ABNORMAL EVENT DETECTION IN SURVEILLANCE VIDEOS USING SPATIOTEMPORAL AUTOENCODER

By

CHONG YONG SHEAN

A dissertation submitted to the Department of Internet Engineering and Computer Science,
Lee Kong Chian Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Engineering Science
February 2019

ABSTRACT

ABNORMAL EVENT DETECTION IN SURVEILLANCE VIDEOS USING SPATIOTEMPORAL AUTOENCODER

Chong Yong Shean

This research presents an efficient method for detecting anomalies in videos. Recent applications of convolutional neural networks have shown promises of convolutional layers for object detection and recognition, especially in images. However, convolutional neural networks are supervised and require labels as learning signals. Hence, a spatiotemporal autoencoder architecture is proposed for anomaly detection in videos including crowded scenes. The proposed architecture includes two main components, one spatial autoencoder for learning feature representation, and one temporal autoencoder for learning the temporal evolution of the spatial features. During training, the model is trained with only normal scenes, with the objective to minimise the reconstruction error between the input video volume and the output video volume reconstructed by the learned model. After the model is trained, normal video volume is expected to have low reconstruction error, whereas abnormal video volume is expected to have a high reconstruction error. By thresholding on the error produced by each testing input volumes, our system will be able to detect when an abnormal event occurs. The model is evaluated on four surveillance video datasets and compared using the area under ROC curve and abnormal event count. Experimental results on UMN, Avenue, and UCSD benchmarks confirm that the

proposed method can detect more abnormal events with lower false alarm rate than some state-of-the-art methods. The advantage of the proposed method is that it is unsupervised — the only ingredient required is long video segments containing most normal events in a fixed view. Also, no feature engineering is required as the model automatically learns the most useful features from the training data. Further investigations will be carried out to improve the result of video anomaly detection by having human feedback to update the learned model for better detection and reduced false alarms.

ACKNOWLEDGEMENT

First, I would like to express my gratitude towards my academic supervisor, Dr Tay Yong Haur, for his continued guidance and support throughout the course of my study and always being available with his critical suggestions for the improvement of my work. I would like to thank my co-supervisor Dr Goh Yong Kheng for offering me valuable advice and encouragement.

I am thankful to my academic institution, Universiti Tunku Abdul Rahman has provided the workspace and facilities which are required to finish my thesis. I would like to extend my thanks to EV-Dynamic (http://www.evd.com.my) for providing partial funding support. Also, I would like to acknowledge Mr Tang Xin Jie, Mr Ng Choon Boon and fellow research group members for providing research ideas and creating a friendly environment at work.

Finally, I am indebted to my family members and friends who have motivated me continuously throughout my life. It would not have been possible to complete this thesis if without their encouragement and loving support.

APPROVAL SHEET

This dissertation entitled "**ABNORMAL EVENT DETECTION IN SURVEILLANCE VIDEOS USING SPATIOTEMPORAL AUTOENCODER"** was prepared by CHONG YONG SHEAN and submitted as partial fulfillment of the requirements for the degree of Master of Engineering Science at Universiti Tunku Abdul Rahman.

Approved by:

_____
(Dr. Tay Yong Haur)
Date:……..
Associate Professor/Supervisor
Department of Internet Engineering and Computer Science
Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

_____
(Dr. Goh Yong Kheng)
Date:……..
Assistant Professor/Co-supervisor
Department of Mathematical and Actuarial Sciences
Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

FACULTY OF ENGINEERING AND SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

Date:

SUBMISSION OF DISSERTATION

It is hereby certified that ***Chong Yong Shean*** (ID No: ***14UEM06689***) has completed this dissertation entitled "*Abnormal Event Detection in Surveillance Videos using Spatiotemporal Autoencoder*" under the supervision of Dr Tay Yong Haur (Supervisor) from the Department of Internet Engineering and Computer Science, Lee Kong Chian Faculty of Engineering and Science, and Dr Goh Yong Kheng (Co-Supervisor)* from the Department of Mathematical and Actuarial Sciences, Lee Kong Chian Faculty of Engineering and Science.

I understand that University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

_____

(*Chong Yong Shean*)

*Delete whichever not applicable

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name Chong Yong Shean

Date _____

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

In daily life, surveillance cameras can be found everywhere, from indoor like shops and workplaces, to outdoor such as roads and highways. The primary purposes of these cameras are to provide a sense of security to the public, provides evidence, and also to prevent crime.

Currently, surveillance videos are either being monitored by human security officials, or for record purpose only. However, It is difficult for security guards to stay alert to what is happening in the videos being monitored, since video consists of mostly monotonous scenes, and unusual events rarely occur. Therefore, there is a need to automate the process so that less human effort is required to monitor the video stream and to scan every footage in search for interesting events.

Existing video surveillance solutions in the market mainly fall into one of these three categories: a) business surveillance, b) home security, c) baby or pet monitoring. In business scenarios, surveillance solutions are employed to monitor activity around a business or retail store. Cameras are installed to monitor office buildings after hours for security, whereas in retail stores they are used to discourage theft. People who are concerned about their home security may purchase a surveillance camera to install at home and monitor activities around their homes. Parents who wish to monitor their baby's needs when they are resting or busy with other activities may also purchase a camera to place it near their baby so they would not miss any event or accident. Surveillance cameras and systems are adverse in the market for consumers to meet their specific needs. Among the popular commercial solutions are provided by large companies such as D-Link, Nest, and Sharp. These systems offer to stream live videos

to users device on demand, while some also send email or SMS alerts to users when events happen at the installed venue.

1.2 Problem Statement

The ability to detect anomalies in real-time is very valuable, so that appropriate actions can be taken as soon as it is detected to avoid or reduce negative consequences. Thus, many research efforts are done to replace the need for manually detecting anomalous situations, to create an automated video surveillance system. Despite the importance, accurately determining anomalies can be very challenging.

The processing pipeline for such systems usually involves several steps including pre-processing, feature detection and description, sequence or context modelling, and anomaly detection based on certain measure or threshold. Depending on the feature detection method, the pre-processing step might include background subtraction, object detection and tracking. To achieve the objective of automatically detecting anomalous events, some appearance and dynamics of events have to be captured. Some examples of conventional feature extractors are optical flow-based descriptors (Xiao et al., 2015; Laptev et al., 2008; Reddy et al., 2011) and trajectory-based descriptors (Zhou et al., 2015; Li et al., 2011; Piciarelli et al., 2008; Mo et al., 2014).

Most research works focus on hand engineering features (Kläser et al., 2008; Cong et al., 2013; Roshtkhari and Levine, 2013) for particular scenes or datasets, but these features need to be manually tuned each time a different scenario is introduced. Meanwhile, deep learning methods are trending in visual-based tasks, due to its capability to produce good representations with raw input. Therefore in this research, we put emphasis on applying deep learning methods to extract discriminative features from video data.

Generally speaking, the term "unusual events" refers to all of the events which can potentially cause security interest. However, the definition of unusual events is highly contextual. For business surveillance, an unusual event can be entering premises after hour; for home surveillance, an unusual event can be loitering around the doors and windows of the house; and for traffic surveillance, an unusual event can be driving a vehicle in the direction opposite to the main flow.

Also, an event considered to be unusual in one context can be a usual event in another environment. For example, running in a restaurant would be unusual, but running at a park would be normal. Moreover, the definition of unusual events can be ambiguous and often vaguely defined. A person may think walking around on a subway platform is normal, but some may think it should be flagged as an anomaly since it could be suspicious. These challenges have made it difficult to automatically identify video patterns that produce anomalies in real-world applications.

1.3 Related Works

There are many successful cases in the related field of action recognition, such as Tran et al. (2015); Karpathy et al. (2014); Ji et al. (2013) and Oneata et al. (2013). However, these methods only apply to labelled video footages where events of interest are clearly defined and do not involve highly occluded scenes, such as crowded scenes. Furthermore, the cost of labelling every type of event is extremely high. Even so, it is not guaranteed to cover every past and future events. The recorded video footage is likely not long enough to capture all types of activities, especially abnormal activities which rarely or never occurred.

Recent effort on detecting anomalies by treating the task as a binary classification problem (normal and abnormal) in Zhou et al. (2016) proved it

being effective and accurate, but the practicality of such method is limited since footages of abnormal events are difficult to obtain due to its rarity.

Therefore, many researchers have turned to models that can be trained with the absence of abnormal footages (Cong et al., 2013; Lu et al., 2013; Sabokrou et al., 2015; Hasan et al., 2016). Typically, researchers build and train a statistical model with features extracted from videos containing only normal events, then based on the trained model, unseen video segments that are unable to fit well into the trained model (i.e. samples with low probability measured based on the trained model) would be detected as unusual events. Since normal events are much more abundant and easily obtained compared to unusual events, this approach is more applicable to real-world scenarios.

Various types of features are used to represent the appearance and motion dynamics in videos. These include, but not limited to trajectory features (Jiang et al., 2009; Wang et al., 2013; Bera et al., 2016), optical flow (Zhou et al., 2011; Fang et al., 2016), and spatiotemporal features (Lu et al., 2013; Zhao et al., 2011). A detailed review of these methodologies is discussed in the next chapter.

However, this group of methods still have its limitations – they employ pre-defined features to represent the appearance and motion dynamics in the videos. These features may do well in a set of videos, but not applicable to another. This results in inconsistent performance of these methods and there is a need to redesign suitable features for different environments.

1.4 Research Objectives

This research aims to develop techniques for unusual event detection in videos, which are robust in the presence of the challenges listed in the previous section. This is achieved by addressing the challenges in the following objectives:

1. Extract useful features in videos of various environments: Though it may seem straightforward to use object trajectories to describe motion dynamics, it is very challenging to track all moving objects in a crowded scene due to occlusions and perspective distortions. As a result, it is impractical to extract object trajectories as the feature to represent events in videos. On the other hand, using pre-defined low-level features such as dynamic textures and optical flow may be optimal for one environment but sub-optimal for another. This research investigates the suitability of using learned features from training a deep neural network architecture such as spatiotemporal autoencoder for event representation, which is robust to various environments and requires no pre-defined features.

2. Eliminate tracking and grid-based processing: To overcome computational complexity in calculating low-level features, videos are usually sliced into small spatiotemporal windows, where feature vectors are computed for each window rather than the whole scene. This approach also helps to detect local events where instances occur within a small region. However, determination of the size of such a window is a challenge, especially in scenes with large crowd or perspective distortion. The development of an end-to-end technique provides a solution for fast processing without splitting each frame into small grids and meets the requirement of real-time detection.

3. Use unsupervised learning methods for unusual event detection: It is very tedious and impractical to label a large video dataset, i.e. marking bounding boxes and label each box with its corresponding type of event. With the unsupervised approach, a training dataset can be built by collecting footages of only normal events. The labour for filtering footages containing unusual events is relatively low compared to a fine level annotation. This research intends to develop techniques for unusual event detection in various environments with little to no prior knowledge while greatly

reducing labelling effort.

4. Avoid pre-determining the types (or the number of types) of unusual events since it is unknown beforehand: Some existing detection methods attempt to anticipate the types of unusual events, while some methods cluster the features into a pre-defined number of groups. These assumptions violate the fact that the number of possible events in any surveillance application is infinite. For practicality, this dissertation proposes a method which is independent of these assumptions.

1.5 Research Scope

This research investigates the capability of unsupervised learning method in capturing regular patterns in surveillance videos. It is assumed that abnormal events are instances that rarely or never occur in the training videos. Because no semantic context nor labelling is provided to our system, the proposed model is expected to learn regular spatiotemporal patterns from long training videos, and during inference, detect those instances which do not fit into our model as anomalies.

Although the proposed method does not rely on tracking, it does come with the limitation of non-tracking approaches. Since non-tracking approaches do not work in moving camera views, the proposed method will only be evaluated on video datasets captured from a fixed camera view.

1.6 Contributions

This dissertation presents a novel framework to represent video data by a set of general features, which are inferred automatically from a long video footage through a deep learning approach. Specifically, a deep neural network composed of a stack of convolutional autoencoders was used to process video frames in an unsupervised manner that captured spatial structures in the data,

which, grouped together, compose the video representation. Then, this representation is fed into a stack of convolutional temporal autoencoders to learn the regular temporal patterns.

The proposed method is domain free (i.e. not related to any specific task, no domain expert required), does not require any additional human effort, and can be easily applied to different scenes. To prove the effectiveness of the proposed method, the method is applied to real-world datasets and show that the proposed method consistently outperforms similar methods while maintaining a short running time. Part of the results in this dissertation was reported in Chong and Tay (2017).

1.7 List of Publications

1. Chong Y.S. and Tay Y.H., 2017. Abnormal event detection in videos using spatiotemporal autoencoder. In: Cong F., Leung A., Wei Q. (eds) Advances in Neural Networks - ISNN 2017. ISNN 2017. Lecture Notes in Computer Science, vol 10262. Springer, Cham.

2. Chong Y.S. and Tay Y.H., 2015. Modeling video-based anomaly detection using deep architectures: Challenges and possibilities. In: 10th Asian Control Conference (ASCC) 2015. pp. 1-8.

1.8 Dissertation Outline

The structure of the dissertation is as follows:

- Chapter 2 reviews prior works in the related domain and their influence on the proposed design.

- Chapter 3 presents the implementation details of the proposed method and the evaluation techniques.

- The experimental results and analysis on various datasets will be detailed in Chapter 4.

- Finally, the effectiveness and potential of the proposed method will be discussed in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

In real-world environments, video segments can be easily obtained provided access to the installed surveillance cameras. Segments of interest are those containing abnormal events. However, most of these abnormal instances are beforehand unknown, as this would require predicting all the ways something could happen out of the norm. It is therefore simply impossible to learn a model for all that is abnormal or irregular. How can we find an anomaly without knowing what to look for?

An anomaly is defined as something that deviates from what is standard, normal, or expected. In terms of probability, anomalies are events of low probability with respect to a probabilistic model of regular events. In the context of videos, anomalies are unusual events that occur very rarely or never occurred in the entire video sequence. Since abnormal events rarely or never occur, the effort of filtering out segments containing abnormal events is relatively little compared to annotating every instance in long videos with its corresponding event label.

Since it is easier to get video data where the scene is normal in contrast to obtaining what is abnormal, we could focus on a setting where the training data contains only normal visual patterns. A popular approach adopted in this area is to first learn the normal patterns from the training videos, then abnormal events are detected as those deviated from the normal patterns. The majority of the work on abnormal event detection relies on the extraction of local features from videos, that are then used to train a normalcy model.

The general process of automatic event detection is depicted in Figure 2.1. Videos are partitioned into a training set and a testing set. The testing set

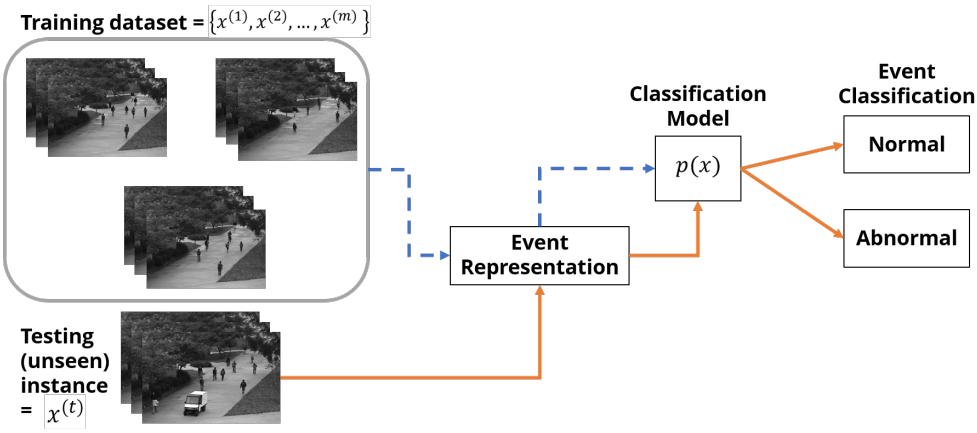


Figure 2.1: Overview of the process of automatic surveillance event detection. Dashed line arrows refer to the training workflow, while solid line arrows outline the evaluation process.

is later used to evaluate the performance of the system. Features are extracted from the training set to represent events. An effective feature descriptor should be able to distinguish the abnormal events from normal events. Next, a normalcy model is built to associate high probabilities with feature vectors which represent normal events. Then, at test time, if the probability computed based on the normalcy model for the unseen video segment is higher, the segment is then classified as containing abnormal event(s).

The success of an event detection system depends on the effectiveness of solving two fundamental problems: (a) the design of feature descriptor to represent the event, and (b) the design of the classifier or model to detect the event.

2.1 Event Representation

This section reviews the literature on addressing the first problem – the design of feature descriptor to represent surveillance events effectively.

2.1.1 Event Representation by Trajectory Features

Following the definition by Ko (2008), trajectories are paths derived from the location of particular points of an object in time. The generation of motion trajectories from a sequence of images typically involves the detection of tokens in each frame and the correspondence of such tokens from one frame to another.

Trajectories have long been popular in video analysis and anomaly detection (Zhou et al., 2015; Li et al., 2011; Piciarelli et al., 2008; Mo et al., 2014) because they are relatively easy to extract and it is straightforward to interpret. Trajectories are effective at capturing the global structure of object motions through accurate long-term observation (Zhou et al., 2011). Typically, trajectories are generated by tracking foreground objects in the scene after performing background subtraction.

