PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

ABOU BEKR BELKAID UNIVERSITY OF TLEMCEN

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

SPECIALTY: MODELE INTELLIGENCE ET DECISION

ANOMALY DETECTION IN VIDEO SURVEILLANCE

A thesis submitted in partial fulfilment of the requirements for the award of the Degree of Master of Science in Intelligent and Decision-Making Models

Before the jury

President:	Dr. Mourtada Benazzouz	Lecturer, University of Tlemcen
Examiner:	Mr. Bricksi-Nigassa Amine	Lecturer, University of Tlemcen
Supervisor:	Dr. Berrabah Sidahmed	Lecturer, University of Tlemcen

Presented by: Boateng Godfred Kyeremeh

June 30, 2024

Abstract

With the rapid advancement and widespread adoption of city monitoring systems, surveillance videos have become increasingly prevalent. Traditional video analysis methods require constant human supervision to identify abnormal events, a process that is both arduous and time-consuming. Consequently, the development of automatic video anomaly detection systems holds substantial practical significance, offering a means to significantly reduce the human resources necessary for video monitoring. This thesis introduces an innovative deep learning method aimed at enhancing video anomaly detection through the use of a spatial autoencoder combined with convolutional Long Short-Term Memory (ConvLSTM) networks. Additionally, it explores a potential framework for the application and expansion of this method to various video sources.

Video anomaly detection involves identifying and classifying unusual events or emergencies that deviate from standard, normal, and expected behaviour. The core challenge in this task lies in effectively extracting spatial and temporal features. The proposed method in this thesis utilizes a spatiotemporal autoencoder to capture both spatial and temporal features within a unified framework. The autoencoder is trained to reconstruct normal video frames accurately, and anomalies are detected based on significant reconstruction errors.

For capturing temporal dynamics, the model employs ConvLSTM networks, which are well-suited for learning temporal dependencies in video data. This combination allows the model to learn intricate patterns and dependencies within the video sequences. Although these enhancements increase model complexity, potentially complicating the training process, this issue is addressed by implementing a clip-based video processing method. This approach enhances training efficiency and mitigates computational demands.

Overall, the proposed deep learning method and the innovative spatiotemporal autoencoder with ConvLSTM present a significant advancement in automatic video anomaly detection, promising more efficient and accurate surveillance solutions. This thesis contributes to the development of robust, automated systems capable of handling diverse and complex video data, paving the way for improved social monitoring and security applications.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Berrabah Sidahmed, for his invaluable guidance, support, and encouragement throughout the course of this project. His expertise and insights were instrumental in shaping this work.

I am profoundly grateful to my family for their unwavering support. To my father, Kyere-Boateng Richard, and my mother, Agyemang Patience, thank you for your constant encouragement and belief in my abilities. To my sister, Evelyn, and my little brother, Nana, your love and support have been a source of strength for me.

I would also like to extend my heartfelt thanks to my friends, Gideon, Bonney, Farouk, Anna, and Linda, for their friendship and support during this journey. Additionally, I am thankful to all my classmates for their camaraderie and assistance.

Your collective support and encouragement have made this journey possible, and I am truly grateful to have you all in my life.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

ConvLSTM	Convolutional Long Short-Term Memory
CNN	Convolutional Neural Networks
LSTM	Long Short-Term Memory
3D CNN	3D Convolutional Neural Networks
RBM	Restricted Boltzmann Machines
BM	Boltzmann Machines
RAE	Regularized AutoEncoders
SAE	Sparse AutoEncoder
CAE	Contractive AutoEncoder
KL	Kullback-Leibler
ConvAE	Convolutional autoencoder ConvAE
RNN	Recurrent Neural Network
HMM	Hidden Markov Models
BS	Background Subtraction
DBSCAN	Density-based Spatial Clustering of Applications with Noise
TV	Total Variation
VAD	Video Anomaly Detection
ReLU	Rectified Linear Unit
FC-LSTM	Fully Connected LSTM
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
EER	Equal Error Rate
GPU	Graphical Processing Unit
TPR	True Positive Rate
FPR	False Positive Rate
AUROC	Area Under the Receiver Operating Characteristic Curve

Chapter 1

Introduction

This chapter commences by providing an overview of the foundational background and underlying motivations that form the basis of the thesis. Following this, it articulates the aims and objectives pursued within the study, further elaborating on the outlined contributions. Finally, the chapter concludes with a succinct summary of the thesis components, offering a clear delineation of its overarching scope and structure.

