, and using both networks back-to-back at testing time further boosts the performance as opposed to using only one of the networks. In similar terms, [62] trains an "inpainter" network that tries to produce uncorrupted version of its input, while the discriminator operates on patches of the outputs of the inpainter. Such a formulation enables robust localization of visual anomalies as well as providing

a way of controlling the trade-off between false-alarm and missed detection rates, generally caused by pixel-level and patch-level detectors, respectively.

In [7, 9, 41], anomalies are assigned directly to frames without localization. In our work [63], we localize anomalies, similar to [6] and [25], by extracting spatiotemporal volumes from a video and processing them independently. The works [11, 43, 44] are of special interest to us. In a similar manner to [11], we autoregressively model features extracted from frames, through the use of LSTM networks as in [44], while operating on the level of spatiotemporal patches [43].

2.3 Novel Contributions and Highlights

- We propose to use covariance features to fuse appearance and motion, both of which convey discriminative information, into a compact form for anomaly detection.
- We propose to incorporate mean information of low level descriptors into the calculation of covariance features, since the average activities of these descriptors also provide useful information for separating anomalous samples from normal ones.
- We model normal modes of behavior in a scene through a sequential prediction framework employing linear/nonlinear regression models, and declare anomalies based on the probabilistic modeling of the prediction errors or final state vectors of LSTMs.
- Methods that we propose achieve average AUC scores of up to 0.95 on three benchmark datasets, indicating the suitability of covariance features for anomaly detection.
- We present our own surveillance dataset, where the footage is collected from two scenes on Sabanci University campus, and present our results on this dataset.

3. PROBLEM DESCRIPTION

The aim of this study is to detect and localize video anomalies in space and time. A video is divided into spatiotemporal patches called *cuboids*, each of which consists of τ consecutive frames. From each cuboid, a sequence of feature vectors $\underline{\mathrm{X}} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_\tau\}$ is obtained. The probability distribution of this sequence can be expressed as the product of conditional probabilities:

$$p(\underline{\mathrm{X}}) = \prod_{i=1}^{\tau} p(\mathbf{x}_i \Big| \mathbf{x}_1, \ldots, \mathbf{x}_{i-1}). \tag{3.1}$$

To detect an anomaly given a feature sequence extracted from a cuboid, the probability distribution in (3.1) can be modeled using different methods such as quantizing each $\mathbf{x}_i$ and modeling them as multinomials [11], or by stacked masked convolutional layers [64]. Then, using these autoregressive probability models, the negative log-likelihood of the feature sequence is compared to a threshold T_a to decide whether the corresponding cuboid is anomalous or not. An anomaly is decided to be present if the threshold is exceeded

$$\phi(\underline{\mathrm{X}}) = \begin{cases} 1 \text{ (abnormal)} & -\log p(\underline{\mathrm{X}}) \geq T_a, \\ 0 \text{ (normal)} & -\log p(\underline{\mathrm{X}}) < T_a. \end{cases}$$

where ϕ is the anomaly decision rule.

In our approach, the minimum mean squared estimator (MMSE) $\theta^{(i)}_{\mathrm{MMSE}}$ estimates the mean of the distribution $p(\mathbf{x}_i \Big| \mathbf{x}_1, \ldots, \mathbf{x}_{i-1})$ for $i = 1, \cdots, \tau$. Each distribution is modeled as Gaussian around the corresponding mean and with variance equal to the mean squared prediction error of the estimator. LSTM networks [65] are suitable for this task as sequential nonlinear estimators [66, 67] since the autoregressive nature of the problem is automatically captured by recurrence. Moreover, LSTMs model all conditional distributions in (3.1) using the same set of parameters by keeping

a summary of all previously applied inputs using a state vector. We also consider using estimators based on support vector regression (SVR) and linear regression. Probabilities are assigned under the typical data distribution based on the prediction errors. Unlike these models that assign probabilities based on the prediction errors, the final state vector of LSTM can also be employed to assign probabilities to cuboids, since it is a condensed version of all sequentially applied inputs. Typical data distribution is approximated through the final state vector, and in test time, final state vectors that are observed to be outside of this distribution are labeled as anomalies. To estimate a feature value in the feature sequence based on the previous observations, dynamic information contained in the sequence is required to be extracted. However, for the initial observation $\mathbf{x}_1$, since there is no history, the best prediction is simply the mean of the marginal distribution, and hence constant. In the next section, we start with extracting covariance features from consecutive frame patches in cuboids.

4. EXTRACTION OF COVARIANCE FEATURES

In order to localize anomalies in space and time, we partitioned an RGB surveillance video into spatiotemporal volumes called *cuboids*, and each cuboid is assigned an anomaly score based on its likelihood under the typical data distribution. These cuboids can overlap in their spatial and temporal axes for precise localization. For a robust likelihood model, the dimensionality of the data must be reduced while preserving information that is relevant for anomaly detection. Covariance features provide an efficient way of summarizing local cuboid activity in a compact manner. Second order statistics that capture the linear relationship between random variables provide sufficient information for visual tasks [20,22] when the random variables are chosen as appropriate low level descriptors, such as image gradients and optical flow maps. Covariance features are also scale-invariant and robust to local distortions [18], which further boosts the generalization ability of the algorithms. Moreover, drastic reduction in dimensionality of the shape and motion descriptors, enabled by the covariance features, increases the computational power and alleviates problems encountered during training which arise due to high dimensional data.

To that end, a video $\mathbf{V} \in \mathbb{R}^{H\times W\times T\times 3}$ is partitioned into cuboids $V^{(c)} \in \mathbb{R}^{H_c\times W_c\times T_c\times 3}$, with a certain stride in space and time, depending on how densely cuboids are to be sampled. Anomalies can occur as a result of irregular shape and/or motion patterns. Image gradients [19, 68] and optical flow maps [22, 23, 25], when used as low level descriptors for the calculation of sample covariance matrix, give satisfactory results for tasks which require good visual and motion representations, respectively. Hence, from video $\mathbf{V}$, we consider computing image gradient, optical flow and binary foreground videos with the same spatial and temporal dimensions ($\mathbf{IG}, \mathbf{OF} \in \mathbb{R}^{H\times W\times T\times 2}$ and $\mathbf{FG} \in \mathbb{Z}_2^{H\times W\times T\times 1}$, respectively). To extract covariance features from a cuboid, we take the corresponding regions of these videos, namely $\mathbf{V}^{(c)}$, $\mathbf{IG}^{(c)}$, and $\mathbf{OF}^{(c)}$, all of which have dimensions (spatial and temporal) of $H_c \times W_c \times T_c$. This process is illustrated in Fig. 4.1.

In order for a cuboid to be anomalous, there needs to be a considerable amount of activity inside it. To determine whether a cuboid is active or not, the percentage of

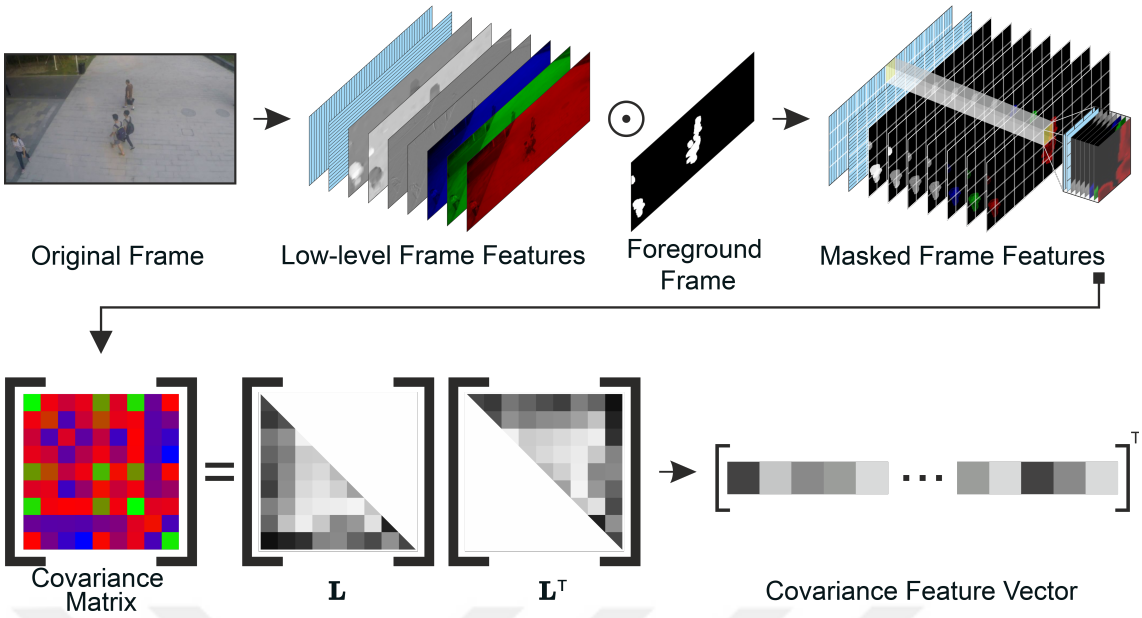


Figure 4.1 Extraction of the Covariance Features. From a video frame, low level descriptors such as RGB luminance values, image gradients, optical flows are obtained and stacked with x-y coordinate grids along the last axis. This frame stack is multiplied element wise with the corresponding foreground frame to obtain the masked frame stack by removing the background information. The cuboids that are determined as active are processed further by computing the covariance matrix from the region which corresponds to that cuboid. To covariance matrices, Cholesky decomposition is applied, and the feature vector is formed by taking all the nonzero elements of the resulting lower triangular matrix. Subsequently, these feature vectors are used by the proposed sequential prediction algorithms in Chapter 5.