In a work by Piciarelli et al. (2008), the trajectories are subsampled to a vector representation and clustered with a one-class SVM to form the feature space containing the normal trajectories. If a new trajectory falls outside the computed hypervolume, it is identified as abnormal.

Mo et al. (2014) applied sparse reconstruction techniques to learn a dictionary of normal trajectories. At the evaluation phase, any new trajectories that could not be well reconstructed using few bases from the learned dictionary would be detected as an anomaly. A similar approach is adopted by Li et al. (2011).

Recently, there are also methods proposed using tracklets (Zhou et al., 2011). Tracklets are fragments of a complete trajectory. They terminate when occlusions and scene clutter arise. According to (Zhou et al., 2011), they are more conservative and less likely to drift than long trajectories. In Mousavi et al. (2015), statistics of the tracklets obtained are computed and the latent

Dirichlet allocations (LDA) generative model is employed for modelling and classification of events.

However, the accuracy of trajectory analysis relies heavily on tracking, which precise tracking still remains a significant challenge in computer vision, particularly in complex situations. To extract accurate spatiotemporal information from trajectories, a fixed view is assumed and any camera movement will result in inaccurate or broken trajectories. Due to these limitations, tracking-based approaches can be applied to scenes with less clutter but are ineffective for detecting unusual patterns in a crowded or complex scenario.

2.1.2 Event Representation by Local Features

To overcome the limitations of relying on motion features extracted from trajectories, a number of studies have been focused on analysing each frame at either the pixel or the region level by dividing each frame into smaller patches.

Unlike trajectory-based features, local features do not rely on tracking. They rely mainly on extracting and analyzing low-level visual features, such as the histogram of oriented gradients (Xiao et al., 2015), the histogram of oriented flows (Laptev et al., 2008) and optical flow (Reddy et al., 2011), by employing spatiotemporal video volumes through dense sampling and interest point selection (Dollár et al., 2005). These approaches perform better in cluttered scenarios, when compared to tracking-based methods.

For example, in the work of Lu et al. (2013), the authors use multiple scales of the same frame, split them into small patches, extract low-level features from each patch, and then learn the combinations that build up the normal pattern. To capture local features from coarse- to fine-level, the authors extract features from a spatial pyramid as shown in Figure 2.2.

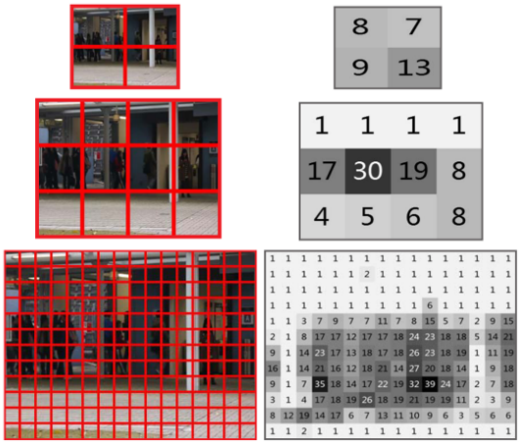


Figure 2.2: A spatial pyramid for local feature extraction. The numbers in the figure refer to the number of combination required to describe each patch location. The image is adopted from Lu et al. (2013).

In the work of Zhong et al. (2004), the authors extract short sequences from the video where each sequence extracted are assumed to contain only one event. After performing background subtraction, they compute a spatial histogram by blocks containing object motion. This method involves tedious manual preprocessing and also unable to locate the anomalous region. Han et al. (2013) adopted a feature descriptor called Multi-scale Histogram of Optical Field (MHOF) (Cong et al., 2011), which not only describes motion direction but also preserves more motion-energy information.

Cong et al. (2013) proposed a region-based descriptor called "Motion Context" to describe both motion and appearance information of the spatio-temporal segment. The author uses Edge Orientation Histogram (EOH) as appearance descriptor and Multi-layer Histogram of Optical Flow (MHOF) as motion descriptor. Then for each queried spatiotemporal segment, it searches for its best match in the training dataset, and determine the normality using a dynamic threshold. This method is more efficient compared to their previous work using the sparse reconstruction method.

A similar category of descriptors is spatiotemporal video volume descriptors (HOG3D) (Kläser et al., 2008): these volumes are characterised by

the histogram of the spatio-temporal gradient in polar coordinates (Roshtkhari and Levine, 2013). In Lu et al. (2013), 3D gradient features of each spatiotemporal cube are extracted from the video sequence and trained to obtain sparse combinations with allowable reconstruction errors.

2.1.3 Event Representation by Learned Features

The problem of how to represent video sequences is the most fundamental problem in surveillance event detection. Instead of introducing the increasingly more complex handcrafted features, recent researches have now moved to use efficient and robust algorithms that learn to extract feature representations from images and videos in a fully unsupervised manner. There are several existing methods for representing images and video sequences using learned features from raw pixel values and frames.

The success of deep learning methods in various applications consequently caused the rise of such methods in anomaly detection. The term deep learning refers to learning a hierarchical set of features through multiple layers of hidden nodes in an artificial neural network. Unlike previously stated methods, there is no need to define a specific set of features to extract from the dataset – deep learning methods learn the useful features directly from the data with minimal preprocessing.

Specifically, convolutional neural networks (ConvNet) have proved its effectiveness in a wide range of applications such as object recognition (Simonyan and Zisserman, 2014b), person detection (Vu et al., 2015), and action recognition (Tran et al., 2015; Simonyan and Zisserman, 2014a). ConvNet consists of a stack of convolutional layers with a fully-connected layer and a softmax classifier, and convolutional autoencoder is essentially a ConvNet with its fully-connected layer and classifier replaced by a mirrored stack of convolutional layers.

Sabokrou et al. (2015) introduce a global anomaly detection framework based on features learned from an autoencoder. Stacked denoising autoencoders are used by Xu et al. (2017) to learn motion and appearance feature representations. Ravanbakhsh et al. (2016) propose a Binary Quantization Layer plugged as a final layer on top of a ConvNet, which represents temporal motion patterns for the task of abnormality segmentation. Hasan et al. (2016) used a convolutional autoencoder to learn normal activity patterns from raw pixels. On the other hand, a convolutional variant of long-short term memory (LSTM) architecture is employed in a work by Medel (2016) to capture the spatial and temporal dynamics of normal events in videos. Their reported result proves the usefulness of learned representation on videos through a stack of neural hidden layers.

Despite its simplicity, some limitations remain in these recently proposed methods. Though 3D ConvNet performed excellently in learning discriminative features between the anomalies and the normal events, it is impractical to apply in real-world scenarios due to the absence of video segments containing abnormal events. Meanwhile, in the convolutional autoencoder proposed by Hasan et al. (2016), convolution and pooling operations are performed only spatially, even though the proposed network takes multiple frames as input, because of the 2D convolutions, after the first convolution layer, temporal information is collapsed completely (Tran et al., 2015).

For a comprehensive review of video features, interested readers can refer to Chong and Tay (2015).

2.1.4 Section Summary

Table 2.1 summarises the advantages, limitations and example applications of each feature category.

Table 2.1: Summary of three categories of video features covered in Section 2.1.

Type of Features	Trajectories	Local Features	Learned Features
Description	Track the location of particular points of an object in time to obtain the motion path of the object.	Low-level visual features extracted through dense sampling and interest point selection.	Extract features from abundant data without explicitly defining a specific set of features.
Advantages	• Able to capture the global structure of object motions • Easy to interpret	• Able to work in cluttered scenarios • Does not rely on tracking • Background subtraction is not compulsory	• Does not rely on tracking and background subtraction • Does not require the selection of the type of features • Does not require splitting into regions
Limitations	• Require background subtraction • Require precise object tracking • Sensitive to occlusions	• Require splitting into multiple regions and subsequently perform summarisation from each region • Require careful selection of the type of local features to be extracted	• Learned features are difficult to interpret • Tuning a large set of hyperparameters can be tedious and time-consuming
Applications	Zhou et al. (2015); Li et al. (2011); Piciarelli et al. (2008); Mo et al. (2014)	Lu et al. (2013); Zhong et al. (2004); Cong et al. (2013); Klaser et al. (2008)	Sakurada and Yairi (2014); Hasan et al. (2016); Chalapathy et al. (2017)

2.2 Classification Model

Once feature vectors have been extracted to represent the events, these feature vectors are fed as the input to a classification model for event detection. Some commonly applied event detection models have already been mentioned in Section 2.1.1 and 2.1.2. In this section, the literature is grouped into two categories: methods based on supervised learning, and unsupervised approaches.

2.2.1 Supervised Approach

Today, the supervised approach is by far the more common across a wide range of industry use cases because it is usually fast and accurate. Input videos are properly annotated with its event type and fed into a supervised model to learn the mapping between the video features and the corresponding event label. These methods build explicit models of normal and abnormal behaviour based on the labelled data. This approach assumes prior knowledge of both normal and abnormal events are available.

In the event where prior knowledge is available, for instance, in traffic surveillance, having a certain level of supervision in the surveillance system proved to perform well (Inoue et al., 2011; Yuan et al., 2009). Among the popular architectures used to classify video features are artificial neural networks (ANN) and support vector machines (SVM).

Inspired by biological nervous systems, an artificial neural network consists of an interconnected network of artificial neurons. Neural networks become widely used in solving nonlinear problems such as prediction, pattern recognition, and function optimisations. For classification tasks, an activation function (usually sigmoid or softmax) is applied to the final layer which consists of the number of classes to classify.

Support Vector Machines (SVM) is also a popular choice for many classification problems. The goal of SVM is to find the optimal separating hyperplane in a feature space (Lauer et al., 2007). By specifying different kernel functions as the decision function, SVM can be versatile to cope with various scenarios encountered in real-world applications. SVM was initially designed to solve binary classification problems, but can be extended to perform multi-class classifications.

For example, Kläser et al. (2008) use non-linear support vector machines with chi-square kernel to learn multi-class event classification based on histograms of visual word occurrences. Rajesh et al. (2013) applied a neural network to classify moving vehicles in their proposed traffic surveillance system. In a more narrowly defined task such as abandoned object detection, supervised approach is also suitable – Tian et al. (2011) defined a set of rules that works together with background subtraction technique and person detection model to detect abandoned objects in a scene.

Generally, the supervised approach is good for abnormal event detection if the abnormal events are well defined and there are sufficient anomalous examples. However, supervised techniques have these disadvantages according to Numenta (2015):

- Most supervised models are not very adaptive to pattern changes. To learn new data patterns, a new model would need to be trained with labelled data. Thus, these models are not suitable for real-time or streaming data.
- The labelling process involves tedious human effort and must be repeated periodically for new data examples.
- The labelling process can be error-prone and mistakes in labelling can cause poor model performance.

- It is difficult to anticipate rare or unseen anomalies, thus even more challenging to model these anomalies with supervised methods.

2.2.2 Unsupervised Approach

Due to the nature that anomaly is unbounded and cannot be anticipated, the most widely adopted models are based on unsupervised learning. In unsupervised learning, there is no labelled dataset and since there are no annotations to guide the learning process, researchers have designed alternative objective functions to serve as the teaching signal.

Since abnormal events are rare and unknown to an unsupervised model, normal events can be utilized to build a normality model. Then a test sequence can be considered abnormal if the probability of generating the sequence from the normality model is very low.

One of the popular architectures based on this assumption is Hidden Markov Models (HMMs). The hidden states and transitions in HMMs are used to model temporal dependencies among components (Gupta et al., 2014). The Viterbi algorithm is applied to extract the most probable sequence.

Kratz and Nishino (2009) propose a method to model statistics of spatio-temporal gradients with a coupled HMM. Jiang et al. (2009) propose a method that models trajectories as HMMs and clusters them into groups of HMMs that are similar in terms of the distributions they represent. A similarity metric between HMMs is designed to compute the distance of each trajectory to the clusters.

Jiang et al. (2011) use HMM for co-occurrence anomalies. In Saligrama et al. (2010), Markov Random Fields (MRFs) are incorporated for spatial co-occurring anomalies. Similarly, the authors in Cui et al. (2007) proposed to model events as HMMs and determine the posterior probability of an observation given past events using a Sequential Monte Carlo framework.

Although HMMs benefit from relatively fast and powerful training and decoding algorithms, it does not model high-dimensional datasets efficiently. To model k bits of information, it needs 2^k hidden states. Hence, HMMs do not scale well to real life datasets.

Another group of graphical models that is capable of modelling co-occurring events is probabilistic topic models (Blei, 2012), including its various extensions such as latent Dirichlet allocation (LDA), probabilistic latent semantic analysis (pLSA), and hierarchical Dirichlet process (HDP). Originally, these generative models were applied to automatic text analysis but also proved effective to analyse video data. By training a generative model with normal events and learning a set of discriminative features which can describe normal events well, then abnormal events can be detected as those that are badly explained by the trained model.

In topic models, documents are random mixtures over latent topics, where each topic is represented by a distribution over words. Similarly, video clips of different action categories can be represented by a distribution over visual words. Abnormal events are characterized by low word-topic probabilities or having visual words from normal topics, but co-occurring in an unusual and unique combination (Popoola and Wang, 2012).

Mousavi et al. (2016) computed a tracking-based feature – histogram of oriented tracklets on 3D video patches and applied LDA to learn normal behavioural patterns in crowded scenarios. An example of using non-tracking based feature is presented in the work of Hu et al. (2016) which uses pixel gradient features as the input to their LDA model.

Varadarajan and Odobez (2009) describe each activity by location, motion, and size features and use these features to form the visual words. By counting for each video clip (document) the number of times a word occurs

in it, a bag-of-words representation is obtained. The obtained representation is then used to compute the log-likelihood of the words in the document.

Though powerful, topic models suffer from a few limitations. In a document, the words are modelled as independent of each other. Ignoring the correlations of words in a document may not fit reality. Furthermore, to facilitate the computation of probability, the high-dimensional video data are often summarised into statistics that may not be representative of the events, which may cause potential inaccuracies.

Besides HMM and topic models, a widely adopted normality model is a Gaussian Mixture Model (GMM). A GMM is a probabilistic model that assumes the data is drawn from a mixture of Gaussian distributions. It is flexible and capable of approximating multi-modal distributions. One can train a GMM using the expectation-maximization (EM) algorithm. Once it is trained with normal events, the probability of test patterns can be computed and abnormal events likely to be associated with low likelihood values (Roshtkhari and Levine, 2013; Basharat et al., 2008; Sabokrou et al., 2015).

However, when there is a cluster or mixture lacking enough samples, the covariance matrix may become singular, leading the algorithm to diverge. This is usually overcome by performing dimension reduction or regularising the covariances artificially. Also, to determine the optimal number of mixtures, information theoretical criteria are required to aid the decision. Therefore, GMMs are limited to applications where the feature vectors are low dimensional compared to the number of training samples.

Another similar technique is sparse reconstruction (Cong et al., 2011; Zhao et al., 2011). The underlying principle of these methods is that any new feature representation of a normal/anomalous event can be approximately represented as a (sparse) linear combination of feature representations (of previously

observed events) in a learned dictionary. This assumes that all previously observed events are normal events. Because the basis function of the dictionary is trained on normal events, the reconstruction cost is expected to be high for abnormal motion patterns. By thresholding the reconstruction cost, abnormal events can be detected. The idea of sparse reconstruction is applied across many papers (Lu et al., 2013; Cong et al., 2011; Xiao et al., 2015) due to the ease of implementation and low computational cost. It is also possible to include some a priori knowledge about the application by adding weights to selected features.

However, since classical bags of visual words approaches group similar volumes, all compositional information are destroyed in the process of grouping visual words. It is also required to pre-determine the number of clusters (Hu et al., 2016), which can only be found through trial-and-error during testing time. Though some methods proposed to automatically determine the suitable number of clusters or dictionary atom (Lu et al., 2013), it requires repeating the training process which increases the training time drastically. In addition, codebook models require searching over a large space (Roshtkhari and Levine, 2013) even during the time of testing, making it impractical for real-time anomaly detection.

As an alternative way to classify instances in the presence of data from only a single class (in this context, the normal class), a one-class classifier such as one-class SVM has been extensively used for anomaly detection problems. For instance, Ma et al. (2015) and Xu et al. (2017) used one-class SVM to learn the boundary enclosing normal patterns from extracted features. Also, a similar one-class approach named space-time Markov random field (MRF) was devised by Kim and Grauman (2009).

Most of the previous work uses hand-crafted features to model normal activity patterns. Following the success of deep learning applications in the field of computer vision (Simonyan and Zisserman, 2014b; Vu et al., 2015; Tran

et al., 2015; Simonyan and Zisserman, 2014a), a few authors (Hasan et al., 2016; Medel, 2016) were inspired to use deep-learning based approach to learn normal patterns from raw pixels. In these end-to-end approaches, there is no need to address the design of feature extractor and event detection model separately.

Hasan et al. (2016) used an end-to-end convolutional autoencoder to detect anomalies in surveillance videos. The idea of using autoencoders for anomaly detection task is also presented in the work of Sakurada and Yairi (2014) and Chalapathy et al. (2017).