1.1 Background

The use of surveillance cameras for the early detection of anomalous human behaviours is critical for crime prevention, counterterrorism, and ensuring public safety. These systems are essential in various environments, from public places to private institutions, where timely human intervention is necessary. However, this task is inherently challenging due to the labour-intensive and continuous attention required, which can lead to human errors and inefficiencies. According to[1], abnormal events occur only 0.01% of the time, meaning that 99.9% of surveillance time is wasted on monitoring normal activities, leading to unnecessary storage costs and significant human oversight burdens.

Surveillance videos contribute substantially to the realm of big data, characterized by their unstructured nature and the sheer volume of data generated. As [2] point out, manual surveillance is tedious and time-consuming, especially in crowded public places where various forms of security threats, such as theft, violence, and potential explosions, must be monitored. The detection of anomalous activities in such environments is particularly complex due to real-world constraints like occlusion, varying lighting conditions, and the dynamic nature of crowds.

The primary challenge in human anomaly detection within surveillance footage lies in the massive volume of data produced, making it difficult to identify rare abnormal events. Furthermore, defining what constitutes an anomaly is inherently subjective and context dependent. For instance, behaviour considered normal in one setting, such as walking on a subway platform, might be perceived as suspicious in another, such as loitering in a shopping mall. This variability in human perception complicates the development of universally applicable anomaly detection algorithms.

Machine learning methods have shown promise in recognizing actions within labelled datasets but often struggle in highly occluded or crowded scenes. The high cost and impracticality of labelling every possible event further limit the effectiveness of supervised approaches in real-world applications. Successful anomaly detection requires models that can identify patterns and irregularities with minimal supervision. Figure 1.1 highlights various anomalies observed in society, emphasizing the importance of detecting deviations from usual behaviour patterns.

Recent advancements have explored treating anomaly detection as a binary classification problem, achieving notable success despite the scarcity of abnormal event footage. Alternative methods leveraging spatiotemporal features, dictionary learning, and autoencoders have been developed to address these challenges.

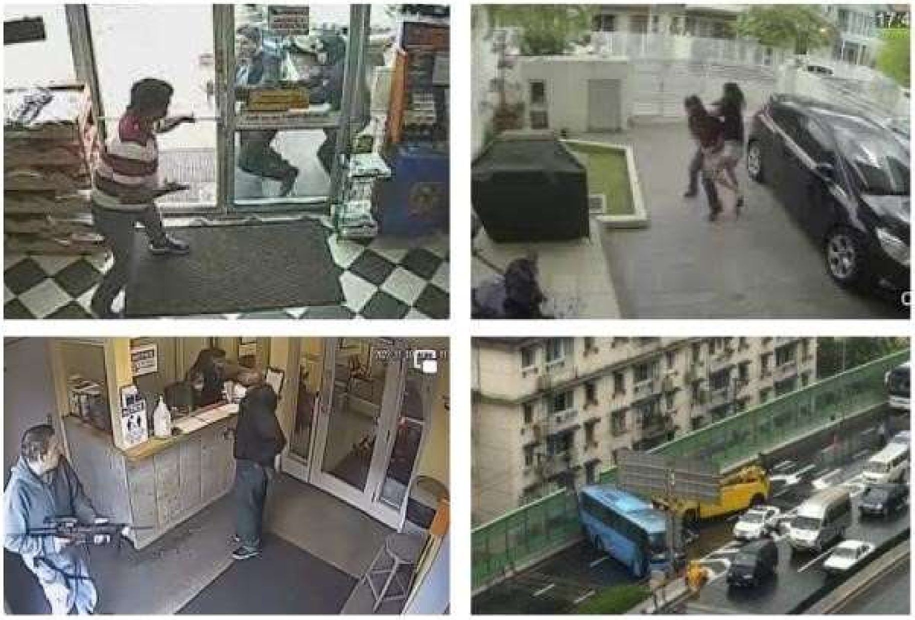


Figure 1.1 *Examples of Anomalies in Societal Behaviour*

Deep learning methods have emerged as powerful tools for video anomaly detection, leveraging their ability to automatically extract relevant features from raw data and generalize well to diverse surveillance scenarios. As illustrated in Figure 1.2, deep learning consistently outperforms traditional approaches as data volume increases, offering superior performance and scalability [3]. Recent advancements have further enhanced anomaly detection by treating it as a binary classification problem and utilizing minimally supervised models with spatiotemporal features, dictionary learning, and autoencoders. These approaches enable efficient surveillance systems to handle large data volumes with minimal supervision and adapt to diverse contexts for effective threat detection and response.