the active pixels inside the corresponding region of the foreground video is thresholded with α_a and cuboids that surpass this threshold are named as active cuboids, i.e., active bricks satisfy the condition

$$\frac{1}{H_c \times W_c \times T_c} \sum_{x,y,t} FG^{(c)}_{x,y,t} \geq \alpha_a, \tag{4.1}$$

where x and y are indices for space, and t is the index for time. In this study, only the active cuboids are used for training the models and deciding for anomalies at test time. For each $t \in \{1,\dots,T_c\}$, the corresponding RGB image, image gradient and optical flow map are stacked along with the x-y coordinate grid $\mathbf{\Omega} \in \mathbb{R}^{H_c \times W_c \times 2}$, at the last axis. To get rid of the background information, this frame stack is multiplied element wise with $\mathbf{FG}^{(c)}_t$, the corresponding foreground image at time t. As the result of this multiplication, we obtain the background-free frame stack $\mathbf{F}^{(c)}_t \in \mathbb{R}^{H_c \times W_c \times 9}$ as follows: $\mathbf{F}^{(c)}_t = \left[\mathbf{V}^{(c)}_t, \mathbf{IG}^{(c)}_t, \mathbf{OF}^{(c)}_t, \mathbf{\Omega}\right] \odot \mathbf{FG}^{(c)}_t$. For notational

convenience, we refer to this tensor shortly as $\bar{\mathbf{F}}$. We compute the sample covariance matrix $\mathbf{C}$ by considering the last axis elements of $\bar{\mathbf{F}}$ as random variables. The i^{th} row and j^{th} column of $\mathbf{C}$ is computed by

$$\mathbf{C}_{i,j} = \frac{1}{N_a - 1} \sum_{k,l} (\bar{\mathbf{F}}_{k,l,i} - \mu_i)(\bar{\mathbf{F}}_{k,l,j} - \mu_j),$$

where N_a is the number of active pixels in the foreground image, and $\mu_i = \frac{1}{N_a} \sum_{k,l} \bar{\mathbf{F}}_{k,l,i}$ is the sample mean of the i^{th} entry in the last axis of $\bar{\mathbf{F}}$, at time t. Lastly, for further dimensionality reduction and achieving a Euclidean embedding, we apply Cholesky decomposition $\mathbf{C} = \mathbf{L}\mathbf{L}^\top$ following the work in [69]. We take all the nonzero elements of the lower triangular matrix $\mathbf{L} \in \mathbb{R}^{9\times 9}$ as the entries of the feature vector $\mathbf{x}_t \in \mathbb{R}^{45}$. Consequently, from a cuboid of length T_c, a feature vector sequence $\{\mathbf{x}_1, \ldots, \mathbf{x}_{T_c}\}$ is obtained.

4.1 Variants of the Proposed Covariance Features

We point out that the use of covariance features disposes of the information of mean activations of feature values, which can play a critical role for anomaly detection. For example, the mean of optical flows can provide useful information for determining the motion magnitude of an object. To incorporate mean information into the covariance features, we also consider adding the mean activities back to the covariance matrix to obtain the correlation matrix as $\mathbf{R} = \mathbf{C} + \mu\mu^\top$, and the Cholesky decomposition is applied to $\mathbf{R}$. Similarly, the use of foreground masks discards the global motion information (motion of the object as a whole) by placing the origin point on the center of mass of the moving object. One can also consider not masking the foreground frame to preserve this information. As a result, we obtain 4 combinations of features: covariance with/without foreground masking, and correlation with/without foreground masking. For ease of exposition, all these features will be simply mentioned as covariance features.

5. PROPOSED METHODS

In this section, we present our methods that model the typical data distribution based on covariance features. First two of these methods work under an autoregressive nonlinear prediction framework and model temporal evolution of covariance features. In these methods, a feature vector $\mathbf{x}$ at time t is predicted as a function of all the observation history. However, in the last two methods, limitations are put on the observation history, and linear prediction models are used. The dataset $\{(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_\tau^{(i)})\}_{i=1}^N$ (we use τ instead of T_c for simplicity and generality) is obtained by extracting covariance features from active cuboids. Each sample, indexed by i, consists of a sequence of τ vector observations, $\mathbf{x}_t \in \mathbb{R}^d$ for $t \in \{1, \ldots, \tau\}$ and $d = 45$. This dataset is used for training the 3 different methods that will be presented shortly. First two of these methods, LSTM-GSV and LSTM-GPE, employ nonlinear regression whereas the last method, MCF-LRGM, uses linear regression to model sequences. Two additional methods, ϵ-SVR and ν-SVM are used as baselines.

5.1 LSTM-based Nonlinear Methods

For the nonlinear methods, LSTM networks are used. This enables modeling sequences of different lengths while having a fixed number of parameters through weight sharing. Additionally, unlike hidden Markov models (HMM) [70, 71], predictions of the network depend on all previously applied inputs, i.e., the length of the window of the observation history is infinite. At each time step t, LSTM network takes the current observation $\mathbf{x}_t$ and the previous state vector $\mathbf{s}_t$ as input, then outputs the prediction for the next observation $\hat{\mathbf{x}}_{t+1}$ in the sequence and the next state vector $\mathbf{s}_{t+1}$, as shown in Fig. 5.1. Training is also conducted in a many-to-many setting, where the first $\tau - 1$ observations in the sequence, in order, are applied as inputs to predict the last $\tau - 1$ of them. Specifically, we calculate the sum of

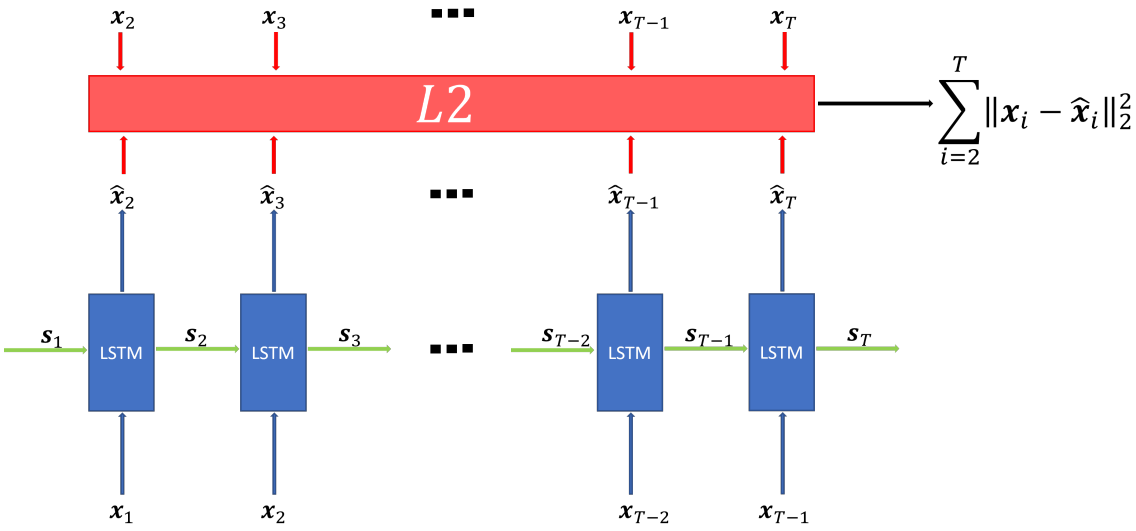


Figure 5.1 Block diagram illustrating the training of the LSTM. The weights are learned in a many-to-many sequential prediction paradigm by minimizing the sum of squared errors of predictions.

squared prediction errors at all timesteps, for a single observation sequence, by the loss function shown below:

$$\mathcal{L}(\{\mathbf{x}_i\}_{i=2}^{\tau}, \{\hat{\mathbf{x}}_i\}_{i=2}^{\tau}) = \frac{1}{\tau - 1} \sum_{t=2}^{\tau} \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2.$$

Then, for a minibatch of M samples (observation sequences), total average loss is calculated by:

$$\mathcal{L}(\{\{\mathbf{x}_i^{(j)}\}_{i=2}^{\tau}, \{\hat{\mathbf{x}}_i^{(j)}\}_{i=2}^{\tau}\}_{j=1}^{M}) = \frac{1}{M} \sum_{k=1}^{M} \mathcal{L}(\{\mathbf{x}_i^{(k)}\}_{i=2}^{\tau}, \{\hat{\mathbf{x}}_i^{(k)}\}_{i=2}^{\tau}),$$

where $j \in \{1, \ldots, M\}$ is the index of the samples.

However, when used in a sequential prediction setting, LSTMs (and RNNs in general) do not fully learn the joint distribution of the observation sequence in (3.1). Since the LSTM is not tasked with predicting the initial observation $\mathbf{x}_1$, its distribution $p(\mathbf{x}_1)$ is not modeled by the network. By training the LSTM in a sequential prediction setting, the network learns to discard the static information carried in the sequence since it's not useful to predict the next observation in the sequence. The initial observation $\mathbf{x}_1$, which we consider as the carrier of static information neglected by the network, might also indicate an anomaly, hence the initial obser-

vation needs to be incorporated into the probability models explicitly. An overview of LSTM-based methods is illustrated in Fig. 5.2.