On the other hand, long short term memory (LSTM) model is well-known for learning temporal patterns and predicting time series data. Medel (2016) has recently proposed to apply convolutional LSTMs for learning the regular temporal patterns in videos and his findings show great promise of what deep neural network can learn. However, convolutional LSTM layers applied by Medel (2016) are memory-intensive – the training will need to be executed on very small mini-batches, which results in slow training and testing time.

2.2.3 Section Summary

Table 2.2 summarises the characteristics, where it is suitable, representative models and example applications of each models' category.

2.3 Chapter Summary

This chapter reviewed the literature on addressing two important parts of the abnormal video event detection problem: 1) designing feature descriptor that represent surveillance events effectively, 2) applying these features to classification models for event detection. Applying research insights from this chapter, the implementation details of the proposed method will be discussed in the following chapter.

Table 2.2: Summary of two categories of classification models covered in Section 2.2.

Type of Approach	Supervised Approach	Unsupervised Approach
Characteristics	Learn the mapping between the video features and the corresponding event label.	Build a probabilistic model of normal events by fitting video features of normal events during training, then determine the probability of generating a test sequence from the model.
When it is suitable	• Normal and abnormal events are clearly defined • Prior knowledge of both normal and abnormal events are available • Has labelled video data of both categories	• Type of events cannot be clearly defined nor anticipated • Difficult to obtain a labelled dataset • The event category of interest (i.e. abnormal events) can be detected probabilistically
Representative models	• Artificial Neural Network (ANN) • Support Vector Machines (SVM) • Gaussian Mixture Models (can be unsupervised too)	• Hidden Markov Models (HMM) • Probabilistic Topic Models • One-class SVM • Sparse reconstruction
Applications	• Human action classification (Klaser et al., 2008) • Vehicle classification and stopped vehicle detection (Rajesh et al., 2013) • Abandoned object detection (Tian et al., 2011)	• Time series analysis and outlier detection (Gupta et al., 2014) • Anomaly detection in extremely crowded scenes (Kratz and Nishino, 2009) • Prediction of video pixels (Medel, 2016)

CHAPTER 3

METHODOLOGY

The proposed method described in this chapter is based on the assumption that when an abnormal event occurs, these frames of video would contain appearance and/or motion patterns which are significantly different from the normal footages. Inspired by Hasan et al. (2016), we train an end-to-end model that consists of a spatial feature extractor and a temporal encoder-decoder which together learns the temporal patterns of the input volume of frames. The model is trained with video volumes consists of only normal scenes, with the objective to minimize the reconstruction error between the input video volume and the output video volume reconstructed by the learned model. After the model is properly trained, normal video volume is expected to have low reconstruction error, whereas video volume consisting of abnormal scenes is expected to have high reconstruction error. By thresholding on the error produced by each testing input volumes, our system will be able to detect when an abnormal event occurs.

Our proposed model takes an input of $T-1$ frames captured at time $t = 1, 2, ..., T-1$, and output the prediction of frame $t = T$. A general overview of the proposed architecture is presented in Figure 3.1.

The proposed approach consists of three main stages: preprocessing, feature learning, and anomaly detection. Each stage is detailed in the following sections.

3.1 Preprocessing

The task of this stage is to convert raw data to the aligned and acceptable input for the model. Each frame is extracted from the raw videos and resized to 224×224. Then the images are converted to grayscale to reduce dimensionality.

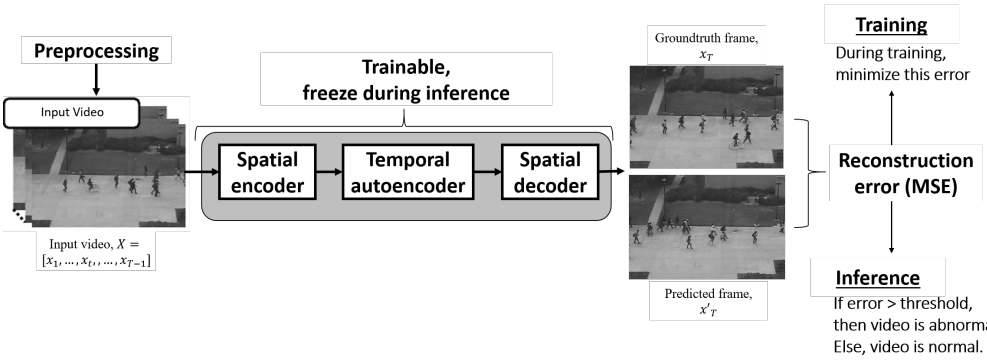


Figure 3.1: Our proposed network architecture. It takes a sequence of length $T-1$ as input, and output a reconstruction of the consequent frame of the input sequence. The spatial encoder takes one frame at a time as input, after which $T-1$ frames have been processed, the encoded features of $T-1$ frames are concatenated and fed into temporal encoder for motion encoding. The output of the model is the T-th frame of the input sequence.

To ensure that the input images are all on the same scale, the pixel values are scaled between 0 and 1 and subtracted every frame from its global mean image for normalisation. The mean image is calculated by averaging the pixel values at each location of every frame in the training dataset. After that, $T-1$ consecutive frames are concatenated into video volumes, while setting the T-th frame as the groundtruth for training.

3.2 Feature Learning

We propose a convolutional spatiotemporal autoencoder to learn the regular patterns in the training videos. Our proposed architecture consists of two parts — spatial autoencoder for learning spatial structures of each video frame, and temporal encoder-decoder for learning temporal patterns of the encoded spatial structures. As illustrated in Figure 3.2, the spatial encoder and decoder have two convolutional and deconvolutional layers respectively, while the temporal encoder is a three-layer convolutional long short term memory (LSTM) model. Convolutional layers are well-known for its superb performance in object recognition, while LSTM model is widely used for sequence learning and time-series modelling and has proved its performance in applications such as

Layer	Kernel size	Number of filters	Stride	Output size
Input	-	-	-	(T-1, 224, 224, 1)
Convolution	(11, 11)	128	2	(T-1, 112, 112, 128)
Convolution	(5, 5)	64	2	(T-1, 56, 56, 64)
ConvLSTM	(3, 3)	64	1	(T-1, 56, 56, 64)
ConvLSTM	(3, 3)	32	1	(T-1, 56, 56, 32)
ConvLSTM	(3, 3)	64	1	(56, 56, 64)
Deconvolution	(5, 5)	128	2	(112, 112, 128)
Deconvolution	(11, 11)	1	2	(224, 224, 1)

Figure 3.2: Layer configuration of the proposed network architecture. The numbers at the rightmost denote the output size of each layer.

speech translation and handwriting recognition.

Generally, in deciding what parameters to use in our proposed model, we look for similar problems and deep learning architectures which have already been shown to work. Then a suitable model can be developed by experimentation. We follow the kernel size as implemented in Hasan et al. (2016) to use 11×11, 5×5, and 3×3 for convolutional layers. However, we replaced the pooling layer with a convolutional layer with increased stride. The intention of pooling is to reduce the number of extracted features and to avoid overfitting. Springenberg et al. (2014) and Zagoruyko and Komodakis (2016) have proved that max-pooling layers can be replaced by convolutional layers with increased stride without loss in accuracy. Besides, we used a smaller number of filters as compared to Hasan et al. (2016) because applying a large number of filters quickly filled up the whole memory due to the extra time dimension. We had to gradually reduce the number of filters until the model can fit into the memory constraint (64GB RAM).

3.2.1 Autoencoder

An autoencoder is an artificial neural network which made up of two components: an encoder and a decoder. It is commonly used for dimensionality reduction by compressing the input to a smaller number of encoder nodes. Similar to ANNs, autoencoders are usually trained using back-propagation but in

an unsupervised manner, by minimizing the reconstruction error of the decoded output from the original inputs. With a nonlinear activation function, an autoencoder can learn more useful features than some common linear transformation methods such as PCA.

3.2.2 Spatial Convolution

The main purpose of convolution in the context of a convolutional network is to extract features from the input image. Convolution preserves the spatial structure by learning image features from small grids of input data. Mathematically, a convolution operation performs dot products between the filters and local regions of the input. Suppose that a convolutional layer is set up and preceded by a $n \times n$ square input layer. If a $m \times m$ filter W is used, then the convolutional layer output will be of size $(n - m + 1) \times (n - m + 1)$.

A convolutional network learns the values of each filter from its training data, with pre-defined parameters such as the number of filters, filter size, the number of layers prior to training. With more number of filters, more features get extracted and the better the model becomes at recognizing patterns in novel instances. However, more filters would add to computational complexity, so we need to find balance by not setting the number of filters too large.

3.2.3 Recurrent Neural Network (RNN)

In a traditional feedforward neural network, all inputs (and outputs) are assumed to be independent of each other. However, learning temporal dependencies between inputs are important in tasks involving sequences, for example, a word predictor model should be able to derive information from the past inputs. RNN works just like a feedforward network, except that the values of its output vector are influenced not only by the input vector but also on the entire history of inputs. Theoretically, RNNs can capture temporal information in ar-

bitrarily long sequences, but due to vanishing gradients, they are often limited to looking back only a few steps.

3.2.4 Long Short Term Memory (LSTM)

To overcome this problem, a variant of RNN is introduced: long short term memory (LSTM) model which incorporates a recurrent gate called forget gate. With the new structure, LSTMs prevent backpropagated errors from vanishing or exploding, thus can work on long sequences and they can be stacked together to capture higher level information. The formulation of a typical LSTM unit is summarised in Figure 3.3 and equations 3.1 through 3.6.

$$f_t = \sigma(W_f \otimes [h_{t-1}, x_t] + b_f) \tag{3.1}$$

$$i_t = \sigma(W_i \otimes [h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\hat{C}_t = \tanh(W_C \otimes [h_{t-1}, x_t] + b_C) \tag{3.3}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \hat{C}_t \tag{3.4}$$

$$o_t = \sigma(W_o \otimes [h_{t-1}, x_t] + b_o) \tag{3.5}$$

$$h_t = o_t \otimes \tanh(C_t) \tag{3.6}$$

Equation 3.1 represents the forget layer, equation 3.2 and 3.3 are where new information is added, 3.4 combines old and new information, whereas equation 3.5 and3.6 output what has been learned so far to the LSTM unit at the next timestep. The variable x_t denotes the input vector, h_t denotes the hidden state,

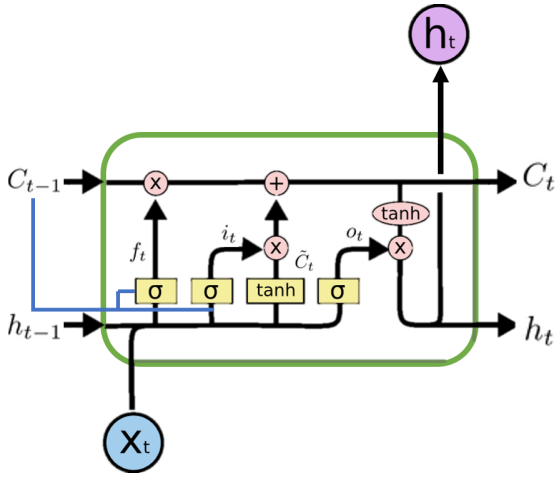


Figure 3.3: The structure of a typical LSTM unit. The blue line represents an optional peephole structure, which allows the internal state to look back (peep) at the previous cell state C_{t-1} for a better decision. Best viewed in colour.

and C_t denotes the cell state at time t. W are the trainable weight matrices, b are the bias vectors, and the symbol $\otimes$ denotes the Hadamard product.

3.2.5 Convolutional LSTM

A variant of the LSTM architecture, namely Convolutional Long Short-term Memory (ConvLSTM) model was introduced by Shi et al. (2015) and has been recently utilised by Patraucean et al. (2016) for video frame prediction. Compared to the usual fully connected LSTM (FC-LSTM), ConvLSTM has its matrix operations replaced with convolutions. By using convolution for both input-to-hidden and hidden-to-hidden connections, ConvLSTM requires fewer weights and yield better spatial feature maps. The formulation of the ConvLSTM unit can be summarised with equations 3.7 through 3.12.

$$f_t = \sigma(W_f * [h_{t-1}, x_t, C_{t-1}] + b_f) \tag{3.7}$$

$$i_t = \sigma(W_i * [h_{t-1}, x_t, C_{t-1}] + b_i) \tag{3.8}$$

$$\hat{C}_t = \tanh(W_C * [h_{t-1}, x_t] + b_C) \tag{3.9}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \hat{C}_t \tag{3.10}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t, C_{t-1}] + b_o) \tag{3.11}$$

$$h_t = o_t \otimes \tanh(C_t) \tag{3.12}$$

While the equations are similar in nature to equations 3.1 through 3.6, the input is fed in as images, while the set of weights for every connection is replaced by convolutional filters (the symbol $*$ denotes a convolution operation). This allows ConvLSTM work better with images than the FC-LSTM due to its ability to propagate spatial characteristics temporally through each ConvLSTM state. Note that this convolutional variant also adds an optional 'peephole' connections to allow the unit to derive past information better.

3.3 Activation Functions

In neural networks, the activation function of a node defines the output of that node given an input or set of inputs. Activation functions play a key role in introducing non-linearity into a neural network. This allows a response variable to vary non-linearly with its input variables.

3.3.1 Rectified Linear Unit (ReLU)

ReLU is a simple element-wise activation function $f(x) = \max(0, x)$ thresholded at zero. Before ReLU was introduced by Nair and Hinton (2010), sigmoid and $\tanh$ were popular choices among researchers working with neural

networks. Sigmoid function is given by $f(x) = 1/(1+e^{-x})$, while $\tanh$ function is defined as $f(x) = (e^x - e^{-x})/(e^x + e^{-x})$.

With the rise of deeper neural networks, the problem of vanishing gradient arises and this decreases the learning capability of deeper layers. ReLU helps to avoid vanishing gradients because of the output of ReLU is not bounded at one like sigmoid does. The constant gradient of ReLUs also results in faster learning. Krizhevsky et al. (2012) state that ReLU train six times fast than $\tanh$ to reach same training error.

A normalisation technique that works well with ReLUs is batch normalisation. Batch normalisation (BN) is introduced by Ioffe and Szegedy (2015) to address the internal covariance shift phenomenon by normalising layer inputs to zero mean and unit variance. Applying this technique to neural networks make the network less sensitive to initialization and can be trained with higher learning rates. It also acts as a regularizer like Dropout (Srivastava et al., 2014). ReLU + BN combination is used extensively in Residual Network developed by He et al. (2016) which won ILSVRC 2015.

3.4 Optimizers

Gradient descent is a method to minimize an objective function $J(\theta)$ parameterized by a model's parameters $\theta \in \mathbb{R}^d$ by updating the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$ with respect to the parameters (Dangeti, 2017). The learning rate α determines the step size required to reach a local minimum.

The stochastic gradient descent (SGD) algorithm (Ng, 2000) updates the parameters θ of the objective $J(\theta)$ for each training example $x^{(i)}$ and label $y^{(i)}$ as,

$$\theta = \theta - \alpha \nabla_\theta J(\theta; x^{(i),y^{(i)}}).$$

Due to SGD performing one update at a time, the high variance in each update causes the objective function to fluctuate heavily. To overcome this, SGD performs updates with respect to minibatch to reduce the variance and thus leads to more stable convergence. Also, the data needs to be randomly shuffled prior to each epoch of training to avoid bias in the gradient and lead to poor convergence.

Though SGD is fairly simple to implement, it does not guarantee good convergence (Ruder, 2016), and has a few disadvantages:

- A number of hyperparameters are required to tuned, such as the regularization parameter and the number of iterations.
- A learning rate that is too small results in slow convergence, while a learning rate too large can cause the loss function to fluctuate or lead to divergence.
- Learning rate schedules which controls the learning rate are pre-defined before training hence unable to adapt to different datasets.

3.4.1 Adaptive Moment Estimation (Adam)

To address some of the limitations of SGD, we propose to use algorithms that adapt the learning rate to the model parameters. Adam (Kingma and Ba, 2014) is a method that calculates adaptive learning rates for each parameter. Besides storing past squared gradients v like RMSprop does, Adam also keeps a history of the past gradients m:

$$g = \nabla_\theta f(\theta)$$

$$m = \beta_1 m + (1 - \beta_1) g$$

$$v = \beta_2 v + (1 - \beta_2) g^2$$

During the initial time steps, as m and v are initialised as vectors of 0's, they are biased towards 0, especially when the decay rates are small. In an attempt to counteract these biases, the first and second moment estimates are calculated as follows:

$$\hat{m} = \frac{m}{1 - \beta_1}$$
$$\hat{v} = \frac{v}{1 - \beta_2}$$

Finally Adam updates parameters as follows:

$$\theta = \theta - \frac{\alpha}{\sqrt{\hat{v}} + \epsilon}\hat{m}$$

The authors of the algorithm propose default values of 0.9 for β_1, 0.999 for β_2, and 10^{-8} for ϵ. Adam was shown to work well in practice and compares favourably to other adaptive learning-based algorithms.

3.5 Reconstruction Error and Regularity Score

The reconstruction error of all pixel values I in frame t of the video sequence is taken as the Euclidean distance between the input frame and the reconstructed frame:

$$e(t) = ||x(t) - f_W(x(t))||_2 \tag{3.13}$$

where f_W denotes the weights in the spatiotemporal model.