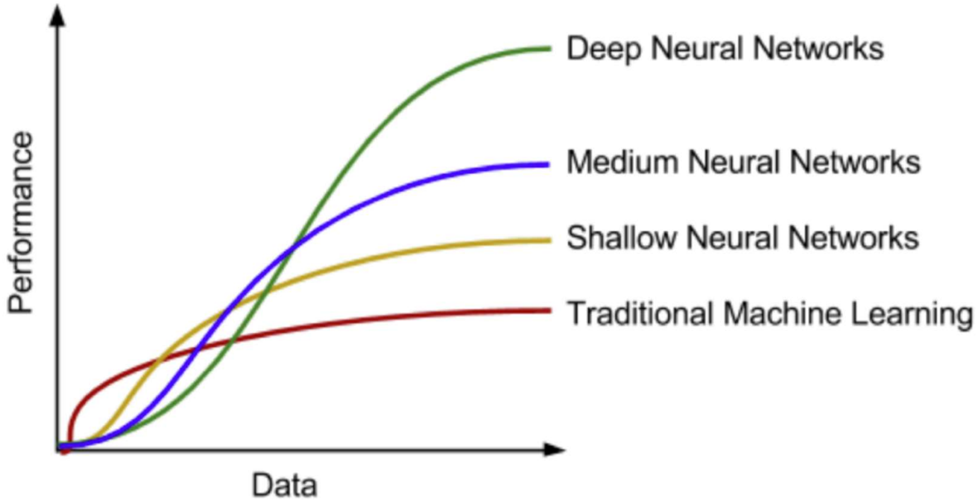


Figure 1.2 *Comparison of Performance between Deep Learning and Traditional Methods with Increasing Data Size [3]*

1.2 Motivation

The motivation for this research stems from the growing need for efficient and scalable automated surveillance systems due to the increasing amount and complexity of video data. Traditional manual monitoring is labour-intensive and prone to errors, emphasizing the need for advanced systems that can accurately detect anomalies in real-time. The main aim is to enhance security and safety across various public and private areas, from preventing threats in public transportation to ensuring smooth operations in industrial settings. Recent advancements in CNNs and LSTM networks show great potential in addressing these challenges. CNNs have proven successful in image recognition and video analysis [4]. LSTMs are good at understanding patterns over time, making them useful for analysing video data.

Furthermore, effective anomaly detection has significant economic and societal benefits, such as cost savings and improved quality of life. It can prevent incidents and ensure operational efficiency, leading to substantial cost savings [5]. Enhanced security and safety measures also contribute to a better quality of life [6].

Developing advanced anomaly detection systems not only meets current security needs but also prepares surveillance infrastructure to handle future threats.

1.3 Problem Statement

The current state of anomaly detection in surveillance videos presents a significant gap between existing methods and the desired goal of accurate and efficient identification of abnormal human behaviour. Existing approaches often struggle to effectively capture nuanced patterns indicative of anomalies, leading to suboptimal detection accuracy and increased false positives. This discrepancy hampers the effectiveness of surveillance systems in ensuring public safety and security [7]. Therefore, there is a pressing need to develop an advanced anomaly detection framework that leverages cutting-edge deep learning techniques, such as Autoencoders and LSTM networks. These techniques have shown promise in enhancing the ability to accurately discern subtle deviations from normal behaviour in surveillance videos [8] [9] [10]. Specifically, deep-anomaly networks and spatiotemporal autoencoders can capture intricate patterns in video data, improving anomaly detection performance [8] [9]. Additionally, LSTM networks can model temporal dependencies, making them well-suited for identifying anomalies over time [10] [11]. By addressing these gaps, the proposed framework aims to provide a robust solution that can be implemented across various surveillance scenarios and environments, thereby improving the overall efficacy and reliability of surveillance systems.