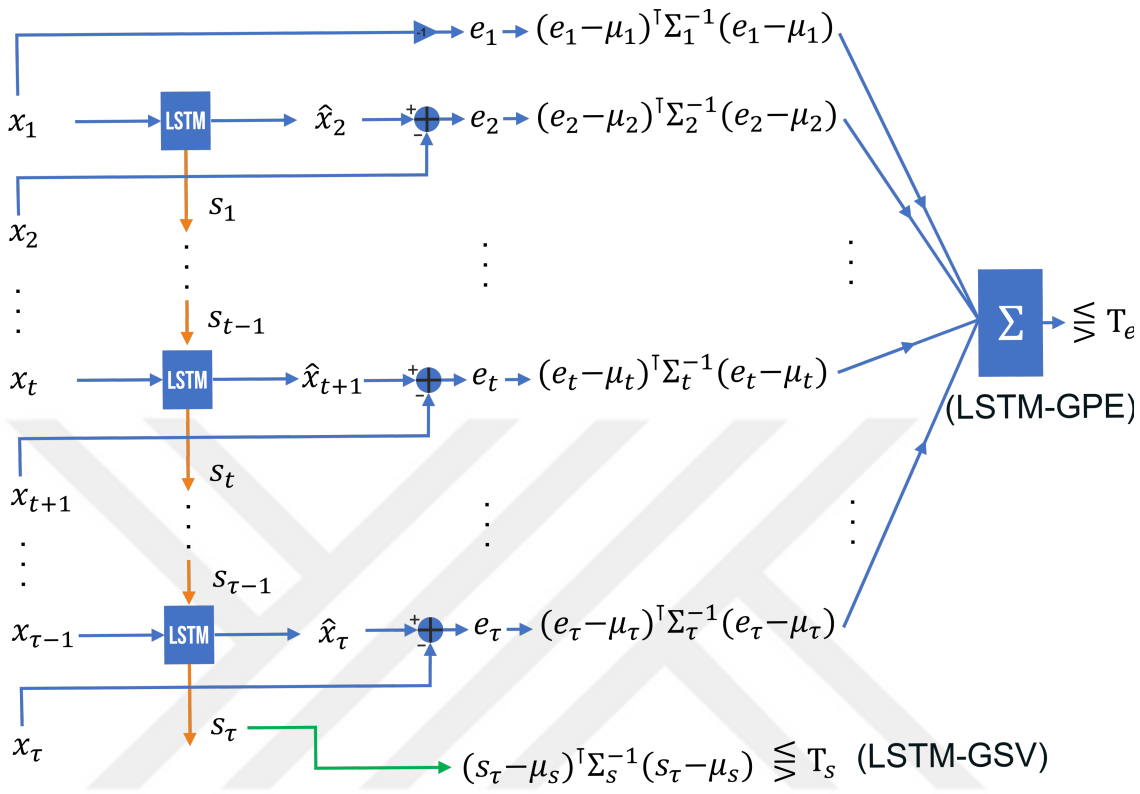


Figure 5.2 Block diagram representations of the LSTM-based nonlinear methods. In the LSTM-GPE method, deviations of the predictions from their ground truths are accumulated for all timesteps $t = 1, \cdots, \tau$, and anomalies are decided by thresholding the accumulation of deviations. The deviations are assumed to be distributed normally, separately for all timesteps. In the LSTM-GSV method, to decide whether the observation sequence contains an anomaly or not, the likelihood of the obtained final state vector after applying all the inputs is thresholded. Gaussianity is assumed for the final state vector. An anomaly is decided to be present if the threshold is not exceeded.

5.1.1 LSTM-GSV: Modeling the Final State Vector of LSTM as Gaussian

The final state vector $\mathbf{s}_\tau$ is obtained by feeding the sequence $\{\mathbf{x}_1, \ldots, \mathbf{x}_{\tau-1}\}$ to the LSTM. The state vector encodes information about previously applied inputs, therefore, the probability distribution in (3.1) can be approximated via the distribution $p(\mathbf{s}_\tau)$:

$$p(\mathbf{s}_\tau) \approx \prod_{i=2}^{\tau} p(\mathbf{x}_i |\ \mathbf{x}_1, \ldots, \mathbf{x}_{i-1}).$$

We assume that the final state $\mathbf{s}_\tau$ is a normally distributed random vector with parameters mean vector μ_s and covariance matrix Σ_s. To estimate these parameters from data, the dataset of covariance observation sequences $\{(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_\tau^{(i)})\}_{i=1}^N$ is applied to the LSTM, and the dataset of final state vectors $\{\mathbf{s}_\tau^{(i)}\}_{i=1}^N$ is obtained. Sample mean vector $\mu_s = \frac{1}{N}\sum_{i=1}^N \mathbf{s}_\tau^{(i)}$, and the sample covariance matrix $\Sigma_s = \frac{1}{N-1}\sum_{i=1}^N (\mathbf{s}_\tau^{(i)} - \mu_s)(\mathbf{s}_\tau^{(i)} - \mu_s)^\top$ are computed. The likelihood of the observation sequence, $p(\underline{\mathrm{X}})$, is calculated through $p(\mathbf{s}_\tau)$ as

$$p(\mathbf{s}_\tau) = \frac{1}{(2\pi)^{d/2}|\Sigma_s|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{s}_\tau - \mu_s)^\top \Sigma_s^{-1}(\mathbf{s}_\tau - \mu_s)\}.$$

By taking the logarithm, the log-likelihood of $\mathbf{s}_\tau$ is obtained:

$$\log p(\mathbf{s}_\tau) = -\frac{1}{2}\log(2\pi)^d|\Sigma_s| - \frac{1}{2}(\mathbf{s}_\tau - \mu_s)^\top (\Sigma_s)^{-1}(\mathbf{s}_\tau - \mu_s).$$

Incorporating the Initial Observation To incorporate the effect of the initial observation $\mathbf{x}_1$, we concatenate $\mathbf{s}_\tau$ and $\mathbf{x}_1$ to obtain the *augmented final state vector* $\bar{\mathbf{s}}_\tau = [\mathbf{s}_\tau, \mathbf{x}_1]$, the same steps mentioned above are followed to calculate $\log p(\bar{\mathbf{s}}_\tau)$.

5.1.2 LSTM-GPE: Modeling Prediction Errors of LSTM as Gaussian

In this method, we model the prediction errors of the LSTM network, separately for all timesteps t. Let $f_t(.)$ be the function that takes as input all feature vectors observed till time t, and outputs a prediction $\hat{\mathbf{x}}_t$ corresponding to the actually observed feature vector $\mathbf{x}_t$, at time t. The input-output relationship can be shown as $\hat{\mathbf{x}}_t = f_t(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$. Since the set of previous observations is empty at $t = 1$, the function $f_1(.)$ that predicts $\mathbf{x}_1$ does not take any inputs, and its output is identically $\mathbf{0}$ by construction. It should be noted that this function can take any constant value, and the end result of the algorithm will not be affected due to the Gaussian assumption on prediction errors. For all t and samples $i \in \{1, \ldots, N\}$,

error vectors $\mathbf{e}_t^{(i)} \triangleq \mathbf{x}_t^{(i)} - \hat{\mathbf{x}}_t^{(i)}$ are calculated and modeled as normally distributed: $p(\mathbf{x}_1) \triangleq \mathcal{N}(\mathbf{e}_1; \mu_1, \Sigma_1)$ and $p(\mathbf{x}_i|\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}) \triangleq \mathcal{N}(\mathbf{e}_i; \mu_i, \Sigma_i)$.

Here, μ_i and Σ_i are the mean vector and covariance matrix of the error vector $\mathbf{e}_i$ corresponding to the i^{th} observation, and computed in the same way as in the LSTM-GSV method in 5.1.1. Then, the likelihood of the observation sequence $\underline{\mathrm{X}}$ is calculated through the likelihood of the sequence of prediction errors $\underline{\mathrm{E}} = \{\mathbf{e}_1, \ldots, \mathbf{e}_\tau\}$ as

$$p(\underline{\mathrm{E}}) = \prod_{i=1}^{\tau} \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{e}_i - \mu_i)^\top \Sigma_i^{-1}(\mathbf{e}_i - \mu_i)\}.$$

By taking the logarithm, $\log p(\underline{\mathrm{E}})$ is calculated as follows:

$$\log p(\underline{\mathrm{E}}) = \sum_{i=1}^{\tau} -\frac{1}{2}\log(2\pi)^d|\Sigma_i| - \frac{1}{2}(\mathbf{e}_i - \mu_i)^\top \Sigma_i^{-1}(\mathbf{e}_i - \mu_i). \tag{5.1}$$

Incorporating the Initial Observation It should be noted that unlike in the LSTM-GSV method, the step of incorporating the initial observation $\mathbf{x}_1$ is seamless and controlled by changing the starting value of the summation variable in 5.1 from $i = 2$ to $i = 1$, as the prediction $\hat{\mathbf{x}}_1$ is identically equal to $\mathbf{0}$, hence $\mathbf{x}_1 = \mathbf{e}_1$.

5.2 MCF-LRGM: Autoregressive Gaussian Modeling of Mean

Covariance Features

This method is based on linear regression and provides an alternative to LSTM-based methods. Although the probability distribution of a feature sequence $\underline{\mathrm{X}}$ can directly be modeled as in (3.1), the learned model can only be applied to sequences of fixed length τ, as the distributions of $\mathbf{s}_t$ and $\mathbf{e}_t$ change with respect to t, and therefore the parameters have to be calculated for all t. Additionally, using different sets of parameters for each step of the prediction increases the parameter complexity of the model as the sequence gets longer. Moreover, for complex models, in the absence of sufficient training data, parameter estimation errors can accumulate and affect the prediction performance in a detrimental way. For these reasons, to reduce the

dimensionality of the observations, we take the average of all τ vector observations to obtain the mean feature vector $\bar{\mathbf{x}} \in \mathbb{R}^d$, for which we hypothesize that it still carries a considerable amount of information about the activity in a cuboid. The probability distribution of $\bar{\mathbf{x}}$ can be expressed as

$$p(\bar{\mathbf{x}}) = \prod_{i=1}^{d} p(\bar{x}_i \Big| \bar{x}_1, \ldots, \bar{x}_{i-1}),$$

where $\bar{x}_i \in \mathbb{R}$ is the i^{th} component of the mean observation vector $\bar{\mathbf{x}}$, computed by averaging the corresponding elements of the observation sequence $\underline{\mathrm{X}}$. Under the assumption of Gaussianity, unconditional distribution $p(x_1)$ and the conditional distributions $p(x_i \Big| x_1, \ldots, x_{i-1})$ for $i = 2, \ldots, d$ are modeled with parameters mean μ_i and variance σ_i^2. For each distribution, linear regression models $f_i(.)$ are trained to predict x_i given all the previous observations $x_1, \ldots, x_{i-1}$. For the distribution $p(x_i \Big| x_1, \ldots, x_{i-1})$, μ_i is chosen as the prediction of the i^{th} model, i.e., $\hat{x}_i = \mu_i = f_i(x_1, \ldots, x_{i-1})$ and the variance $\sigma_i^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$ is calculated as the mean squared prediction error of the model. For the initial observation x_1, the linear prediction function $f_1(.)$ does not take any argument since the history of observations before x_1 is an empty set. For this reason the prediction for x_1 is constant, i.e. $f_1(.) \equiv \bar{x}_1$, and it is simply taken as the mean of all initial observations in the dataset. Variance σ_1^2 is calculated in the same manner. The likelihood of $\underline{\mathrm{X}}$ is calculated through the mean observation vector $\bar{\mathbf{x}}$ as

$$p(\bar{\mathbf{x}}) = \prod_{i=1}^{d} \frac{1}{(2\pi\sigma_i^2)^{1/2}} \exp\{-\frac{1}{2}(\frac{x_i - \mu_i}{\sigma_i})^2\}.$$