Once the model is trained, we can evaluate our model's performance by feeding in testing data and check whether it is capable of detecting abnormal events while keeping false alarm rate low. To better compare with Hasan et al. (2016), we used the same formula to calculate the regularity score for all frames, the only difference being the learned model is of a different kind.

Based on the reconstruction error, we then compute the abnormality score $s_a(t)$ by scaling the error values between 0 and 1. Subsequently, regularity score $s_r(t)$ can be simply derived by subtracting abnormality score from 1:

$$s_a(t) = \frac{e(t) - e(t)_{\min}}{e(t)_{\max}} \tag{3.14}$$

$$s_r(t) = 1 - s_a(t) \tag{3.15}$$

3.6 Anomaly Detection

It is straightforward to determine whether a video frame is normal or anomalous. The reconstruction error of each frame determines whether the frame is classified as anomalous. The threshold determines how sensitive we wish the detection system to behave — for example, setting a low threshold makes the system become sensitive to the happenings in the scene, where more alarms would be triggered.

3.7 Evaluation metrics

This section introduces the evaluation metrics used in this study. Section 3.7.1 will first introduce the Receiver Operating Characteristic (ROC) Curve, which is a fundamental evaluation tool for anomaly detection. A complete evaluation also requires the annotation of ground truth. Then in Section 3.7.2, we discuss how the detected event count for each dataset is computed based on a post-processing technique.

3.7.1 Receiver Operating Characteristic (ROC) Curve

We obtain the true positive rate (TPR) and false positive rate (FPR) by setting at different error threshold obtained from Section 3.6 in order to calculate

the area under the receiver operating characteristic (ROC) curve (AUC). The equal error rate (EER) is obtained when the false positive rate equals to the false negative rate. The larger AUC indicates a better system. Because the range of FPR and TPR are [0, 1], the range of AUC is also [0, 1]. A perfect ROC will yield an AUC of 1. In the case of anomaly detection, a low EER also indicates a better performing system, since it implies that the system is able to detect more abnormal events at a low false alarm rate.

3.7.2 Anomalous Event Count

Following the practice in Hasan et al. (2016), to reduce the noisy and unmeaningful minima in the regularity score, we group local minima with a fixed temporal window of 50 frames. We assume local minima within 50 frames belong to the same abnormal event. This is a reasonable length of the temporal window as an abnormal event should be at least 2-3 seconds long to be meaningful (videos are captured at 24-30 fps).

3.8 Chapter Summary

This chapter discussed the three stages of our proposed approach. We have also explained the justifications of the techniques and parameters we applied in our proposed model. By applying these techniques, the effectiveness of these techniques will be shown experimentally in the following chapter. In the next chapter, the datasets we used to evaluate our method and the results obtained will be discussed in detail.

CHAPTER 4

EXPERIMENTS AND DISCUSSIONS

4.1 Datasets

We train our model on the three most commonly used benchmarking datasets: UMN [1], Avenue (Lu et al., 2013), and UCSD Ped1 and Ped2 (Mahadevan et al., 2010). All videos are taken from a fixed position for each dataset. All training videos contain only normal events. Testing videos have both normal and abnormal events.

All three datasets have different activity contexts and definitions of normal and abnormal events. In general, we define normal activities as those occur frequently in each dataset, and abnormal activities are those fall outside the norm. Thus, normal activities can differ across datasets, as the contexts are different.

There are two major categories of anomalous events: global and local anomalous events. Global anomalous events are defined as those events which abnormalities involve the whole scene, where local anomalous events are those which only a portion of the scene contains some abnormal activities. A standard video dataset for global anomalies is the UMN dataset. Avenue and UCSD Pedestrians are benchmarking video datasets for local anomaly detection.

4.1.1 UMN

The UMN dataset is captured from 3 different scenes, including indoor and outdoor scenes. The footage from each scene has a different duration, and each contains several clips. The total number of frames is 7739. We set aside 60% of all frames for training, while leaving the rest for testing.

[1] http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi

4.1.2 Avenue

In Avenue dataset, there is a total of 16 training and 21 testing video clips. Each clip's duration varies between less than a minute to two minutes long. The normal scenes consist of people walking between the staircase and subway entrance, whereas the abnormal events are people running, walking in the opposite direction, loitering and etc. The challenges of this dataset include camera shakes and a few outliers in the training data. Also, some normal pattern seldom appears in the training data.

4.1.3 UCSD Pedestrians

UCSD Ped1 dataset has 34 training and 36 testing video clips, where each clip contains 200 frames. The videos consist of groups of people walking towards and away from the camera. UCSD Ped2 dataset has 16 training and 12 testing video clips, where the number of frames of each clip varies. The videos consist of walking pedestrians parallel to the camera plane. Anomalies of the two datasets include bikers, skaters, carts, wheelchairs and people walking in the grass area.

4.2 Experimental Setup

We train the reconstructive model by minimising the reconstruction error of the input volume, whereas the predictive model is trained by minimising the prediction error of the subsequent frame of the input volume. We use Adam optimiser with an initial learning rate of 0.001 and allow it taking the role of setting the learning rate automatically based on the model's weight update history. Setting a learning rate too large will cause the training to diverge, while setting a learning rate too small will take too long to converge. The values of the initial learning rate and other hyperparameters in the optimiser follow those provided by Kingma and Ba (2014). To ensure the data fit into the GPU memory, we

use mini-batches of size between 8 and 32, depending on the time length. 15% of training volumes are held out for use as the validation data. Each training volume is trained for a maximum of 500 epochs or until the reconstruction loss of validation data stop decreasing after 20 consecutive epochs. The maximum number of training epochs is empirically selected at 500 as we observed that the experiments we performed would have converged before 100 epochs. The time length is preset at $T = 8$ so that it is long enough but not too long to fit into our GPU memory during training.

The weights of all convolutional and recurrent layers are initialised using the Glorot uniform initialiser (Glorot and Bengio, 2010). The algorithm automatically scales the initial weights based on the number of input and output neurons to prevent the weights from starting out too small or large. If the weights start too small, the learning signal shrinks as it passes through each layer until they become too small to be useful. Too large of the weights might cause the output to become saturated and the gradients to approach zero, rendering the weights useless. Glorot initialiser ensures that the weights are in the reasonable range prior to training, thus speeds up the convergence of the network. The weight values are drawn from a uniform distribution between $[-m, m]$, where $m = \sqrt{6/(n_{in} + n_{out})}$, where n_{in} is the number of input neurons and n_{out} is the number of output neurons.

All models are implemented in Python programming language using a popular deep learning framework Keras [2]. Keras is a high-level neural networks application programming interface (API), capable of running on top of deep learning library such as Google Tensorflow [3]. Keras supports both convolutional networks and recurrent networks, as well as combinations of both. It is also highly modular – neural layers, cost functions, optimisers, initialisation schemes, activation functions, regularisation schemes are all standalone mod-

[2]https://keras.io
[3]https://www.tensorflow.org

ules that users can combine to create new models. The computing-intensive operations are accelerated by parallelising the computations on graphical processing units (GPUs). All evaluations and visualisations were also executed in Python with the aid of scientific computing libraries such as Numpy [4] and Scikit-Learn [5].

4.3 Results

In this section, we used time length $T = 8$ for all experiments and applied the same preprocessing steps prescribed in Section 3.1 for all datasets unless otherwise specified.

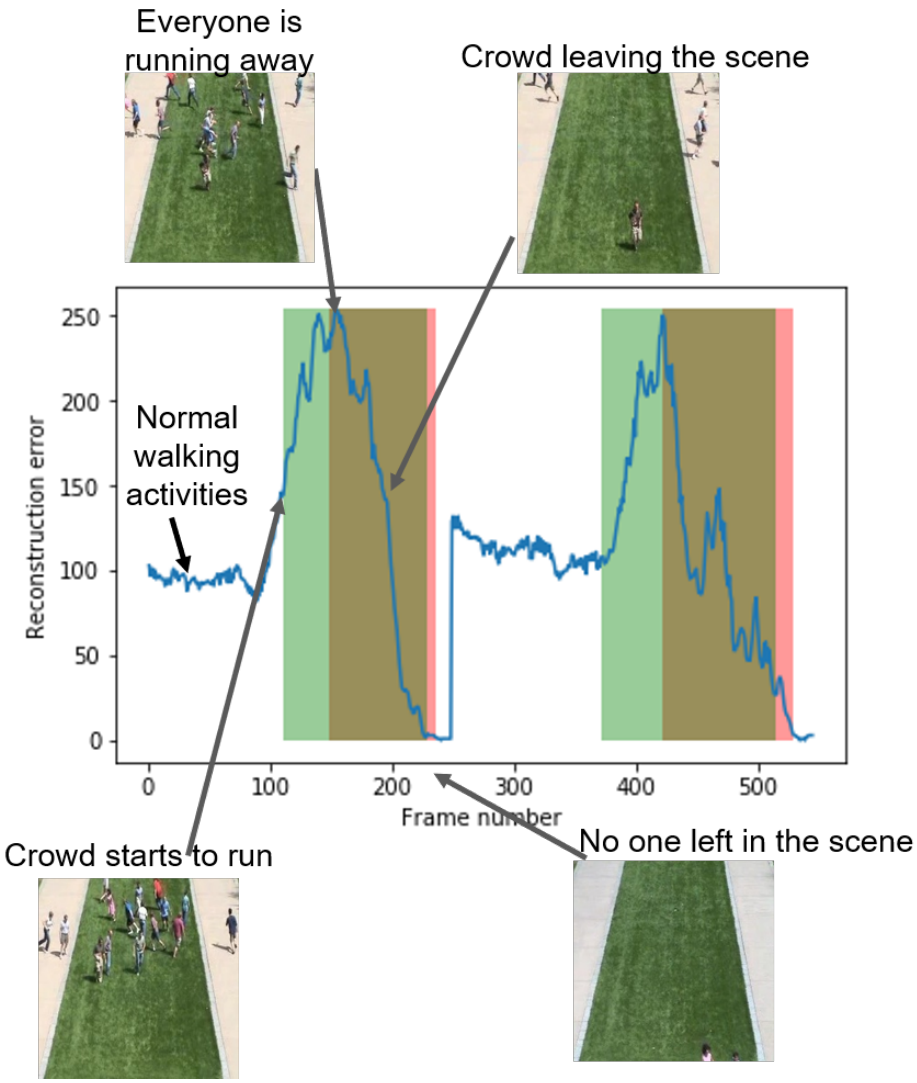
4.3.1 Global Anomalous Events: UMN dataset

We noticed that there is no predefined groundtruth for this dataset – most researchers who evaluated on this dataset uses the result of the publisher of this dataset as the groundtruth label. Though every abnormal event is well-captured by the publisher, it was slightly late at detecting the abnormalities. For each scene, we present in Figure 4.1 the original "groundtruth" and the adjusted groundtruth labels along with our detection result for comparison.

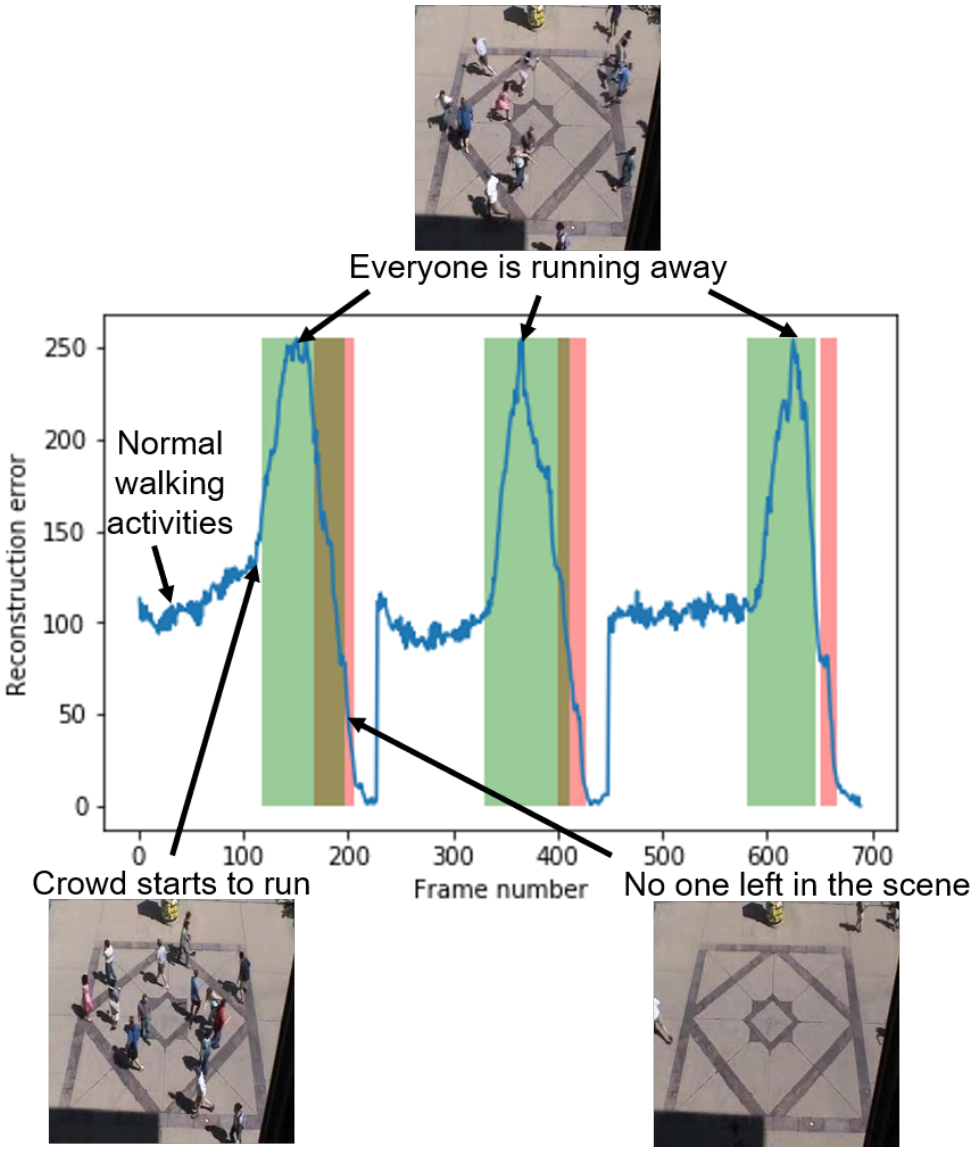
When the abnormal event is starting, the reconstruction error rose and created a spike when the whole crowd is dispersing. The error drops when the level of abnormality in the scene drops, as the crowd leave the scene entirely. All three scenes share the same type of abnormal event, which is the crowd dispersing from the scene. We showcase the reconstruction error plot against each frame from each scene in Figure 4.1. Figure 4.2 shows an example of a detected abnormal event where it was not labelled in the adjusted groundtruth.