1.4 Objective of Study

The objective of this study is to devise and implement an innovative anomaly detection system for surveillance videos, utilizing spatial autoencoder and convolutional LSTM architectures. The primary goal is to construct a robust deep learning framework capable of accurately detecting abnormal events in surveillance videos. By leveraging the capabilities of spatial autoencoders for feature extraction and ConvLSTMs for capturing temporal dependencies, the system aims to achieve superior anomaly detection performance. Furthermore, the study intends to evaluate the efficacy of the developed framework across diverse surveillance datasets, assessing its ability to detect anomalies effectively in various scenarios and environments. Through comprehensive experimentation and analysis, the objective is to establish the practical viability and effectiveness of the proposed anomaly detection system for deployment in real-world surveillance applications.

1.5 Scope and Limitations

This study aims to develop a sophisticated framework for anomaly detection in video sequences using deep learning techniques. Our proposed architecture integrates a spatial autoencoder for extracting spatial features and a temporal encoder-decoder to capture and analyse temporal patterns. This approach is designed to enhance anomaly detection in video streams, with applications in video surveillance, security, and monitoring systems.

However, the scope of this study is limited by several factors. The model is trained exclusively on normal scenes, which may limit its ability to detect a diverse range of anomalies not present in the training data. Additionally, the model's performance is dependent on the quality and variety of the training dataset, and may be affected by changes in lighting, occlusions, and camera angles. Despite these limitations, the proposed model provides a strong basis for future research and advancements in automated anomaly detection.

1.6 Organisation of Thesis

Chapter 2 offers a literature review centred on anomaly detection in surveillance videos. It begins with an overview of anomaly detection, discussing its significance and applications. Section 2.1 delves into techniques and algorithms frequently employed for anomaly detection, followed by Section 2.2, which reviews previous research specifically related to human anomaly detection in surveillance videos. The chapter wraps up with Section 2.3, discussing datasets commonly utilized in anomaly detection studies, highlighting their characteristics and suitability for research purposes.

Chapter 3 details the methodology proposed for human anomaly detection in surveillance videos. It begins with an overview of the proposed approach in Section 3.1, outlining its main objectives and components. Section 3.2 discusses the preprocessing of surveillance videos, while Section 3.3 explores various feature extraction techniques employed in the methodology. Anomaly detection algorithms are covered in Section 3.4, followed by a discussion on evaluation metrics used to assess the performance of the proposed approach in Section 3.5.

Chapter 4 focuses on the implementation of the proposed methodology. It begins with an overview of the Avenue Dataset and the Dataset in Sections 4.1 and 4.2, respectively, detailing their characteristics and relevance to the research. Section 4.3 discusses the data preprocessing steps undertaken to prepare the datasets for analysis, while Section 4.4 provides a description of the hardware and software used in the implementation process. The setting for anomaly detection algorithms is covered in Section 4.5, followed by a detailed explanation of the training and testing procedure in Section 4.6.

Chapter 5 presents case studies, results, and discussions derived from the implementation of the proposed methodology. It starts with the presentation of experimental results in Section 5.1, followed by a comparison of different algorithms in Section 5.2. Section 5.3 delves into the discussion of findings, analysing the outcomes of the experiments conducted. Challenges faced during the implementation process are examined in Section 5.4, followed by a discussion on the real-world applications of human anomaly detection in Section 5.5. The chapter concludes with case studies showcasing the use of anomaly detection in surveillance systems in Section 5.6.

Chapter 6 serves as the conclusion of the thesis, summarizing the key findings and contributions of the study. It discusses possible enhancements to the proposed methodology in Section 6.1 and potential research directions in human anomaly detection in Section 6.2.

Section 6.3 explores emerging technologies and trends in the field, while Section 6.4 provides a summary of key findings. The chapter concludes with a discussion on the contributions of the study and its implications for future research and practical applications in Section 6.5.

Chapter 2

Literature Review

2.1 Overview of Anomaly Detection

Anomaly detection involves identifying deviations from expected data patterns, often called anomalies or outliers, across various domains. The importance of anomaly detection is paramount, as anomalies in data often signify critical actionable information in a wide variety of application domains. The statistical community has been investigating outlier and anomaly detection since the 19th century [5], laying the foundation for subsequent advancements in the field. Over time, various research communities, including machine learning, data mining, and computer vision, have contributed to the development of a diverse range of anomaly detection techniques. While some of these methods are tailored to specific application domains, such as finance or cybersecurity, others exhibit broader applicability across different fields, underscoring the interdisciplinary nature of anomaly detection research [12].