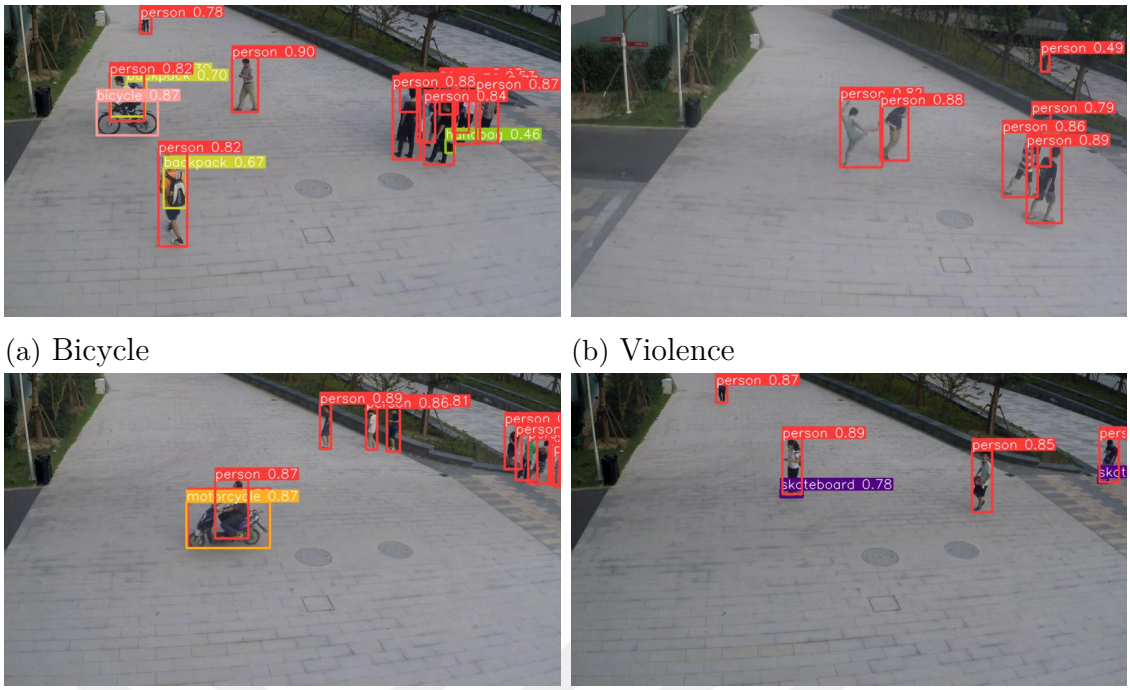
The log-likelihood of $p(\bar{\mathbf{x}})$ is calculated as

$$\log p(\bar{\mathbf{x}}) = -\frac{1}{2}\sum_{i=1}^{d} \log 2\pi\sigma_i^2 - \frac{1}{2}\sum_{i=1}^{d} \frac{(\bar{x}_i - \mu_i)^2}{\sigma_i^2}.$$

5.3 Using Object Detection Modules for Detecting Scene Activity

The criterion based on a background subtraction algorithm for determining active regions in surveillance footage tend not to work as desired when the camera is not static, there are sudden illumination changes, and the objects in motion are the same color as the background, resulting in false alarms and misses in the detection of the activities, respectively. Also, the object in motion can be much smaller than the cuboid in terms of their spatial size, in which case the foreground activity generated by the object will be absorbed by the relative largeness of the cuboid, hence the activity will not be detected. In order to alleviate these issues, as an alternative to the activity detection method based on the average foreground activity to determine whether a brick is active or not, we consider using an object detection module, YOLO [72], a fast and lightweight deep neural network that can robustly work in diverse settings. The YOLO object detection module is mostly invariant to the textures and colors of the objects that it detects, and it can work in different illumination conditions and when there is camera motion. The output of this module is a bounding box that encapsulates the detected object. Possibly, it can output multiple bounding boxes, each for a different detected object. In most cases, a non maximum suppression (NMS) step is not needed due to the additional training the networks that forces them to output bounding boxes that place the objects of interest at the dead center [73], enabling the overall detection pipeline to be much more faster for real-time applications.

Given an RGB image $I_t \in \mathbb{R}^{H \times W \times 3}$ assumed to be containing objects, taken from a video at time t, the YOLO module outputs a list of region of interests (RoI) $\{(x_{tl}^{(i)}, y_{tl}^{(i)}, x_{br}^{(i)}, y_{br}^{(i)})\}_{i=1}^{C}$ where $(x_{tl}^{(i)}, y_{tl}^{(i)})$ is the coordinate of the top-left, and $(x_{br}^{(i)}, y_{br}^{(i)})$ is the coordinate of the bottom-right corner of the i^{th} bounding box, and C is the total number of detected objects (as illustrated in Fig. 5.3). We crop spatiotemporal patches of length T_c starting with time t forward in time, with the given coordinates, to obtain cuboids. Then, from each frame patch, we extract covariance figures. It should be noted that, cuboids that are extracted via the YOLO module do not have a fixed spatial size. However, the extracted covariance features will have a fixed dimensionality of 45, and with the scale-invariance property of the covariance features, the generalization capacity of the algorithms will be promoted. The proposed methods in Chapter 5 can be directly applied to the cuboids extracted using the YOLO module.



(a) Bicycle

(b) Violence

(c) Motorcycle

(d) Skateboard

Figure 5.3 Obtained bounding boxes by the YOLOv5 object detection module. Using the coordinates of these bounding boxes, we extract spatiotemporal patches from the consecutive frames of the video, and compute covariance features to represent the spatiotemporal patch.

5.3.1 YOLO-GMM: Gaussian Mixture Modeling of Object Proposals

We propose an additional method that shows covariance features can also yield good performance without the sequential modeling via LSTM networks. To this end, we closely follow the work in [74] for detecting regions of interest using YOLO object detection module. Different from this work, to represent the regions of interest, we compute covariance features of the region instead of extracting features using autoencoders. By passing through all the data to obtain bounding boxes with the YOLO module, we extract frame patches and compute the covariance features. In the proposed methods introduced above, due to multiple steps of linear/nonlinear projections of the covariance features as input data, Gaussianity assumption that is made for the outputs were appropriate, as also validated by the experimental results. However, in the case of modeling raw covariance features (covariance features that are not processed in any way, e.g. LSTM-based methods), Gaussianity assumption yields subpar results. For this reason, we model raw covariance features via mixture of Gaussians.

A covariance feature $\mathbf{x}$'s probability density under a mixture model is found as follows:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k),$$

with π_k's being the mixing coefficients (priors) where

$$\sum_{k=1}^{K} \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \quad \forall k,$$

and where K is a hyperparameter for determining the number of Gaussian components. The parameters $\pi_k, \mu_k,$ and Σ_k of the mixture are found by the Expectation-Maximization (EM) algorithm [75]. Deriving an analytical expression for $\log p(\mathbf{x})$ is not trivial, hence, we directly threshold $p(\mathbf{x})$ to decide for abnormal regions.

5.4 Anomaly Score Function

The anomaly scores are determined based on the likelihood ratio test (LRT). Let $\mathbf{y}$ be the observation that can be the final state vector $\mathbf{s}_\tau$, a temporal stack of prediction errors $\{\mathbf{e}_1, \ldots, \mathbf{e}_\tau\}$, and so on, depending on the method of choice. Let $H_0 \colon \mathbf{y} \sim \mathbf{p}_0$ and $H_1 \colon \mathbf{y} \sim \mathbf{p}_1$ be the hypotheses that $\mathbf{y}$ follows the typical data distribution $\mathbf{p}_0$ or the anomalous data distribution $\mathbf{p}_1$, respectively. Since we do not have any information regarding the distribution of anomalies, they are assumed to be uniformly distributed, i.e., anomalies can be encountered anywhere on the observation space equally likely. In that case, the anomaly score function $\mathbf{S}_a(\mathbf{y})$ can be expressed as

$$\mathbf{S}_a(\mathbf{y}) \triangleq L(\mathbf{y}) = \frac{\mathbf{p}_1(\mathbf{y})}{\mathbf{p}_0(\mathbf{y})} \propto -\log \mathbf{p}_0(\mathbf{y})$$

since $\mathbf{p}_1(\mathbf{y})$ is constant for all observations of $\mathbf{y}$, based on the uniformity assumption. Anomaly detection can be carried out by thresholding the negative log-likelihood of observations under the typical data distribution, which is modeled by the proposed methods.

5.5 Baseline Methods

We explain two baseline methods based on support vector machines, namely Support Vector Regressor (ϵ-SVR) and One-Class Support Vector Machine (ν-SVM). These powerful methods are used for comparing our proposed algorithms in terms of area under curve (AUC) scores.