The Lawn scene has even lighting condition in all frames, whereas there

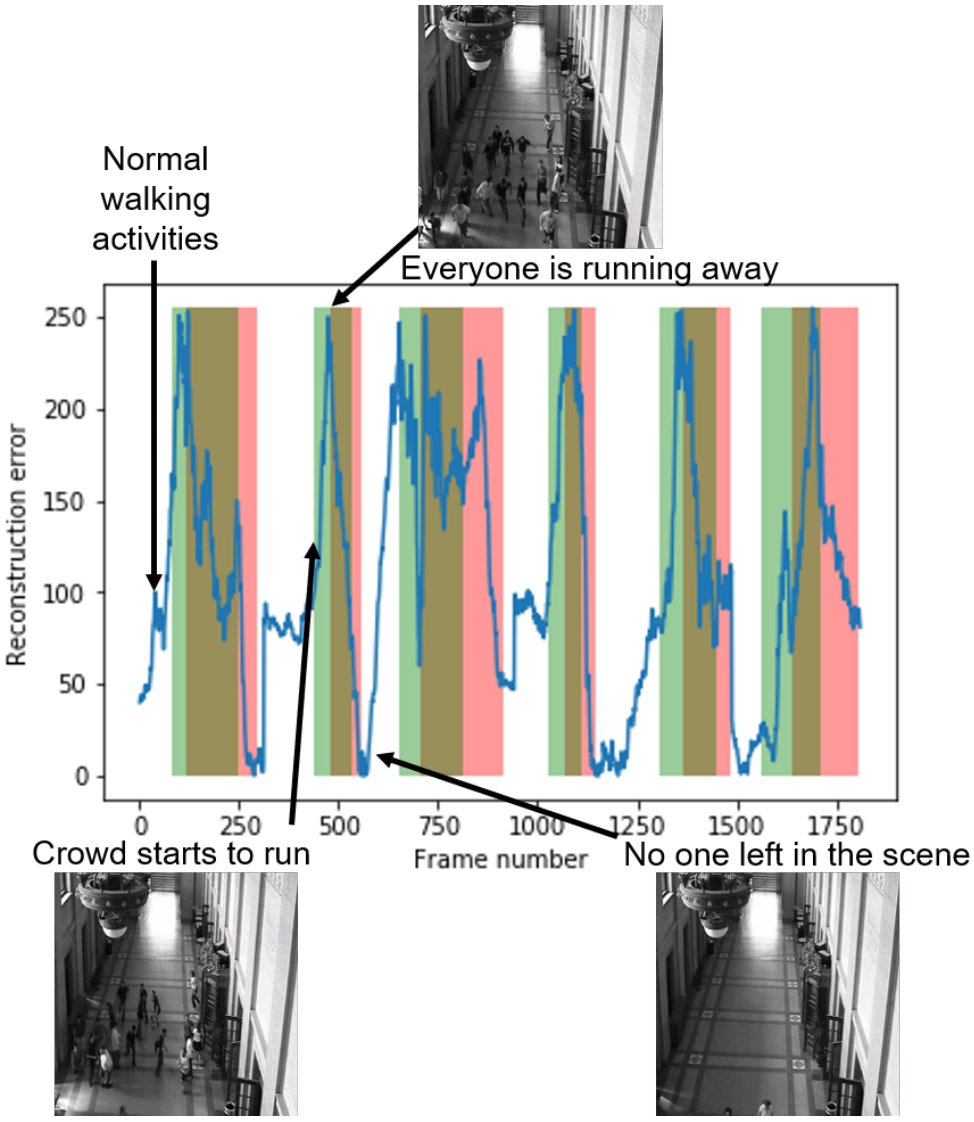
[4] http://www.numpy.org
[5] http://scikit-learn.org


Everyone is running away
Crowd leaving the scene
Normal walking activities
Reconstruction error
Frame number
Crowd starts to run
No one left in the scene

(a) Lawn Scene

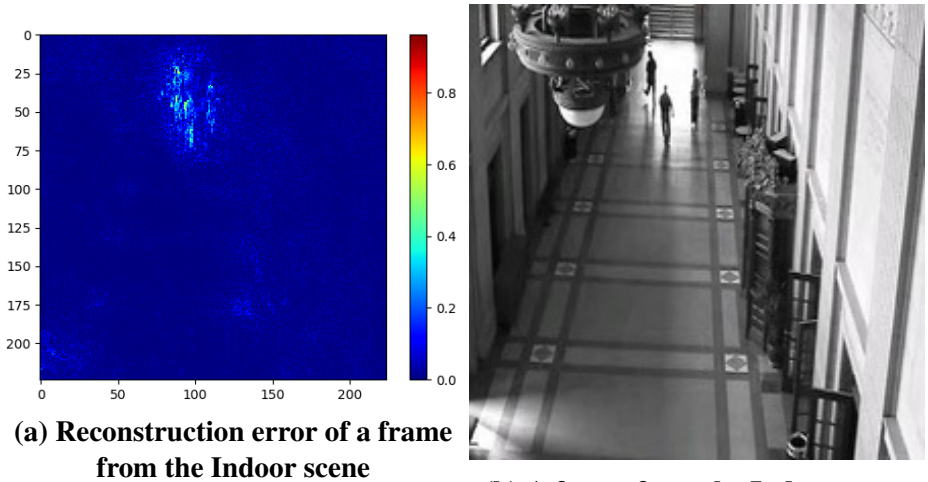


(b) Plaza Scene



(c) Indoor Scene

Figure 4.1: Reconstruction error of all three scenes from the UMN dataset, output by the proposed method. The red shaded region indicates the original groundtruth, where the green region represents the adjusted groundtruth. All events are successfully captured by our models.



(a) Reconstruction error of a frame from the Indoor scene

(b) A frame from the Indoor scene

Figure 4.2: Figure 4.2a shows the reconstruction error of the testing frame on the right. The brighter region on the left figure highlights the region where the reconstruction error is higher. We observed that in the training dataset there was no such scenario where person enters the upper region of the scene.

were slight lighting changes in the Plaza scene. The Indoor scene is more challenging in terms of lighting changes when the door opens and closes. Though motion patterns vary in each clip, all three scenes are simplistic and have a similar definition of abnormal events thus most existing methods could achieve very high AUC ($> 90\%$) on this dataset (based on the original groundtruth).

Table 4.1 shows the frame-level AUC of ours and of other methods on the three scenes from the UMN dataset. Our method did not achieve a comparable AUC value. As reconstruction error is calculated by the Euclidean distance between the input frame and the reconstructed frame, when the crowd is leaving the scene and only a few people are left in sight, we observed that the reconstructed frame resembles the background more than a walking crowd. As our model try to predict the next frame provided $T-1$ consecutive frames, if no one is walking or running in a particular region in $T-1$ frames, our model will not "hallucinate" walking crowds in that region too. This cause the reconstruction error to be low when only a few people are left in the scene, even when they are running.

Table 4.1: Comparison of area under ROC curve (AUC) of different methods on each scene from the UMN dataset. Higher AUC is better. Some papers only publish the average AUC of all three scenes.

Method	AUC(%)		
	Lawn	Plaza	Indoor
Optical Flow (Mehran et al., 2009)		84.0	
Social Force (Mehran et al., 2009)		96.0	
Sparse Reconstruction (Cong et al., 2011)	99.5	96.4	97.5
Ours (original groundtruth)	42.3	28.0	70.6
Ours (adjusted groundtruth)	69.3	86.4	86.0

Our result is far from perfect, however, our objective is to showcase the generalisation capability of our model to detect various types of abnormal events without manually defining appearance or motion attributes and heuristics. We have shown that our method can distinguish abnormal events from the normal activities in each scene, and have even detected one more which were not defined in the groundtruth. In the following sections, our model will be evaluated against the more challenging local anomaly datasets.

4.3.2 Local Anomalous Events: Avenue and UCSD datasets

Table 4.2 shows the frame-level AUC and EER of ours and of other methods on the three local anomaly datasets. These figures are obtained from the published papers respectively, except Hasan et al. (2016) where we have access to the authors' source code and have it re-evaluated using our evaluation method. We outperform all other considered methods in two out of three datasets, with respect to frame-level AUC and EER. As pointed out by Medel (2016), the groundtruth annotation is incomplete as it is missing several instances of pedestrians walking off the walkway in Ped1 testing dataset. Plus, a few corrupted video frames were included in the testing set, triggering of our

Table 4.2: Comparison of area under ROC curve (AUC) and Equal Error Rate (EER) of different methods on local anomaly datasets. Higher AUC and lower EER are better. Most papers did not publish their AUC/EER for Avenue dataset.

Method	AUC/EER(%)		
	Ped1	Ped2	Avenue
Adam et al. (2008)	77.1/38.0	-/42.0	N/A
Mehran et al. (2009)	67.5/31.0	55.6/42.0	
Mahadevan et al. (2010)	74.2/32.0	61.3/36.0	
Wang and Snoussi (2013)	72.7/33.1	87.5/20.0	
Hasan et al. (2016)	74.9/29.5	85.9/21.7	80.4/27.3
Xu et al. (2017)	**92.1/16.0**	90.8/17.0	N/A
Ours	77.1/27.5	**93.0/13.3**	**80.8/26.2**

Table 4.3: Anomalous event and false alarm count detected by different methods on various event type in Avenue dataset.

	Run	Loiter	Throw	Opposite Direction	False Alarm
Groundtruth	12	8	19	8	0
Ours	11	8	19	8	9

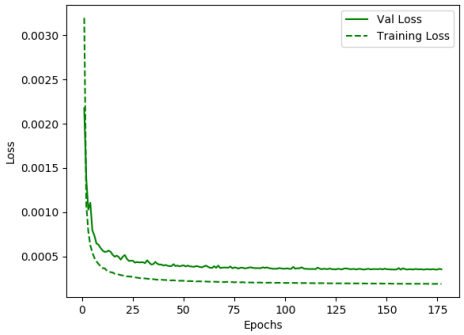
anomaly detector.

4.3.3 Avenue dataset

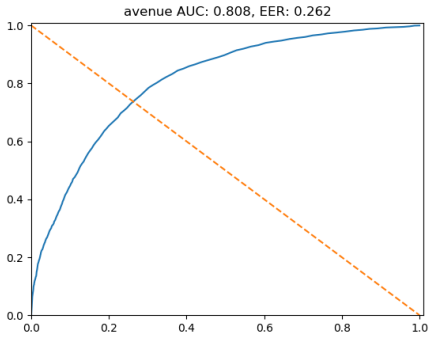
Figure 4.3 shows the training profile and the ROC curve of Avenue dataset using the proposed model. Note that we applied Adam optimiser with the same hyperparameters across all experiments.

Anomalous event count by event type

The event count breakdown according to the types of event is presented in Table 4.3 for Avenue dataset. All throwing, loitering and irregular interaction



(a) Model training and validation loss on Avenue dataset.



(b) ROC curve of the Avenue dataset.

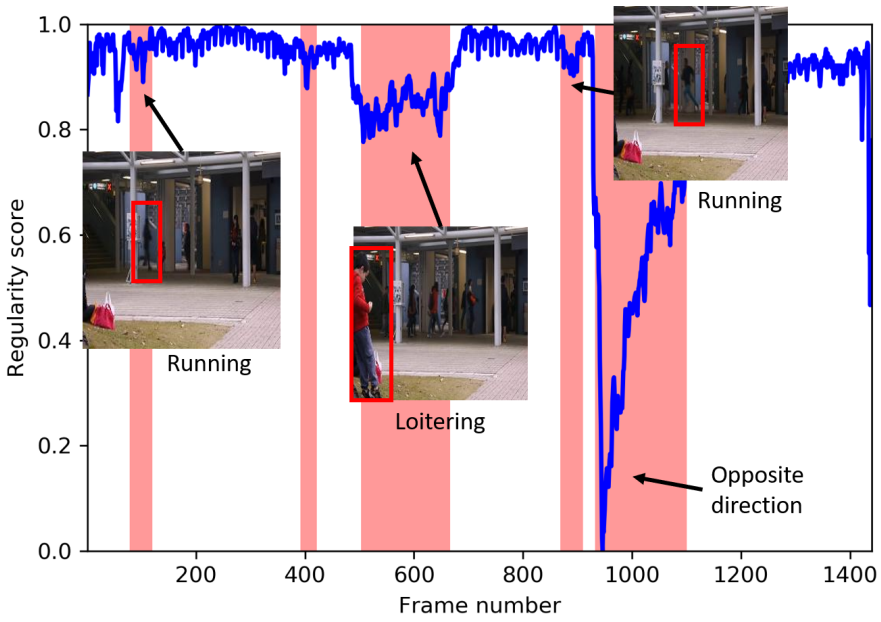
Figure 4.3: Training profile and ROC curve of Avenue dataset using the proposed model.

events are well captured by our proposed system. These are strong abnormalities that are significantly different from what was captured in the normal scenes. Some examples of abnormalities detected are presented in Figure 4.4.

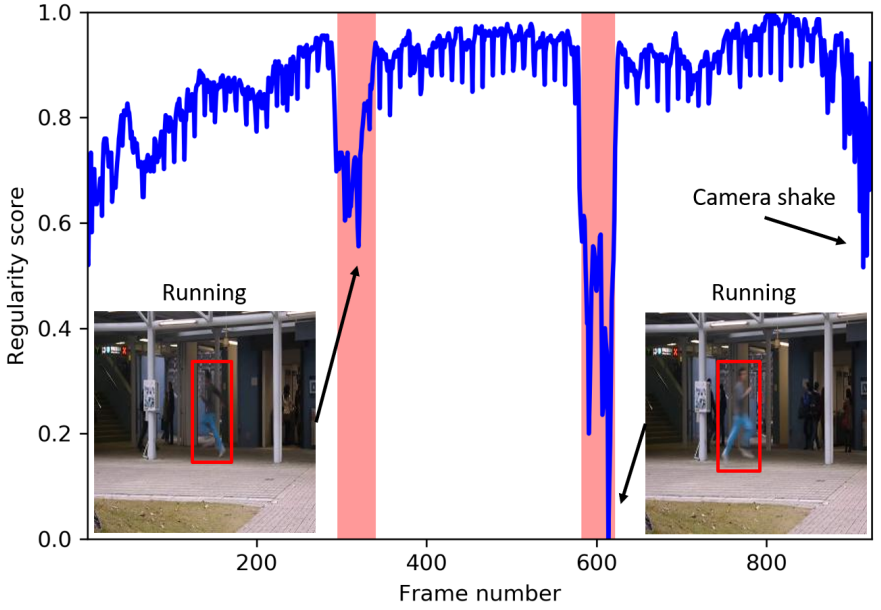
However, our system does have difficulties in detecting certain types of event. Missed detection of running events are due to (1) the crowded activities where multiple foreground events take place; and (2) the object of interest is far away from the camera. 5 out of 9 false alarms were due to camera shake, whereas the rest of the false alarms are caused by obstruction to the camera, such as walking outside the shaded area of the station. Examples of false alarms can be seen in Figure 4.5.

Predicting future frames with a predictive model

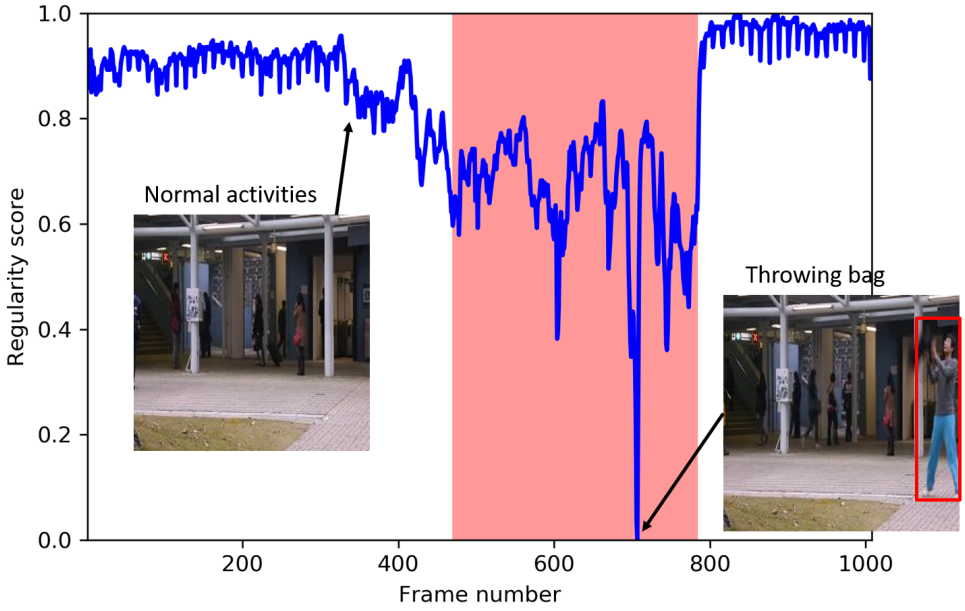
An advantage of using the predictive model to learn regularities is that its prediction can be inspected visually, enables us to gain an insight into what the model learned. We allowed the predictive model to predict 4 consecutive frames at timestep $t+1, t+2, t+3$ and $t+4$, given the initial 7 groundtruth frames at timestep $t-6, t-5, t-4, t-3, t-2, t-1$, and t. The upper row of each figure denotes the groundtruth frames and predicted frames are at the bottom row.