2.1.1 What are anomalies?

Anomalies are essentially patterns within data that deviate from the expected or well-defined notion of normal behaviour, often indicating unusual or unexpected occurrences that may warrant further investigation or attention.

In Figure 2.12, we observe a simple example of anomaly detection, where normal regions are marked as N and anomalies as O. Anomalies are evidently situated outside the bounds of normal behaviour. However, it's important to note that anomalies such as O2 may appear to be close to normal regions.

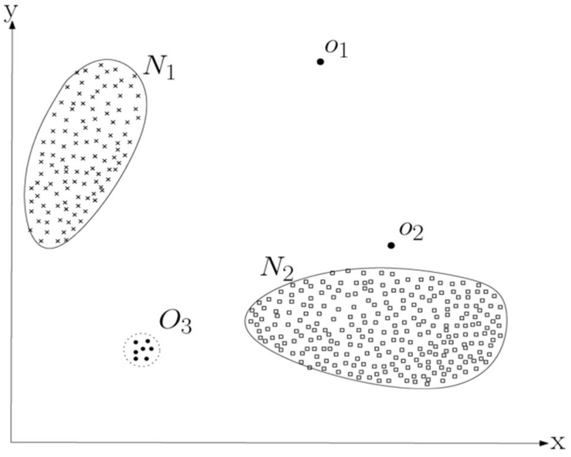


Figure 2.1 *A case of Anomaly Detection [5] (Chandola et al., 2009)*

2.1.2 Types of Video Anomalies

Distinguishing types of anomalies can vary in difficulty depending on factors such as the complexity of the scene, the quality of the video, and the specific characteristics of the anomalies themselves. Some anomalies may be relatively easy to spot, especially if they involve drastic changes or actions that are clearly out of the ordinary. However, others may be more subtle and require careful observation and analysis to differentiate from normal behaviour. Additionally, the presence of noise or other confounding factors in the video can further complicate the task of distinguishing between different types of anomalies. [13] attempts to specify the different types of anomalies in benchmark datasets and practical scenarios.

1. **Appearance Anomalies:** These anomalies involve unusual objects appearing in a scene, like a cyclist on a pedestrian walkway or a boulder on a road. Detecting them only requires examining a local area of a single video frame.

2. **Short-term Motion Anomalies:** These anomalies involve unusual object movements within a scene, such as a person running in a library or lingering around foreign embassies. Detecting them typically only requires observing a short segment of video in a local region. Appearance-only and short-term motion-only anomalies can also be referred to as local anomalies due to their distinct characteristics.

3. Long-term Trajectory Anomalies: These anomalies involve unusual object paths or movements over an extended period, like individuals walking in a zig-zag pattern on a sidewalk or a car weaving in and out of traffic. Detecting trajectory anomalies necessitates examining longer video segments.

4. Group Anomalies: Group anomalies involve unusual interactions between objects in a scene, such as a group of people walking in a formation like a marching band. Detecting these anomalies involves analysing the relationships between two or more regions of the video.

5. Time-of-Day Anomalies: Time-of-day anomalies are unique in that they are defined by when certain activities occur rather than what they entail. For instance, when people enter a movie theatre during the early hours of dawn. Detecting these anomalies usually requires using different models of normal behaviour tailored to different times of the day.

2.1.3 Anomaly Detection Methods

1. Trajectories-based Methods

Trajectory-based methods in anomaly detection operate on the principle that anomalies typically manifest as sudden deviations from regular patterns within a video stream. By analysing a large corpus of videos, these methods can effectively learn the typical trajectories of normal events. When an event occurs that diverges significantly from these learned trajectories, it is flagged as anomalous [14]. Enhancing the efficacy of clustering, two distinct models can be constructed to address spatial changes and dynamic movements within the video [15]**,** thereby improving the accuracy of anomaly detection.

2. Low-level Feature Extraction Methods

Traditional clustering methods encounter complexities when attempting to derive learning trajectories from normal events. Furthermore, clustering-based approaches often exhibit a high dependency on moving objects, which can pose challenges. To address these issues, low-level feature extraction methods emphasize capturing nuanced details within videos, such as changes in grayscale, motion vectors [16], and textures [17]**.** This approach enables a more comprehensive understanding of the underlying dynamics, facilitating more effective trajectory analysis and prediction.