5.5.1 ϵ-SVR: Support Vector Machine for Regression

Support Vector regression (ϵ-SV regression) is a machine learning technique introduced in [76] that aims to regress a target variable y through the use of a sparse weight vector w, which is determined only by the samples x_i that are the Support Vectors. For the linear regression case where $x \in \mathbb{R}$ the target function f is in the form of

$$f(x) = \langle w, x \rangle + b,$$

and the goal is to find f that makes no absolute error larger than ϵ, and oblivious to errors as long as the predictions are inside the "ϵ-tube" of their corresponding ground truths. Additionally, flatness of w is enforced, meaning that out of all w's that satisfy the ϵ-tube condition, the one with the smallest norm is preferred as a way of regularization. However, such a target function f that makes a maximum absolute error of ϵ does not always exist, therefore, through the introduction of slack variables ξ_i and ξ_i^* for each instance x_i, the optimization problem that was infeasible becomes copeable. Now, the optimization problem is formulated as

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\
\text{subject to} \quad & \\
& y_i - \langle w, x \rangle - b \geq \epsilon + \xi_i, \\
& \langle w, x \rangle + b - y_i \geq \epsilon + \xi_i^*, \\
& \xi_i, \xi_i^* \geq 0,
\end{aligned}$$

where the constant $C \geq 0$ is a hyperparameter for determining the trade-off between the flatness of w and tolerance for errors that are larger than ϵ. This idea for the linear regression can be generalized for the nonlinear case as well, through the use of appropriate kernel functions [76].

We use the $\epsilon - SV$ regression algorithm in the context of modeling mean covariance features introduced in 5.2, to model $\bar{\mathbf{x}}$ in an autoregressive manner. For this, we use the Gaussian Kernel (RBF), where the kernel parameter σ and box constraint C are chosen by using 5-fold nested cross-validation.

5.5.2 ν-SVM: One-Class Support Vector Machine

One-class SVM (ν-SVM) was introduced in [77], for problems of outlier/novelty detection. In the case of having a single class where the task is to find the deviations from the distribution of this class, the data is lifted to a high dimensional space through a kernel mapping, and a hyperplane which encapsulates a specified portion of the data is found by solving an optimization problem. This corresponds to finding the minimum volume set of the data that covers $N(1-\nu)$ of the samples, where N is the number of samples, and $\nu \in (0, 1]$ is a hyperparameter of the algorithm.

More specifically, ν-SVM finds a decision rule f such that, on average, ν portion of the data sampled from the classes' distribution is mapped to 1 through

$$f(x) = \operatorname{sgn}\left(\sum_i \alpha_i k(x_i, x) - \rho\right),$$

where $k(x_i, x) = (\Phi(x_i).\Phi(x))$, and Φ is a feature mapping function. $k(x_i, x)$ selected as the radial basis kernel function (RBF), and x_i's are the support vectors with coefficients α_i's that are nonzero. The coefficients α_i's are found by solving the optimization problem given below:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||w||^2 + \frac{1}{\nu l}\sum_{i=1}^{N}\xi_i - \rho \\ \text{subject to} \quad & \\ & y_i - (w, \Phi(x_i)) \geq \rho - \xi_i, \\ & \xi_i \geq 0. \end{aligned}$$

These coefficients can also be found by solving the dual problem with respect to coefficients:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j k(x_i, x_j) \\ \text{subject to} \quad & \\ & 0 \leq \alpha_i \leq \frac{1}{\nu l}, \\ & \sum_i \alpha_i = 1. \end{aligned}$$

This method is also used in the context of modeling mean covariance features in 5.2. Different from the ϵ-SVR method, no probabilistic modeling is carried out. To determine whether an observation sequence contains an anomaly or not, we calculate the anomaly score $S(x)$ as

$$S(x) = -\left(\sum_i \alpha_i k(x_i, x) - \rho\right),$$

where a larger score indicates an out-of-distribution sample with high probability.

In the next chapter, we explain our experimental setup and provide performance evaluation results.

6. PERFORMANCE EVALUATIONS

In this chapter, we present our experimental results on three benchmark datasets: ShanghaiTech Campus [9], UCSD Ped1 and UCSD Ped2 [78]. We also introduce a new evaluation criterion called *cuboid-level* criterion to measure the performance of the presented algorithms.

6.1 Datasets and Experimental Setup

Datasets UCSD Ped 1&2 datasets contain recordings of a single scene and anomalies are defined as vehicles such as cars, bicycles and skateboards passing through pedestrian walkways. ShanghaiTech Campus dataset includes recordings of multiple scenes. In addition to vehicles with anomalous activities, this dataset introduces anomalies that are characterized by sudden motion such as brawling and chasing. In this dataset, we use the scene with the prefix “01" for evaluation. This is the scene that has the most training/testing samples, making up a big portion of the dataset with approximately 70000 training and 12000 testing frames.

Cuboid-level evaluation criteria The evaluation criteria proposed in [78], namely *frame-level* and *pixel-level* criteria, are used for determining the locality agnostic detection and localization performance, respectively. Based on these criteria, it is sufficient to find a single anomalous pixel to label the corresponding frame as abnormal. We consider that this procedure can cause relatively more false alarms, without any substantial and observable change in the pixel-level decisions. The pixel-level decisions should cover a considerable number of pixels to declare anomalies in a meaningful way. For that, we introduce a new evaluation criterion called *cuboid-level* criterion for robustness against pixel-level classification noise.

Cuboid-level anomaly labels are determined based on the experimental setup (user

specified cuboid size and the spatial overlap factor) so that each cuboid has a corresponding label. This label assignment is carried out in a similar manner to cuboid activity detection. Let $\Omega \in \mathbb{N}^{H_c \times W_c \times T_c}$ denote the cubic grid of coordinates corresponding to a cuboid, and $\mathbf{GT}(x, y, t)$ be the function that maps a coordinate in the video to its pixel-level ground truth. A cuboid-level ground truth is determined as **1** (abnormal) if

$$\frac{1}{|\Omega|} \sum_{(x,y,t)\in\Omega} \mathbf{GT}(x, y, t) \geq \beta,$$

where $|\Omega|$ is the cardinality of Ω and β is a user specified threshold. We reasonably choose $\beta = 0.2$ as it covers a considerable percentage of the anomaly volume. In this study, Receiver Operating Characteristic (ROC) curves are generated based only on the cuboids that are determined as active.

6.2 Experimental Results

To extract covariance features, cuboid activity threshold α_a is chosen as 0.25. Spatial cuboid sizes are chosen as 24×16 and 64×32 for UCSD Ped 1&2 and ShanghaiTech Campus datasets, respectively. These spatial sizes are chosen in such a way that pedestrians that are close to the cameras can fit inside, and the ones that are far away are also captured by the action detection algorithm. In ShanghaiTech Campus dataset, we test out three different temporal cuboid sizes 6, 11, 21 and report performances for each. In UCSD Ped1&2 datasets, the temporal cuboid size of 21 is not considered in the experiments, since the spatial sizes of the cuboids are already small, and moving objects simply enter and leave the cuboids in much less number of frames.

For the LSTM-GSV and LSTM-GPE methods, the state dimensionality is chosen as 1024 and 8, respectively. For training, 10% of the dataset is used as validation data. Minibatch size is chosen as 512, initial learning rate is 10^{-3} and is reduced with a factor of 0.75 when validation loss plateaus. For the LSTM based methods, we repeat the experiment 10 times with different initializations and report their average result. As previously mentioned, we observe that the covariance feature variant of the best performance varies from a dataset to another. Furthermore, we

provide results for two other methods based on mean covariance features in Method 5.2 (MCF-LRGM), based on one-class support vector machine (ν-SVM) [77] and support vector regressor (ϵ-SVR) [79]. For each dataset and method, the reported AUC scores correspond to the best performing variant of the covariance features.

For all datasets, ϵ-SVR and MCF-LRGM methods achieve the highest performance using correlation features with FG masking while no FG masking variant of the covariance feature performs better in the SVR method. Information about used covariance feature variant for LSTM methods are provided with the results tables. Performance evaluations for ShanghaiTech Campus dataset are shown in Table 6.1. It can be seen that SVR and MCF-LRGM are the highest performing methods with averages of 0.92 and 0.894, respectively. Fig. 6.2 shows the successful detection of anomalous activities, such as a car and bicycle passing through the pedestrian walkway. Results for UCSD Ped1 and Ped2 are presented in Table 6.2 and Table 6.3, respectively indicating that SVR and MCF-LRGM perform favorably.

6.2.1 Results on the ShanghaiTech Campus Dataset

	AUC			
Method	$T_c = 6$	$T_c = 11$	$T_c = 21$	Avg.
LSTM-GSV	0.876	0.877	0.903	0.885
LSTM-GPE	0.881	0.880	0.900	0.887
MCF-LRGM	0.888	0.888	0.907	0.894
ν-SVM	0.778	0.793	0.810	0.794
SVR	0.917	0.916	0.931	0.921

Table 6.1 ShanghaiTech Campus Dataset Results. LSTM-based methods using covariance features without FG masking.

As can be seen from Table 6.1, all methods except the ν-SVM algorithm perform favorably, in terms of AUC score, on ShanghaiTech Campus dataset. Although SVR is the best performing method for this dataset, it has much longer training time compared to all other algorithms.