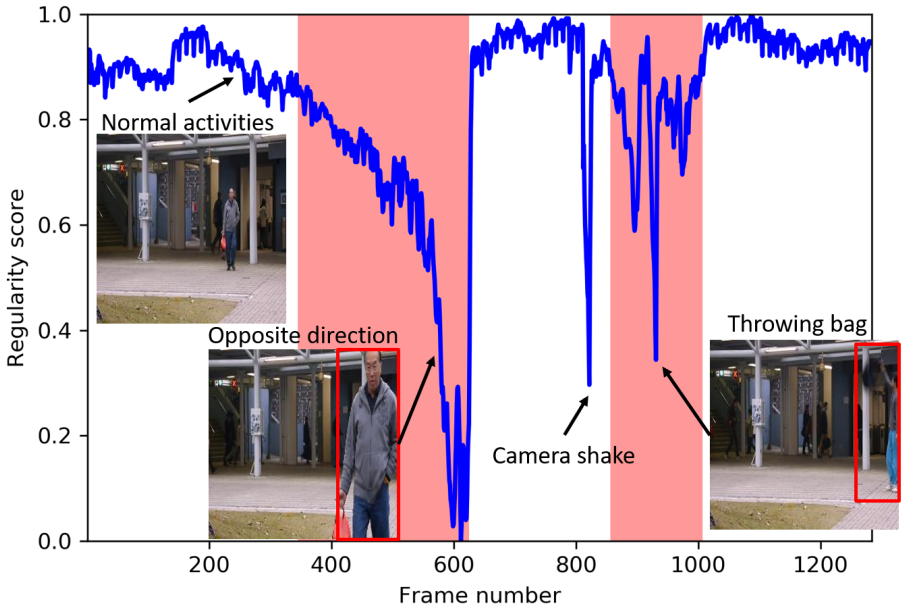
(a) Video #1: Loitering, running, and opposite direction



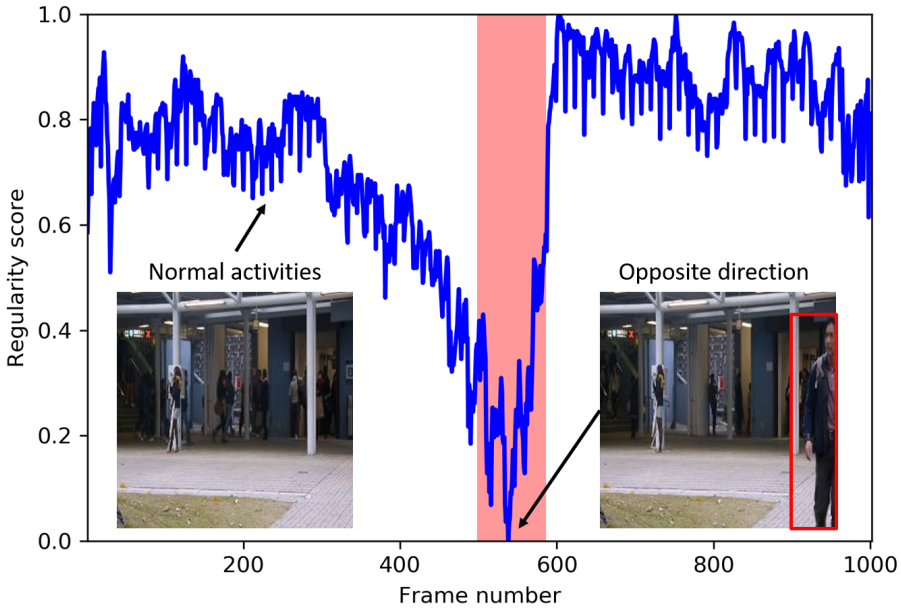
(b) Video #3: Running



(c) Video #5: Bag throwing

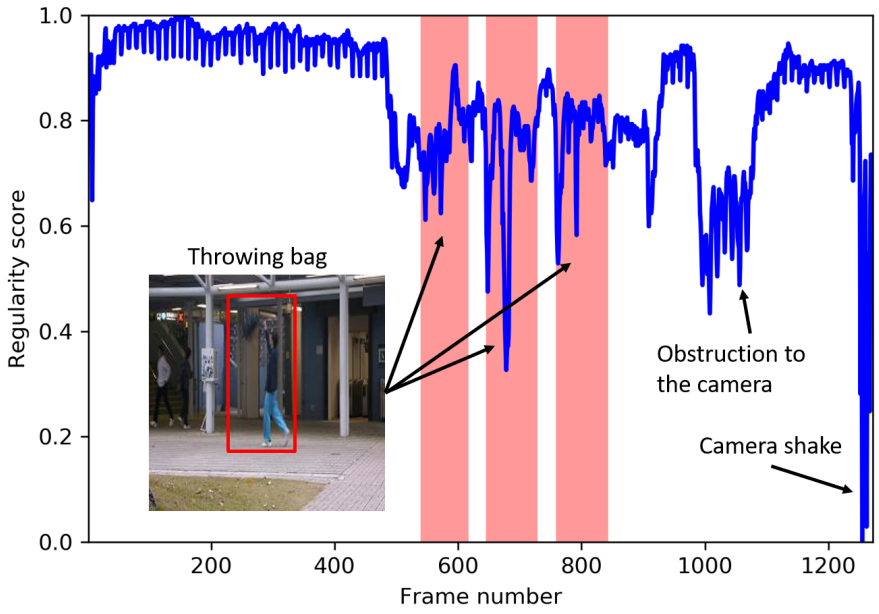


(d) Video #6: Opposite direction and bag throwing

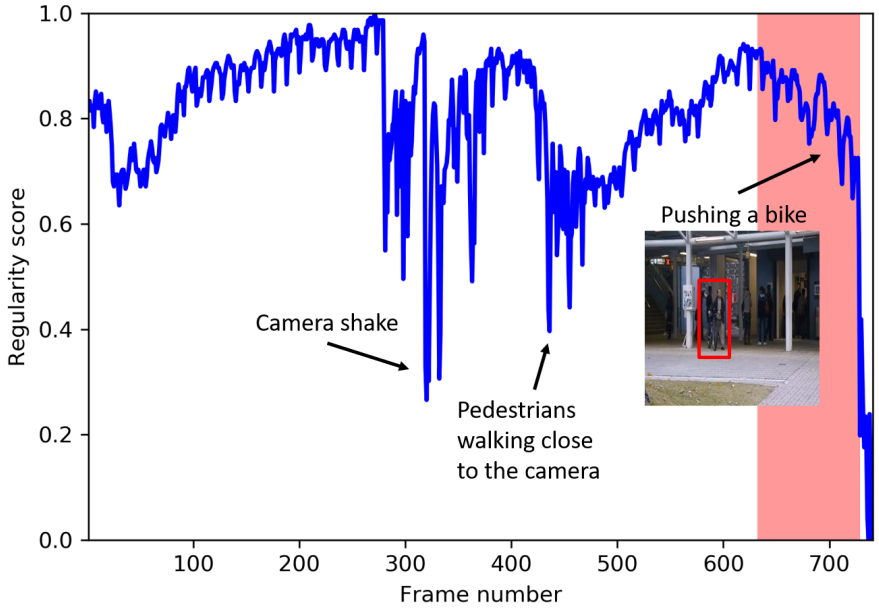


(e) Video #15: Opposite direction

Figure 4.4: Regularity score of video #1, #3, #5, #6 and #15 from the Avenue dataset, output by the proposed method. These events are successfully captured by our model.



(a) Video #12: Throwing



(b) Video #16: Bicycle (opposite direction)

Figure 4.5: Regularity score of video #12 and #16 from the Avenue dataset. There are some false positives due to camera shake and activities that were closer to the camera. Our model was also late at detecting the bicycle event in video #16.

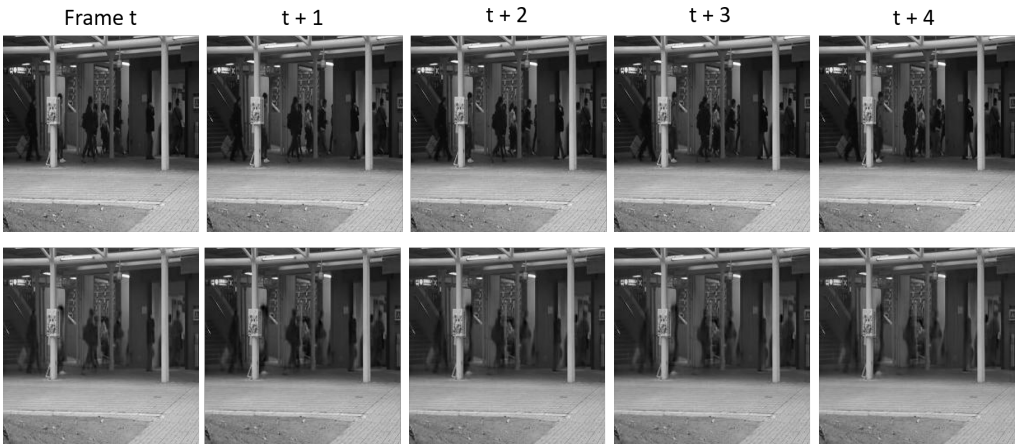


Figure 4.6: Predicting frames in normal scenes of Avenue test set video #7. Though the details of pedestrians in the future reconstruction are slightly blurred, the motion of pedestrian walking can still be seen across the predicted frames.

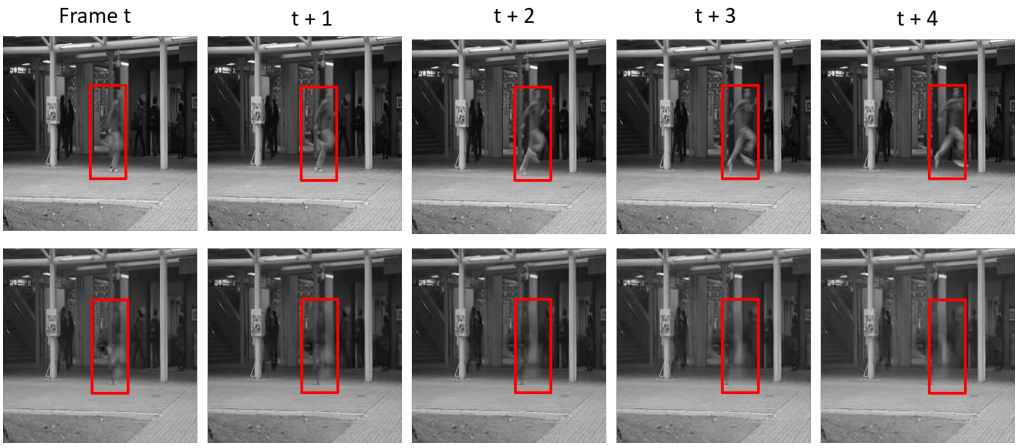
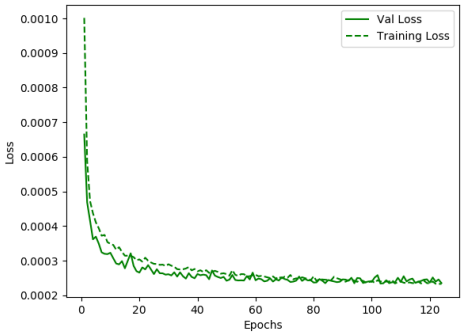


Figure 4.7: Predicting frames in abnormal scenes of Avenue test set video #3. It can be observed that the shape of the running person disappears in later frames.

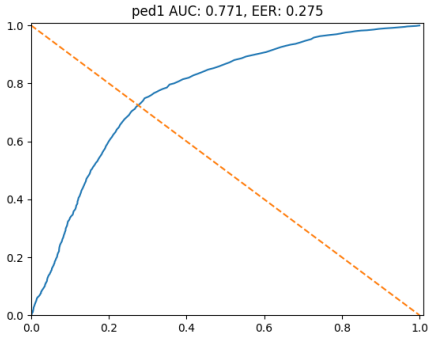
From Figure 4.6, the normal appearance and motions are well constructed, though some details of the pedestrians are lost due to the generalisation of a person's appearance by the predictive model. On the other hand, a running person cannot be constructed because such instance is unseen in the training data, hence 'disappeared' in future frames, as shown in Figure 4.7.

4.3.4 Ped1 dataset

Figure 4.8 shows the training profile and the ROC curve of UCSD Ped1 dataset using the proposed model.



(a) Model training and validation loss on Ped1 dataset.



(b) ROC curve of the Ped1 dataset.

Figure 4.8: Training profile and ROC curve of Ped1 dataset using the proposed model.

Table 4.4: Anomalous event and false alarm count detected by different methods on various event type in Ped1 dataset. Grass refers to pedestrians walking on grass event, while miscellaneous events include running and walking in a group.

	Biker	Skater	Cart	Wheelchair / Trolley	Grass	Misc.	False Alarm
Groundtruth	30	13	6	3	4	4	0
Ours	30	13	6	2	3	3	10

Anomalous event count by event type

The event count breakdown according to type of event is presented in Table 4.4 for Ped1 dataset. All bicycle and cart instances are well captured by our proposed system. These are strong abnormalities that are significantly different from what was captured in the normal scenes. Some examples of abnormalities detected are presented in Figure 4.9.

We observed that several instances of defined anomalies occurred in the training videos of Ped1 dataset. For instance, there were two instances of biker and three instances of walking on grass event. However, this does not stop the anomaly detector from classifying these observed instances as anomalous because these events are rare relative to the pedestrian walking event by the

number of occurrences in the training videos.

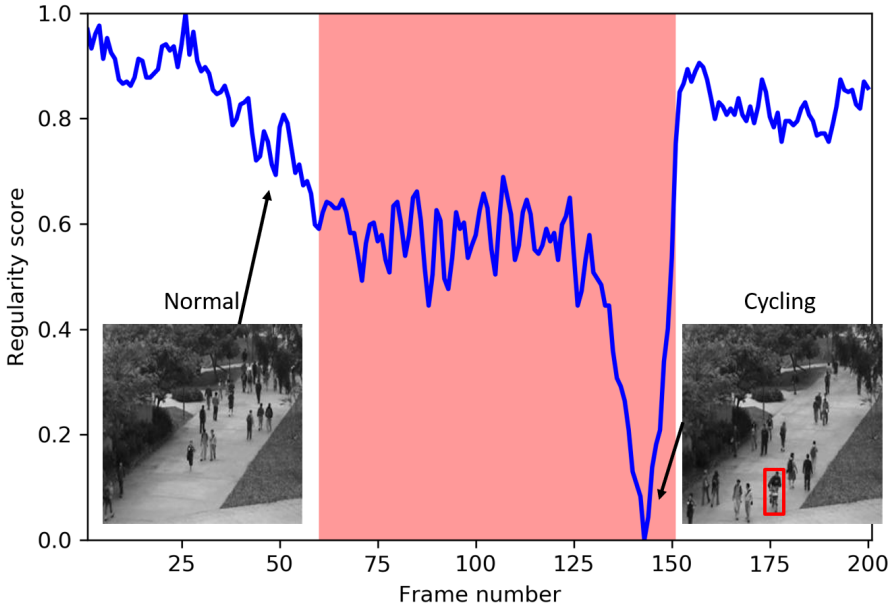
However, our system does have difficulties in detecting certain types of event. A walking on grass and a running event were missed by both models. Missed detection of the skater is due to the crowded usual activities around the skater. Some examples of missed events are shown in Figure 4.10. Similar to the observation in Avenue dataset, our model is sensitive to camera shakes and glitches. Other false alarms are due to the pedestrian motion walking in an unusual direction as depicted in Figure 4.11.

Predicting future frames with a predictive model

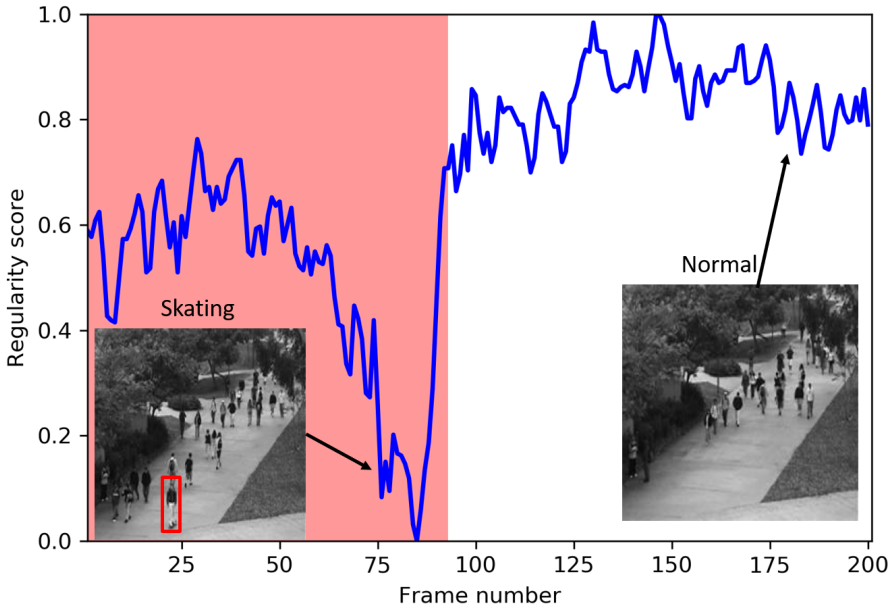
The predictive model is trained to predict $(t + 1)$-th frame based on the given t frames, thus able to predict future frames given the past input frames. This section showcases the future frames predicted by the predictive model. To amplify the visual differences, we allow the model to predict future frames at multiple timesteps.

Figure 4.12 shows the ability of the model to extrapolate the motion of pedestrians towards the direction each pedestrian is taking in the past given frames. The pedestrians on the walkway lose very little resolution at every predicted timestep even though the model is not trained to predict more than one frame at a time. It can also be seen that the position of each pedestrian's feet changes at each timestep and the changes resemble closely to the actual walking motion.

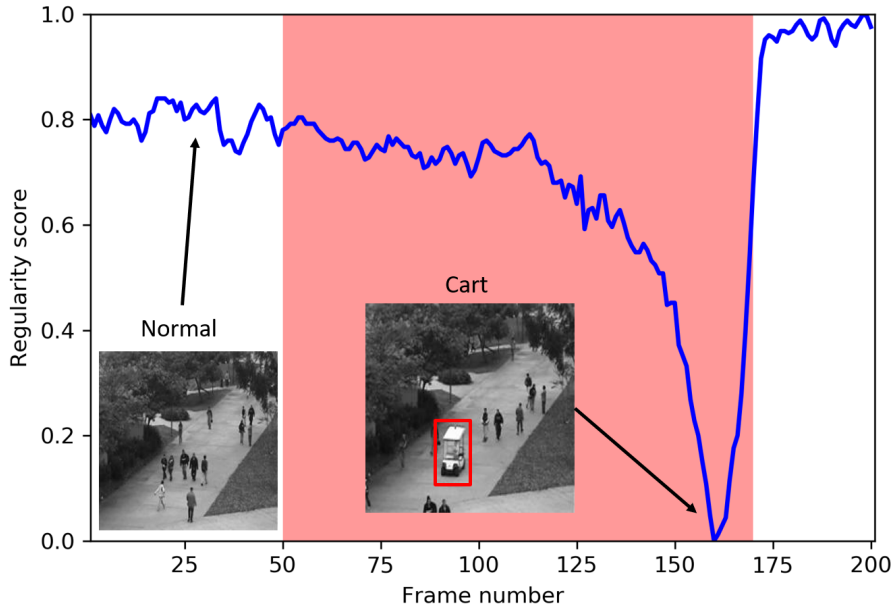
The cart seen in Figure 4.13 does not belong on a pedestrian walkway, as it is not observed in the training videos. As such, it is abnormal and unable to be predicted properly. It can be seen that the prediction of pedestrians walking – a normal event, within the anomalous sequence is portrayed correctly by the model, while the cart loses detail with each timestep. Similarly, bicycles are not observed in the training set, thus the model could not predict properly the shape



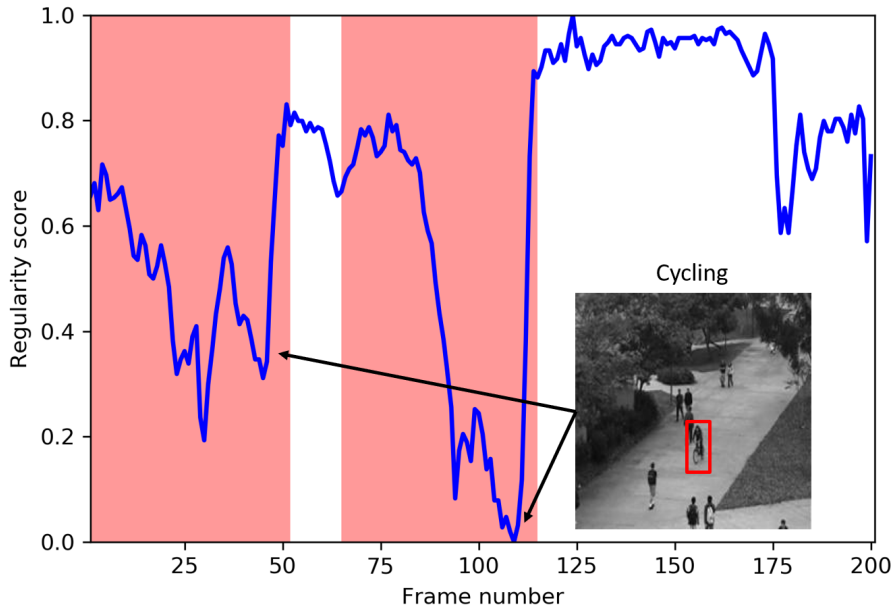
(a) Video #1: Biker



(b) Video #8: Skater



(c) Video #24: Cart



(d) Video #32: Biker

Figure 4.9: Regularity score of video #1, #8, #24 and #32 from the Ped1 dataset, output by the proposed method. These events are successfully captured by our model.

(a) Video #17: Running

(b) Video #31: Walking on grass

(c) Video #23: Wheelchair

Figure 4.10: Examples of missed events in video #17, #23 and #31 from the Ped1 dataset. Our model missed the above running and walking on grass event, and a wheelchair event.

(a) Video #10: Frame glitch

(b) Video #12: Unusual direction

Figure 4.11: Examples of false positives in video #10 and #12 from the Ped1 dataset. There are some false positives due to frame glitches as seen in Figure 4.11a. As presented in Figure 4.11b, walking in an unusual direction is detected as an anomaly by the proposed models.

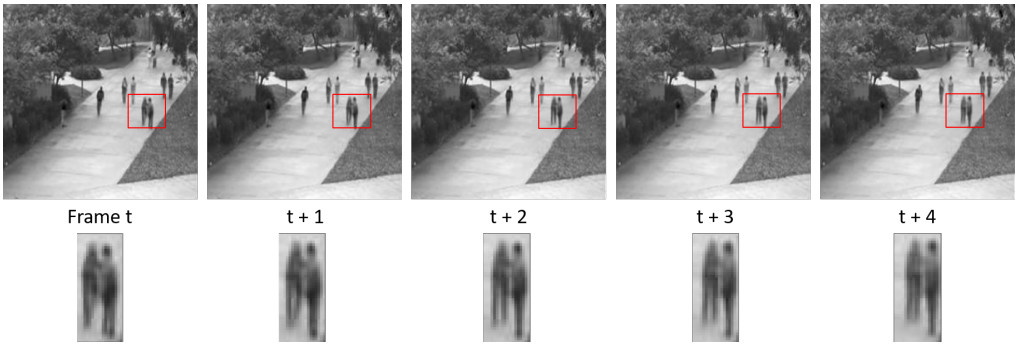


Figure 4.12: Predicting frames in normal scenes of Ped1 test set video #36. Bottom row magnifies the portion annotated by the bounding box in the upper row for a clearer view. The walking motion can be observed in the legs of the pedestrians.

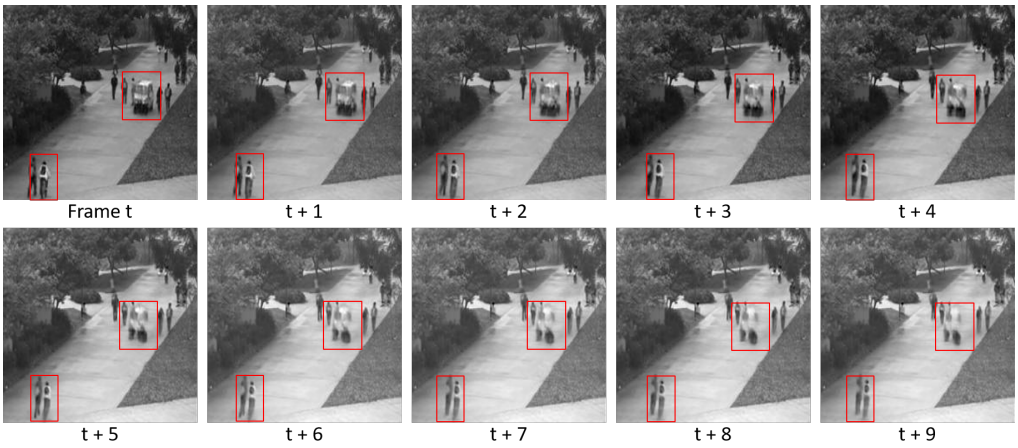
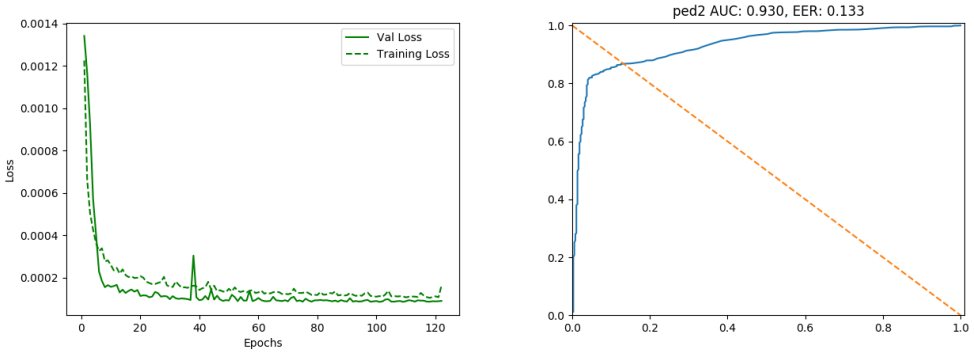


Figure 4.13: Predicting frames in abnormal scenes of Ped1 test set video #36. It can be observed that the shape of the cart 'evolves' into a pedestrian-like shape in later frames. Also, the appearance of the bicycle has collapsed and disappeared in future frames.

of a bicycle – at each successive timestep, the appearance of bicycle collapses and eventually only the biker remains visible in the scene.

4.3.5 Ped2 dataset

Figure 4.14 shows the training profile and the ROC curve of UCSD Ped2 dataset using the proposed model.



(a) Model training and validation loss on Ped2 dataset.

(b) Frame-level ROC curve of the Ped2 dataset.

Figure 4.14: Training profile and ROC curve of Ped2 dataset using the proposed model.

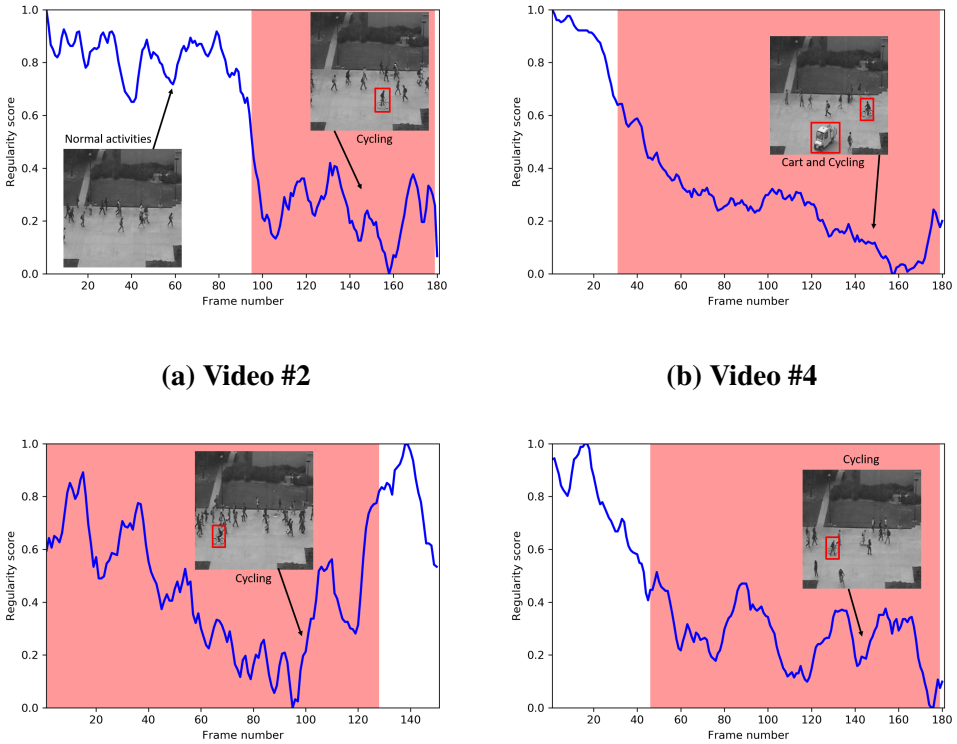
Table 4.5: Anomalous event and false alarm count detected by different methods on various event type in Ped2 dataset.

	Biker	Skater	Cart	False Alarm
Groundtruth	15	3	1	0
Ours	13	2	1	0

Anomalous event count by event type

The event count breakdown according to type of event is presented in Table 4.5 for Ped2 dataset. Most bicycle and cart instances are well captured by our proposed system. These are strong abnormalities that are significantly different from what was captured in the normal scenes. Some examples of abnormalities detected are presented in Figure 4.15.

Several cycling events were missed by our model, as shown in Figure 4.16a. Two of the missed bike instances were located far away from the camera. Another bike was missed due to occlusions in a crowded scene. Several false alarms were triggered and these are mostly due to camera shake and occlusions of multiple pedestrians.



(a) Video #2

(b) Video #4

(c) Video #5

(d) Video #7

Figure 4.15: Regularity score of video #2, #4, #5 and #7 from the UCSD Ped2 dataset, output by the reconstructive and predictive variants of the proposed method.

(a) Video #11

(b) Video #12

Figure 4.16: Snapshots of video #11 and #12 from the UCSD Ped2 dataset, showing the anomalous events which were failed to be captured by the proposed method. Each of the anomalous instances is labelled with a red bounding box.

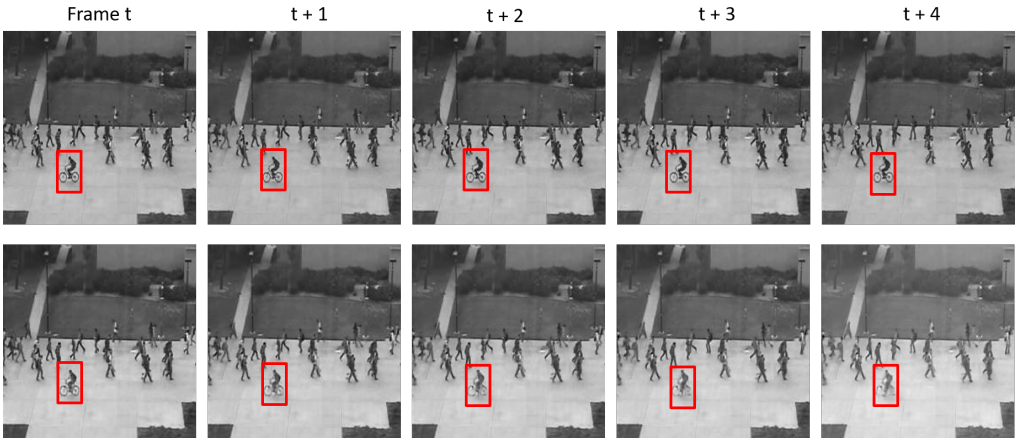


Figure 4.17: Predicting frames of Ped2 test set video #5. The bicycle disappears in later frames while the biker evolves into a walking pedestrian. The walking motion can be observed in the legs of the other pedestrians.

Predicting future frames with a predictive model

The predictive model is trained to predict $(t+1)$-th frame based on the given t frames, thus able to predict future frames given the past input frames. This section showcases the future frames predicted by the predictive model. To amplify the visual differences, we allow the model to predict future frames at multiple timesteps. The upper row of Figure 4.17 and 4.18 denotes the groundtruth frames and predicted frames are at the bottom row.

The bicycles seen in Figure 4.17 and 4.18 do not belong on a pedestrian walkway, as it is not observed in the training videos. As such, it is abnormal and unable to be predicted properly. It can be seen that the prediction of pedestrians walking – a normal event, within the anomalous sequence is portrayed correctly by the model, while the bicycle disappears with each timestep and the cyclist evolves into a walking pedestrian. Similarly, skateboards are not observed in the training set, thus the model could not predict properly the shape of a skateboard – at each successive timestep, the appearance of skateboard collapses and eventually only the skater remains visible in the scene.

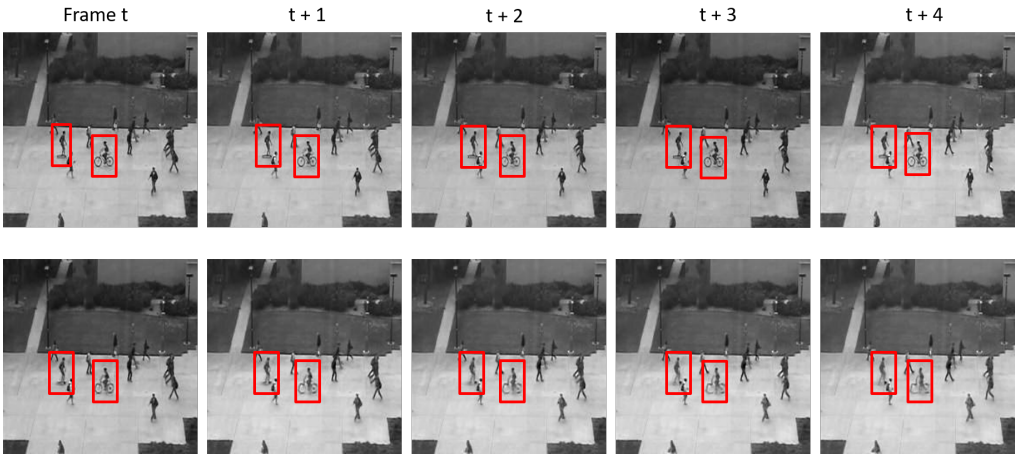


Figure 4.18: Predicting frames of Ped2 test set video #8. It can be observed that the shape of the bicycle and skateboard disappear at each timestep.

4.4 Chapter Summary

This chapter presented the quantitative and qualitative analysis of the results evaluated on the benchmark datasets using our proposed model and compared with other existing methods. We have proved experimentally the capability of unsupervised learning for automated abnormal event detection in surveillance videos. We are able to achieve comparable performance for local anomaly detection without preprocessing and manually defining spatiotemporal features. In the following chapter, we will conclude this research and suggest a few ideas on how our work can be improved.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

Video surveillance systems are of increasing importance in providing public safety and security. However, the current implementation requires human monitoring at all times. There has been a lot of research interest in automating this process by allowing machines to learn the spatiotemporal patterns in videos to identify abnormal events automatically. The challenges to delegate the task of monitoring video events to the machine were listed in the first chapter of the dissertation. This research aims to address these challenges by achieving the following objectives:-

1. Extract useful features of videos in various environments
2. Eliminate tracking and grid-based processing
3. Use unsupervised learning methods for unusual event detection
4. Avoid pre-determining the types or the number of types of events

We have achieved these research objectives in previous chapters collectively, as follows:

- **Extract useful features of videos in various environments**

  We have developed an efficient model to capture spatiotemporal similarities between frames. We introduce a deep neural network model consisting of convolutional autoencoders to learn the regular activity patterns while avoiding the curse of dimensionality. The first part of the neural network is a spatial convolutional autoencoder, which serves the purpose of extracting appearance features from individual frames. Then, the spatial

features are fed into a temporal autoencoder to capture temporal patterns. By incorporating both autoencoders, the resulted features not only group similar events together, but also able to differentiate event patterns which have not been observed before. For example, our model perceives cycling in Ped1 dataset and running events in Avenue dataset as anomalous, because these events are not observed in the training dataset.