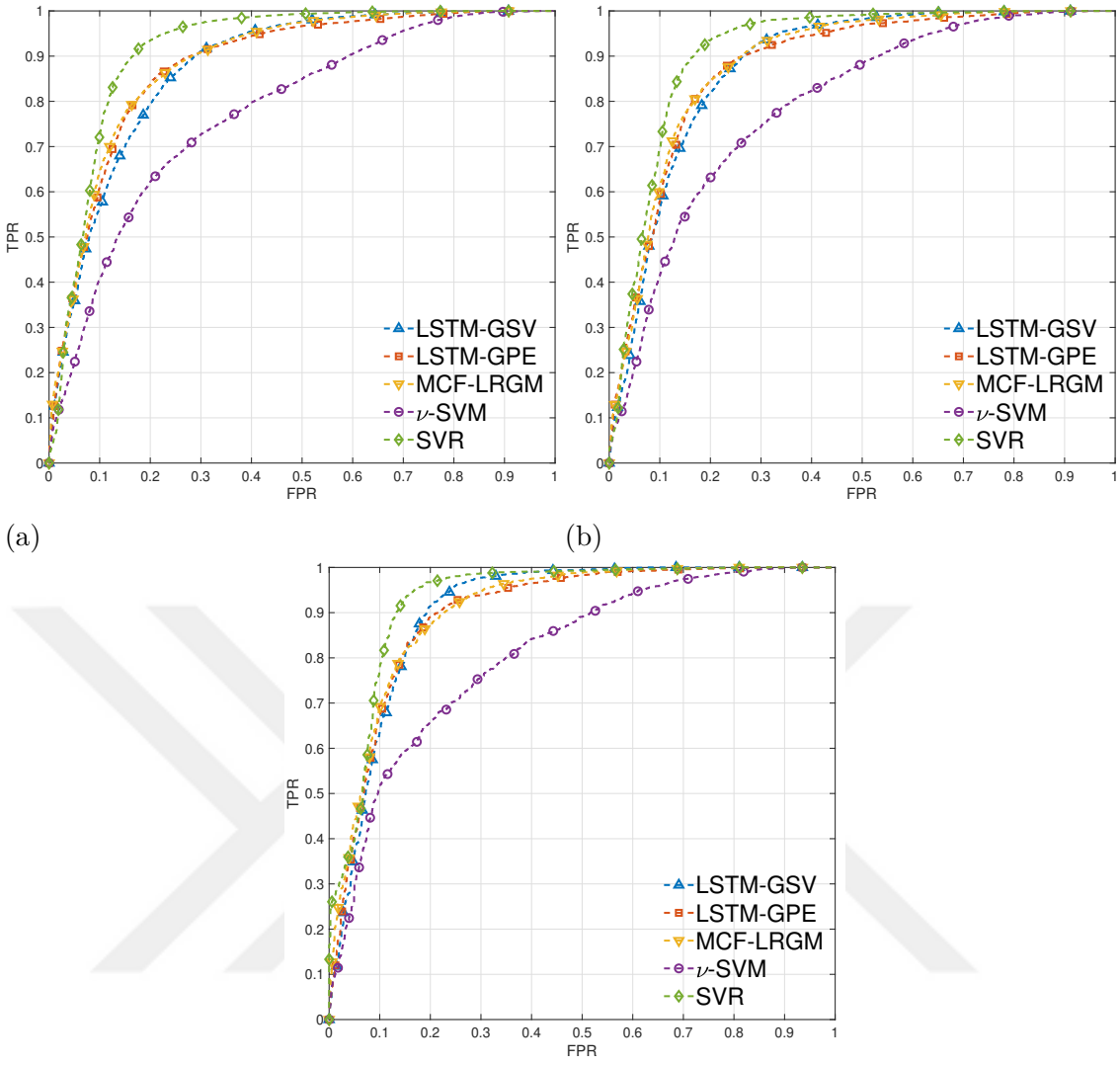


Figure 6.1 ROC curves obtained from the ShanghaiTech Campus dataset for (a) $T_c = 6$, (b) $T_c = 11$ and (c) $T_c = 21$ by the proposed methods.

ROC figures in Fig 6.1 for ShanghaiTech Campus dataset indicates that, for all cuboid lengths $T_c = 6, 11, 21$, SVR performs with the lowest false positive rate (FPR) when the true positive rate (TPR) is kept constant across all other methods. When compared against the ROC curves for UCSD Ped1 and UCSD Ped2 datasets in Fig. 6.3 and Fig. 6.4 respectively, the earlier parts of the plots are much less steep. This suggests that differentiating between the normal and abnormal samples in ShanghaiTech Campus dataset is a much complicated task.

(a) Car (b) Bicycle

(c) Motorcycle (d) Handcart

Figure 6.2 Detected anomalies on ShanghaiTech Campus dataset by the proposed LSTM-GSV method.

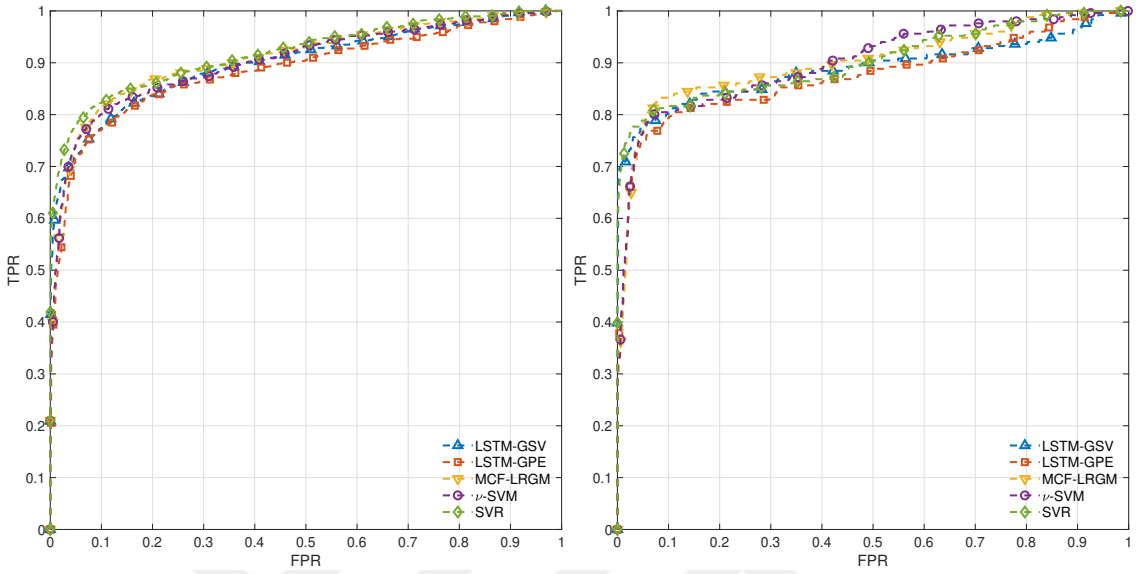
The detected anomalies in ShanghaiTech Campus dataset are shown in Fig 6.2. The LSTM-GSV method can successfully detect abnormal events and objects, such as illegal crossings from a pedestrian walkway and an unlikely object (handcart), without producing false alarms for regular pedestrians.

6.2.2 Results on the UCSD Ped1 Dataset

	AUC		
Method	$T_c = 6$	$T_c = 11$	Avg.
LSTM-GSV	0.900	0.889	0.899
LSTM-GPE	0.886	0.875	0.881
MCF-LRGM	0.909	0.902	0.906
ν-SVM	0.904	0.902	0.903
SVR	0.919	0.902	0.911

Table 6.2 UCSD Ped1 Dataset Results. LSTM-based methods using correlation features with FG masking.

UCSD Ped1 dataset consists of short 34 training and 36 testing video samples. Videos are recorded with a frames per second (FPS) and have low resolution. For this reason, almost no meaningful events are captured within cuboids of length 21. We carry out our experiments only with $T_c = 6, 11$.



(a) (b)

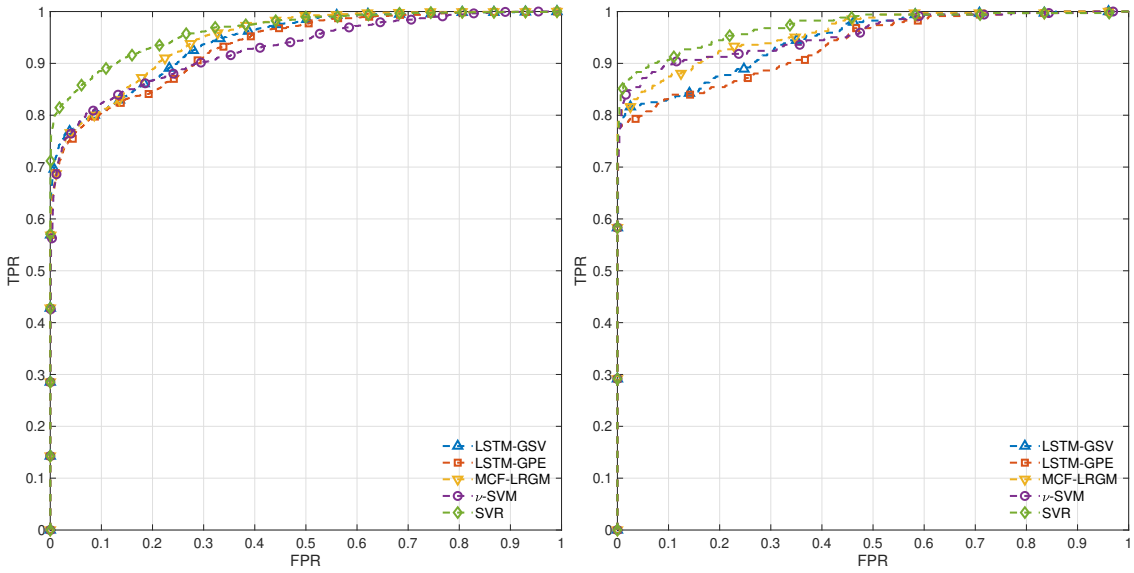
Figure 6.3 ROC curves obtained from the UCSD Ped1 dataset for (a) $T_c = 6$, (b) $T_c = 11$ by the proposed methods.

6.2.3 Results on the UCSD Ped2 Dataset

UCSD Ped2 dataset consists of short 16 training and 12 testing video samples. With similar reasons as in UCSD Ped1 dataset, we only use cuboids of length 6 and 11.

	AUC		
Method	$T_c = 6$	$T_c = 11$	Avg.
LSTM-GSV	0.941	0.945	0.943
LSTM-GPE	0.932	0.933	0.932
MCF-LRGM	0.946	0.959	0.953
ν-SVM	0.927	0.955	0.941
SVR	0.963	0.971	0.967

Table 6.3 UCSD Ped2 Dataset Results. LSTM-based methods using correlation features with FG masking.



(a) (b)

Figure 6.4 ROC curves obtained from the UCSD Ped2 dataset for (a) $T_c = 6$, (b) $T_c = 11$ by the proposed methods.

As can be seen in Tables 6.2 and 6.3 for UCSD Ped1 and Ped2 datasets, all methods perform favorably, while SVR produces the best results in terms of AUC scores. It should be noted that correlation features with foreground (FG) masking is used for both of these datasets, whereas covariance features without FG masking is used for ShanghaiTech Campus dataset. This demonstrates that global motion information plays a discriminative role in the case of ShanghaiTech Campus dataset. Qualitatively, it can be observed that the discplacement of different types of objects (regular pedestrian versus a vehicle) between frames provide useful information for anomaly detection. This distinction between global motion of objects is much more bleak in the case of UCSD Ped1&2 datasets, since the velocity of vehicles and pedestrians are closer to each other compared to the ShanghaiTech Campus dataset. On the other hand, using correlation features incorporates mean values of low-level descriptors back to the feature vector, which could give further cues about the spatial distribution of optical flows inside a spatiotemporal patch and also about the shape of the objects. It is reasonable to assume that this type of information would help to differentiate between normal and abnormal scenes when the global motion information does not play a discriminative role.

6.2.3.1 YOLO-GMM Result on UCSD Ped2 Dataset

Here, we present our results obtained by the YOLO-GMM method on the UCSD Ped2 dataset. It should be noted that different from other results reporting cuboid-level performances, we evaluate YOLO-GMM method using frame-level criteria.

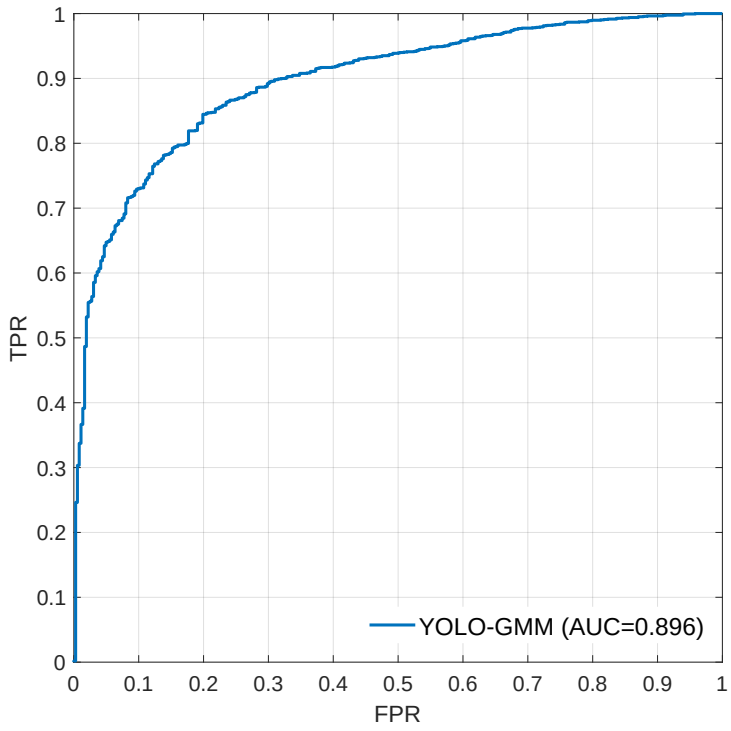


Figure 6.5 Frame-level ROC curve obtained from the UCSD Ped2 dataset using YOLO-GMM method. Here, we use correlation features without FG masking.

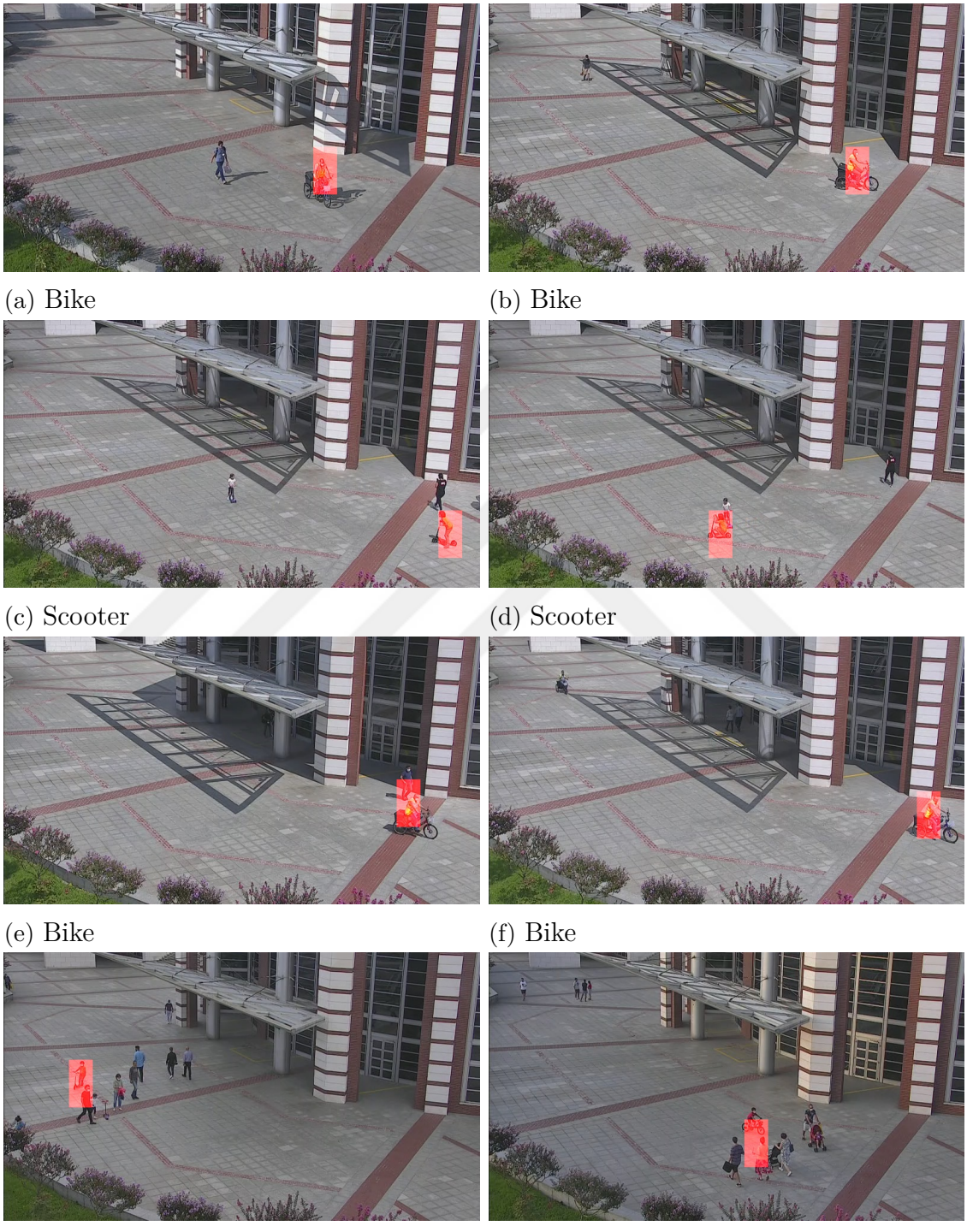
YOLO-GMM method achieves a frame-level AUC score of 0.896 (Fig. 6.5) with $K = 20$, indicating the effectiveness of covariance features when combined with object detectors.

In the following chapter, we present our qualitative findings on the surveillance dataset recorded on Sabanci University campus.

7. SURVEILLANCE DATASET

We publish a new dataset for surveillance-based anomaly detection. The videos are recorded at 30 frames per second (FPS) from two high resolution cameras at the Sabanci University campus center, in different times of the day with varying illumination and weather conditions, crowd behavior and rich set of events. In line with the published datasets for anomaly detection, the durations of the recordings are between 10 and 30 seconds, in which meaningful scene events can be observed. Some of the key challenges in real-world anomaly detection applications are camera jitter, sudden illumination changes between daytime/nighttime or due to weather conditions, installment of new stationary objects to the scene, and changes in scene dynamics depending on the time of the day. All of these challenges can be observed in the dataset, making it a pushing factor for further development of new algorithms for anomaly detection, scene analysis and action recognition. Moreover, the two cameras in the campus are in close proximity and can be configured such that their fields of view overlap for studying multi-view anomaly detection problems. Here, we present some of the scene samples from the dataset and present our findings via the methods in 5, in the context of anomaly detection.

7.1 Detected Anomalies From University Center Camera



(a) Bike (b) Bike

(c) Scooter (d) Scooter

(e) Bike (f) Bike

(g) Scooter (h) Bicycle

Figure 7.1 Frames from SUrveillance Dataset. Cuboids that have maximum anomaly scores are painted in red. LSTM-GSV method is used.

The detected anomalies shown in Fig. 7.1 correspond to the spatiotemporal patches that produce maximum anomaly scores when the LSTM-GSV method in 5.1.1 is used. It can be seen that these patches contain objects with abnormal shapes (bike, bicycle, scooter), but also have irregular motion patterns.

In the following chapter, we conclude with final remarks and future directions.