- **Eliminate tracking and grid-based processing**

  In Chapter 2 we reviewed several groups of methods which used tracking or applied grid-based processing in order to reduce dimensionality and to cluster similar events. This approach is helpful in detecting local anomalies where instances occur in a small region of the whole scene, however, it is difficult to use tracking for large crowds or to determine the optimal size of such grid. Using the proposed method, we are able to eliminate the use of tracking and grids to detect both global and local abnormal events.

- **Use unsupervised learning methods for unusual event detection**

  For long video footages, it is very tedious and impractical to label each region of the video with its event category. We have developed a technique that can work on video footages with minimal labelling. To prepare the training dataset, the only required ingredient is long video segments with the assumption that no abnormal events have occurred in the provided segments. There is no need to draw bounding boxes of each event or categorise the events into smaller categories. Though we have shown that in the result discussion section where our probabilistic model would still flag certain rare events that were occurred in the training dataset as anomalous, this may be desirable and these events are worth noticing because these events may be the outliers of the training data.

- **Avoid pre-determining the types or the number of types of events**

  In Chapter 2 we have reviewed several methods which are greatly depen-

dent on a specified number of event types in each video. In practical scenarios, it is impossible to anticipate the number of possible types of events that would occur in each video stream since it is unknown beforehand. We have shown experimentally that our model still performs comparably well without making such assumptions.

We have addressed the above-mentioned challenges and achieved a comparable result of 77.1%, 93.0% and 80.8% respectively on local anomaly datasets (UCSD Ped1, Ped2, and Avenue dataset). Though it is not new to implement convolutional models nor LSTM for spatiotemporal anomaly detection, the convolutional autoencoder proposed by Hasan et al. (2016), convolution and pooling operations are performed only spatially, even though the proposed network takes multiple frames as input, because of the 2D convolutions, after the first convolution layer, temporal information is collapsed completely (Tran et al., 2015). While Medel (2016) used convolutional LSTM to learn spatiotemporal features, we employed a combination of 3D convolutions and convolutional LSTM layers to achieve comparable result with less computational time and space.

5.2 Limitations and Future Works

Firstly, the spatiotemporal model operates in batch mode, which requires the entire training video to be made available before computing the spatio-temporal features. The model can be finetuned with new video segments, however, unlike the online models, it must be retrained and could only be updated periodically but not real-time.

Secondly, it is difficult to manually modify the events to be flagged as normal or abnormal. The basic assumption of the model is that abnormal events are those that never happened or occurred rarely. Since the spatio-temporal pat-

terns are learned probabilistically from the training dataset, therefore it is difficult to manually flag video events and adjust according to domain knowledge and human supervision. Human supervision may be useful to provide a certain level of semantics. Using the analogy of how human beings learn during their infancy, in the first year, a baby starts to observe and explore the world around him, but he may not know the semantics of things around him (unsupervised learning). In the later years, he is taught by their parents the semantics and meanings of these things (supervised learning).

In the context of learning a normality model, if an event is new to the model but it should be considered normal, such event has to occur frequently enough in order to be categorised as a 'normal' event by our model. Similarly, if an event should be considered 'abnormal' but it occurs frequently in the training videos, then these segments containing the specified event should be excluded from the training data and have the model retrained to get these events flagged as abnormal. For future work, we may look into alternative methods that incorporate a human feedback loop, so that the learned model can adjust itself according to feedbacks for better detection and reduced false alarms. One idea is to add a supervised module to the current system, which the supervised module works only on the video segments filtered by our proposed method, then train a discriminative model to classify anomalies when enough video data has been acquired.

Thirdly, most of the benchmark datasets are subjective or synthetic. The UMN dataset is staged and acted by a fixed group of people. The Avenue dataset is also partially staged. It is understandable because it is difficult to collect abnormal events. Furthermore, it is very difficult to acquire real data due to privacy. These datasets are collected based on the opinions of the published authors, and the subjectivity of the groundtruth cannot be ruled out. Therefore, a method with excellent performance for these datasets does not mean it will work well in real applications. In future work, we would like to apply our framework

to real environments to evaluate its performance.

Also, we would like to investigate if the model could benefit from longer video samples. Since none of the video segments are labelled, the definition of anomaly depends on the training video if an event does not occur or occur rarely in the training video, it is still considered anomalous even though it may be considered normal according to human common sense. This may be improved through longer video samples where more samples of normal events could have been captured.

Lastly, though our method could work without domain knowledge, it only works in a fixed perspective and sensitive to camera shakes and glitches. This is because our model does not understand the objects and semantics in the videos, therefore unable to differentiate between a moving scene and moving objects. A tradeoff between unsupervised learning and semantics understanding to achieve robustness can be further explored.

LIST OF REFERENCES

Adam, A., Rivlin, E., Shimshoni, I. and Reinitz, D., 2008. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3), pp. 555–560.

Basharat, A., Gritai, A. and Shah, M., 2008. Learning object motion patterns for anomaly detection and improved object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 24-26 June 2008 Anchorage, Alaska, pp. 1–8.

Bera, A., Kim, S. and Manocha, D., 2016. Realtime anomaly detection using trajectory-level crowd behavior learning. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 26 June–1 July 2016 Nevada: pp. 1289–1296.

Blei, D.M., 2012. Probabilistic topic models. *Communications of the ACM, 55*(4), pp.77-84.

Chalapathy, R., Menon, A.K. and Chawla, S., 2017. Robust, deep and inductive anomaly detection. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 18-22 September 2017, Skopje, Macedonia: Springer, pp. 36-51.

Chong, Y. S. and Tay, Y. H., 2015. Modeling video-based anomaly detection using deep architectures: Challenges and possibilities. *10th Asian Control Conference (ASCC)*, 31 May-3 June 2015, Kota Kinabalu, Malaysia: pp. 1–8.

Chong Y.S. and Tay Y.H., 2017. Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder. In: Cong F., Leung A., Wei Q. (eds.). *Advances in Neural Networks - ISNN 2017, Lecture Notes in Computer Science, vol 10262*. Springer, Cham.

Cong, Y., Yuan, J. and Liu, J., 2011. Sparse reconstruction cost for abnormal event detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 21-23 June 2011, Colorado, USA: pp. 3449–3456.

Cong, Y., Yuan, J. and Tang, Y., 2013. Video anomaly search in crowded scenes via spatio-temporal motion context. *IEEE Transactions on Information Forensics and Security*, 8(10), pp. 1590–1599.

Cui, P., Sun, L. F., Liu, Z. Q. and Yang, S. Q., 2007. A sequential monte carlo approach to anomaly detection in tracking visual events. *IEEE Conference on Computer Vision and Pattern Recognition*, 18-23 June 2007 Minnesota, USA: pp. 1–8.

Dangeti, P., 2017. *Statistics for Machine Learning: Techniques for Exploring Supervised, Unsupervised, and Reinforcement Learning Models with Python and R*. Packt Publishing.

Doll´ar, P., Rabaud, V., Cottrell, G. and Belongie, S., 2005. Behavior recognition via sparse spatio-temporal features. *Proceedings - 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, vol. 2005, pp. 65–72.

Fang, Z., Fei, F., Fang, Y., Lee, C., Xiong, N., Shu, L. and Chen, S., 2016. Abnormal event detection in crowded scenes based on deep learning. *Multimedia Tools and Applications* 75(22), pp. 14617–14639.

Glorot, X. and Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256.

Gupta, M., Gao, J., Aggarwal, C. C. and Han, J., 2014. Outlier detection for temporal data: A survey. IEEE Transactions on Knowledge and Data Engineering 26(9), pp. 2250–2267.

Han, S., Fu, R., Wang, S. and Wu, X., 2013. Online adaptive dictionary learning and weighted sparse coding for abnormality detection. *2013 IEEE International Conference on Image Processing*, Melbourne, VIC, 2013, pp. 151-155.

Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K. and Davis, L. S., 2016. Learning temporal regularity in video sequences. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 733-742.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hu, X., Hu, S., Luo, L. and Li, G., 2016. Abnormal event detection in crowded scenes via bag-of-atomic-events-based topic model. *Turkish Journal of Electrical Engineering & Computer Sciences* 24(4), pp. 2638–2653.

Inoue, N., Kamishima, Y., Wada, T., Shinoda, K. and Sato, S., 2011. *TokyoTech+Canon at TRECVID 2011* [Online]. Available at: https://www-nlpir.nist.gov/projects/tvpubs/tv11.papers/tokyotechcanon.pdf [Accessed: 20 March 2018].

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, pp. 448–456.

Ji, S., Yang, M. and Yu, K., 2013. 3D Convolutional Neural Networks for Human Action Recognition. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), pp. 221–31.

Jiang, F., Wu, Y. and Katsaggelos, A. K., 2009. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. *IEEE Transactions on Image Processing*, 18(4), pp. 907–913.

Jiang, F., Yuan, J., Tsaftaris, S. A. and Katsaggelos, A. K., 2011. Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, 115(3), pp. 323 – 333.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732.

Kim, J. and Grauman, K., 2009. Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pp. 2921–2928.

Kingma, D. and Ba, J., 2014. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*, 7-9 May 2015 San Diego, California: arXiv.org.

Klaser, A., Marszaek, M. and Schmid, C., 2008. A spatio-temporal descriptor based on 3d-gradients. *BMVC 2008 - 19th British Machine Vision Conference*,

Sep 2008, Leeds, United Kingdom: British Machine Vision Association, pp.275:1-10.

Ko, T., 2008. A survey on behavior analysis in video surveillance for homeland security applications. In: *2008 37th IEEE Applied Imagery Pattern Recognition Workshop*, pp. 1–8.

Kratz, L. and Nishino, K., 2009. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In*: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1446–1453.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.

Laptev, I., Marszałek, M., Schmid, C. and Rozenfeld, B., 2008. Learning realistic human actions from movies. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1-8.

Lauer, F., Suen, C. Y. and Bloch, G., 2007. A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*, 40(6), pp. 1816–1824.

Li, C., Han, Z., Ye, Q. and Jiao, J., 2011. Abnormal behavior detection via sparse reconstruction analysis of trajectory. *2011 Sixth International Conference on Image and Graphics*, Hefei, Anhui, 2011, pp. 807-810.

Lu, C., Shi, J. and Jia, J., 2013. Abnormal event detection at 150 FPS in MATLAB. *2013 IEEE International Conference on Computer Vision*, Sydney, NSW, 2013, pp. 2720-2727.

Ma, K., Doescher, M. and Bodden, C., 2015. Anomaly detection in crowded scenes using dense trajectories [Online]. Available at: https://www.semanticscholar.org/paper/Anomaly-Detection-In-Crowded-Scenes-Using-Dense-Tr-Ma-Doescher/eaef3547ec6c32fe683d58c7789dbff57c585b92 [Accessed: 25 March 2017].

Mahadevan, V., Li, W., Bhalodia, V. and Vasconcelos, N., 2010. Anomaly detection in crowded scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1975–1981.

Medel, J. R., 2016. *Anomaly Detection Using Predictive Convolutional Long Short-Term Memory Units*. Master's thesis, Rochester Institute of Technology, New York.

Mehran, R., Oyama, A. and Shah, M., 2009. Abnormal crowd behavior detection using social force model. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 935–942.

Mo, X., Monga, V., Bala, R. and Fan, Z., 2014. Adaptive sparse representations for video anomaly detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(4), pp. 631–645.

Mousavi, H., Galoogahi, H. K., Perina, A. and Murino, V., 2016. Detecting abnormal behavioral patterns in crowd scenarios. In: Esposito, A. and Jain, L.C. (eds.). *Toward Robotic Socially Believable Behaving Systems – Volume II: Modeling Social Signals*. Cham: Springer International Publishing, pp. 185-205.

Mousavi, H., Nabi, M., Galoogahi, H. K., Perina, A. and Murino, V., 2015. Abnormality detection with improved histogram of oriented tracklets. In: Murino, V. and Puppo, E. (eds.). *Image Analysis and Processing – ICIAP 2015*. Cham: Springer International Publishing, pp. 722-732.

Nair, V. and Hinton, G. E., 2010. Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, USA: Omnipress, pp. 807–814.

Ng, A., 2000. *CS229 lecture notes* [Online]. Available at: http://cs229.stanford.edu/notes/cs229-notes1.pdf [Accessed: 5 January 2017].

Numenta, 2015. The science of anomaly detection [Online]. Available at: https://numenta.com/assets/pdf/whitepapers/Numenta%20White%20Paper%20-%20Science%20of%20Anomaly%20Detection.pdf [Accessed: 5 January 2017].

Oneata, D., Verbeek, J. and Schmid, C., 2013. Action and event recognition with Fisher vectors on a compact feature set. *2013 IEEE International Conference on Computer Vision*, Sydney, NSW, 2013, pp. 1817-1824.

Patraucean, V., Handa, A. and Cipolla, R., 2015. Spatio-temporal video autoencoder with differentiable memory. International Conference on Learning Representations (2015), pp. 1–10. Available at: http://arxiv.org/abs/1511.06309.

Piciarelli, C., Micheloni, C. and Foresti, G. L., 2008. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11), pp. 1544–1554.

Popoola, O. P. and Wang, K., 2012. Video-based abnormal human behaviour recognition – a review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), pp. 865–878.

Rajesh, P., Geetha, M. K. and Ramu, R., 2013. Traffic density estimation, vehicle classification and stopped vehicle detection for traffic surveillance system using predefined traffic videos. *Elixir Computer Science and Engineering*, pp. 13671–13676.

Ravanbakhsh, M., Nabi, M., Mousavi, H., Sangineto, E. and Sebe, N., 2016. Plug-and-play CNN for crowd motion analysis: An application in abnormal event detection. *arXiv preprint arXiv:1610.00307*.

Reddy, V., Sanderson, C. and Lovell, B. C., 2011. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. *CVPR 2011 Workshops*, Colorado Springs, CO, 2011, pp. 55-61.

Roshtkhari, M. J. and Levine, M. D., 2013. An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. *Computer Vision and Image Understanding*, 117(10), pp. 1436–1452.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Sabokrou, M., Fathy, M., Hoseini, M. and Klette, R., 2015. Real-time anomaly detection and localization in crowded scenes. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 56–62.

Sakurada, M. and Yairi, T., 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pp. 4:4-4:11.

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

Shi, X. et al., 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In: Cortes, C., Lawrence N. D., Lee D.

D., Sugiyama M., and Garnett R. (eds.). *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pp. 802—810.

Simonyan, K. and Zisserman, A., 2014a. Two-stream convolutional networks for action recognition in videos. In: *Advances in Neural Information Processing Systems*, pp. 568–576.

Simonyan, K. and Zisserman, A., 2014b. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ImageNet Challenge*, pp. 1–10.

Springenberg, J. T., Dosovitskiy, A., Brox, T. and Riedmiller, M. A., 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, pp. 1929–1958.

Tian, Y., Feris, R. S., Liu, H., Hampapur, A. and Sun, M. T., 2011. Robust detection of abandoned and removed objects in complex surveillance videos. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(5), pp. 565–576.

Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M., 2015. Learning spatiotemporal features with 3D convolutional networks. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497.

Varadarajan, J. and Odobez, J. M., 2009. Topic models for scene analysis and abnormality detection. In: *2009 IEEE 12th International Conference on Computer Vision Workshops*, ICCV Workshops, pp. 1338–1345.

Vu, T., Osokin, A. and Laptev, I., 2015. Context-aware CNNs for person head detection. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2893–2901.

Wang, T. and Snoussi, H., 2013. Histograms of optical flow orientation for abnormal events detection. In: *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 45–52.

Wang, Y., Wang, D. and Chen, F., 2013. Abnormal behavior detection using trajectory analysis in camera sensor networks. *International Journal of Distributed Sensor Networks*, 10(1), Article ID 839045.

Xiao, T., Zhang, C., Zha, H. and Wei, F., 2015. Anomaly detection via local coordinate factorization and spatio-temporal pyramid. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9007, pp. 66–82.

Xu, D., Yan, Y., Ricci, E. and Sebe, N., 2017. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding 156 (Supplement C)*, pp. 117 – 127.

Yuan, J., Liu, Z. and Wu, Y., 2009. Discriminative subvolume search for efficient action detection. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2442–2449.

Zagoruyko, S. and Komodakis, N., 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhao, B., Fei-Fei, L. and Xing, E. P., 2011. Online detection of unusual events in videos via dynamic sparse coding. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3313–3320.

Zhong, H., Shi, J. and Visontai, M., 2004. Detecting unusual activity in video. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 819–826.

Zhou, B., Wang, X. and Tang, X., 2011. Random field topic model for semantic region analysis in crowded scenes from tracklets. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3441–3448.

Zhou, S., Shen, W., Zeng, D., Fang, M., Wei, Y. and Zhang, Z., 2016. Spatialtemporal convolutional neural networks for anomaly detection and localization in crowded scenes. *Signal Processing: Image Communication 47(Supplement C)*, pp. 358 – 368.

Zhou, S., Shen, W., Zeng, D. and Zhang, Z., 2015. Unusual event detection in crowded scenes by trajectory analysis. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Institute of Electrical and Electronics Engineers Inc., vol. 2015-August, pp. 1300–1304.