8. CONCLUSION

In this thesis, we proposed methods for anomaly detection in surveillance videos based on autoregressive probability modeling and estimation. We investigated the suitability and effectiveness of different types of covariance features for this task. LSTM-based methods model the temporal dynamics of these features using autoregressive probability estimation, whereas the other proposed methods model the average activity in a spatiotemporal volume. Successes of both types of methods indicate the overall usefulness of covariance features in the framework of autoregressive probability estimations for video anomaly detection. Moreover, we used object detectors as an alternative to the foreground-based activity detection criterion, and observed that this choice results in more accurate detection of activities in the scene and fewer miss detections, as well as an improved richness in the extracted features. We published a dataset gathered on the campus of Sabanci University, and showed that our proposed methods are able to successfully differentiate between normal and abnormal scene activities.

BIBLIOGRAPHY

[1] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9592–9600, 2019.

[2] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*, pp. 703–716, Springer, 2019.

[3] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, "Sensor fault and patient anomaly detection and classification in medical wireless sensor networks," in *2013 IEEE international conference on communications (ICC)*, pp. 4373–4378, IEEE, 2013.

[4] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*, pp. 146–157, Springer, 2017.

[5] V. Saligrama and M. Zhao, "Local anomaly detection," in *Artificial Intelligence and Statistics*, pp. 969–983, PMLR, 2012.

[6] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," *arXiv preprint arXiv:1510.01553*, 2015.

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

[8] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded

scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.

[9] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection–a new baseline," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6536–6545, 2018.

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

[11] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 481–490, 2019.

[12] V. Saligrama, J. Konrad, and P.-M. Jodoin, "Video anomaly identification," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 18–33, 2010.

[13] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[14] M. Zhao and V. Saligrama, "Anomaly detection with score functions based on nearest neighbor graphs," *Advances in neural information processing systems*, vol. 22, pp. 2250–2258, 2009.

[15] B. Can and H. Ozkan, "A neural network approach for online nonlinear neyman-pearson classification," *IEEE Access*, vol. 8, pp. 210234–210250, 2020.

[16] H. Ozkan, F. Ozkan, and S. S. Kozat, "Online anomaly detection under markov statistics with controllable type-i error," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1435–1445, 2015.

[17] H. Ozkan, F. Ozkan, I. Delibalta, and S. S. Kozat, "Efficient np tests for anomaly detection over birth-death type dtmcs," *Journal of Signal Processing Systems*, vol. 90, no. 2, pp. 175–184, 2018.

[18] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 728–735, 2006.

[19] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *European conference on computer vision*, pp. 589–600, Springer, 2006.

[20] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.

[21] S. Paisitkriangkrai, C. Shen, and J. Zhang, "Fast pedestrian detection using a cascade of boosted covariance features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1140–1151, 2008.

[22] K. Guo, P. Ishwar, and J. Konrad, "Action recognition from video using feature covariance matrices," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2479–2494, 2013.

[23] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell, "Spatio-temporal covariance descriptors for action and gesture recognition," in *2013 IEEE Workshop on applications of Computer Vision (WACV)*, pp. 103–110, IEEE, 2013.

[24] Y. H. Habiboğlu, O. Günay, and A. E. Çetin, "Covariance matrix-based fire and flame detection method in video," *Machine Vision and Applications*, vol. 23, no. 6, pp. 1103–1113, 2012.

[25] T. Wang, M. Qiao, A. Zhu, Y. Niu, C. Li, and H. Snoussi, "Abnormal event detection via covariance matrix for optical flow based feature," *Multimedia Tools and Applications*, vol. 77, no. 13, pp. 17375–17395, 2018.

[26] H. Ergezer and K. Leblebicioğlu, "Anomaly detection and activity perception using covariance descriptor for trajectories," *European Conference on Computer Vision*, pp. 728–742, 2016.

[27] H. Ozkan, O. S. Pelvan, and S. S. Kozat, "Data imputation through the identification of local anomalies," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2381–2395, 2015.

[28] M. Kerpicci, H. Ozkan, and S. S. Kozat, "Online anomaly detection with bandwidth optimized hierarchical kernel density estimators," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[29] H. Ozkan, N. D. Vanli, and S. S. Kozat, "Online classification via self-organizing space partitioning," *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 3895–3908, 2016.

[30] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[31] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 9, pp. 1450–1464, 2006.

[32] X. Wang, K. Tieu, and E. Grimson, "Learning semantic scene models by trajectory analysis," in *European conference on computer vision*, pp. 110–123, Springer, 2006.

[33] I. N. Junejo, O. Javed, and M. Shah, "Multi feature path modeling for video surveillance," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, pp. 716–719, IEEE, 2004.

[34] E. Gundogdu, H. Ozkan, and A. A. Alatan, "Extending correlation filter-based visual tracking by tree-structured ensemble and spatial windowing," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5270–5283, 2017.

[35] S. Wang, E. Zhu, J. Yin, and F. Porikli, "Video anomaly detection and localization by local motion based joint video representation and ocelm," *Neurocomputing*, vol. 277, pp. 161–175, 2018.

[36] M. Javan Roshtkhari and M. D. Levine, "Online dominant and anomalous behavior detection in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2611–2618, 2013.

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

[38] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-time anomaly detection and localization in crowded scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 56–62, 2015.

[39] M. Sabokrou, M. Fathy, and M. Hoseini, "Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder," *Electronics Letters*, vol. 52, no. 13, pp. 1122–1124, 2016.

[40] S. Zhou, W. Shen, D. Zeng, M. Fang, Y. Wei, and Z. Zhang, "Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Signal Processing: Image Communication*, vol. 47, pp. 358–368, 2016.

[41] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *International symposium on neural networks*, pp. 189–196, Springer, 2017.

[42] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *arXiv preprint arXiv:1506.04214*, 2015.

[43] Y. Feng, Y. Yuan, and X. Lu, "Deep representation for abnormal event detection in crowded scenes," in *Proceedings of the 24th ACM international conference on Multimedia*, pp. 591–595, 2016.

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

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

[46] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6479–6488, 2018.

[47] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

[48] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1237–1246, 2019.

[49] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1992–2004, 2017.

[50] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, "A study of deep convolutional auto-encoders for anomaly detection in videos," *Pattern Recognition Letters*, vol. 105, pp. 13–22, 2018.

[51] Y. Feng, Y. Yuan, and X. Lu, "Learning deep event models for crowd anomaly detection," *Neurocomputing*, vol. 219, pp. 548–556, 2017.

[52] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: A simple deep learning baseline for image classification?," *IEEE transactions on image processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[53] H. T. Tran and D. Hogg, "Anomaly detection using a convolutional winner-take-all autoencoder," in *Proceedings of the British Machine Vision Conference 2017*, British Machine Vision Association, 2017.

[54] A. Makhzani and B. Frey, "Winner-take-all autoencoders," *arXiv preprint arXiv:1409.2752*, 2014.

[55] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and 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)*, pp. 1689–1698, IEEE, 2018.

[56] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2916–2929, 2012.

[57] R. Tudor Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Unmasking the abnormal events in video," in *Proceedings of the IEEE international conference on computer vision*, pp. 2895–2903, 2017.

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

[59] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1896–1904, IEEE, 2019.

[60] H. Wu, J. Shao, X. Xu, F. Shen, and H. T. Shen, "A system for spatiotemporal anomaly localization in surveillance videos," in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1225–1226, 2017.

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

[62] M. Sabokrou, M. Pourreza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, "Avid: Adversarial visual irregularity detection," in *Asian Conference on Computer Vision*, pp. 488–505, Springer, 2018.

[63] A. E. Bilecen, A. Ozalp, M. S. Yavuz, and H. Ozkan, "Video anomaly detection with autoregressive modeling of covariance features," *Signal, Image and Video Processing*, pp. 1–8, 2021.

[64] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[65] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[66] H. Ozkan, M. A. Donmez, S. Tunc, and S. S. Kozat, "A deterministic analysis of an online convex mixture of experts algorithm," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 7, pp. 1575–1580, 2014.

[67] N. D. Vanli, M. O. Sayin, M. Mohaghegh, H. Ozkan, and S. S. Kozat, "Nonlinear regression via incremental decision trees," *Pattern Recognition*, vol. 86, pp. 1–13, 2019.

[68] F. Porikli and T. Kocak, "Robust license plate detection using covariance descriptor in a neural network framework," in *2006 IEEE International Conference on Video and Signal Based Surveillance*, pp. 107–107, IEEE, 2006.

[69] K. J. Shih, A. Dundar, A. Garg, R. Pottorf, A. Tao, and B. Catanzaro, "Video interpolation and prediction with unsupervised landmarks," *arXiv preprint arXiv:1909.02749*, 2019.

[70] H. Ozkan, A. Akman, and S. S. Kozat, "A novel and robust parameter training approach for hmms under noisy and partial access to states," *Signal Processing*, vol. 94, pp. 490–497, 2014.

[71] H. Ozkan, R. Temelli, O. Gurbuz, O. K. Koksal, A. K. Ipekoren, F. Canbal, B. D. Karahan, and M. Ş. Kuran, "Multimedia traffic classification with mixture of markov components," *Ad Hoc Networks*, vol. 121, p. 102608, 2021.

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

[73] D. Wang, C. Li, S. Wen, Q.-L. Han, S. Nepal, X. Zhang, and Y. Xiang, "Daedalus: Breaking nonmaximum suppression in object detection via adversarial examples," *IEEE Transactions on Cybernetics*, 2021.

[74] Y. Ouyang and V. Sanchez, "Video anomaly detection by estimating likelihood of representations," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8984–8991, IEEE, 2021.

[75] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[76] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1996.

[77] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt, *et al.*, "Support vector method for novelty detection.," in *NIPS*, vol. 12, pp. 582–588, Citeseer, 1999.

[78] 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, 2013.

[79] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.