3. Deep Learning Methods

With the surge in video data due to the advancement of smart cities, traditional methods struggle to handle the scale and identify outliers effectively. Consequently, deep learning approaches have gained popularity for such tasks. Among these, utilizing reconstruction error stands out as a prevalent direction for video anomaly detection [18] and [9]. This method involves learning a model of normal videos, where abnormal events exhibit higher reconstruction errors compared to normal events, as they deviate further from the learned normal patterns. Drawing inspiration from image-based models like CNNs, models for video anomaly detection integrate temporal feature processing methods such as LSTM, 3DCNN, and Two-Stream Models. In addition to reconstruction error, some models employ autoencoders for future frame prediction [19] and [20]. By generating anomalous frames, these autoencoders leverage the notion that anomalies diverge from expected patterns, aiding in outlier detection. This approach, exemplified by GANs [21] distinguishes anomalies based on their deviation from predicted frames.

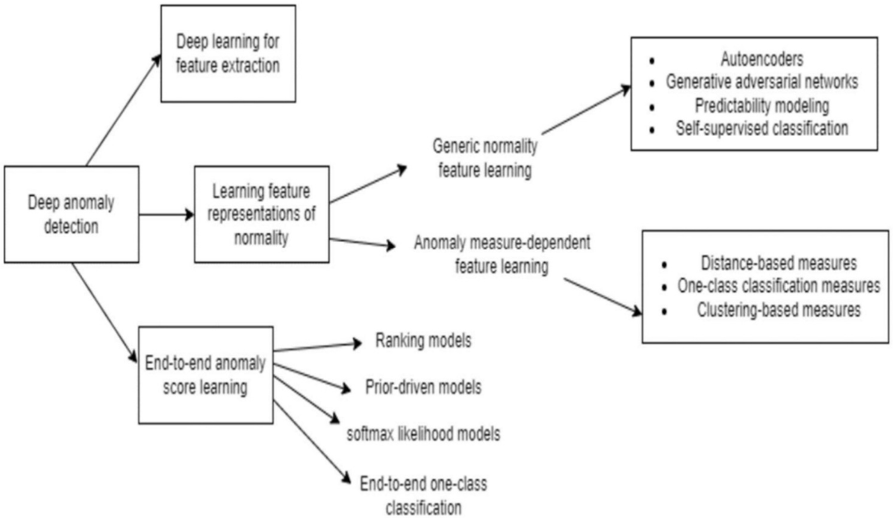

Deep learning for feature extraction
Deep anomaly detection
Learning feature representations of normality
Generic normality feature learning
Autoencoders
Generative adversarial networks
Predictability modeling
Self-supervised classification
Anomaly measure-dependent feature learning
Distance-based measures
One-class classification measures
Clustering-based measures
End-to-end anomaly score learning
Ranking models
Prior-driven models
softmax likelihood models
End-to-end one-class classification

Figure 2.2 *Deep-Learning Based Anomaly Detection*

2.2 Autoencoder (AE)

Autoencoders, are neural network architectures designed to organize, compress, and extract high-level features from data [21]. They are essential for building hierarchical models, facilitating unsupervised learning, and extracting non-linear features [22]. Unlike generative models like RBMs or BMs, which are fully connected, autoencoders are feed-forward neural networks that encode input data into a compressed, semantically meaningful form and then decode it to reconstruct the original input data [23].

The core components of autoencoders include an encoder, a latent feature representation, and a decoder [22]. The encoder compresses the input into a more compact representation, while the decoder reconstructs the original input from this encoding, aiming to learn an informative representation of the data in an unsupervised manner [23]. Specifically, the problem is formulated as finding functions

$A : \mathrm{R}^n \rightarrow \mathrm{R}^p$ (encoder) and $B : \mathrm{R}^p \rightarrow \mathrm{R}^n$ (decoder) that minimize the reconstruction loss function, typically the ℓ2-norm, over the distribution of the input data.

In its vanilla form, an autoencoder comprises an input layer, one or more hidden layers, and an output layer. However, variations and complexities can be introduced, such as convolutional autoencoders or denoising autoencoders, which enhance the model's capabilities. Training an autoencoder involves finding the encoder and decoder functions that minimize the discrepancy between the input and output data, preventing the model from learning the identity function. Strategies such as creating a bottleneck and adding regularization are employed to ensure that the learned representation is both compact and meaningful.

Hyperparameters play a crucial role in the performance of autoencoders. These parameters include the number of hidden layers, the number of neurons in each layer, the size of the latent space, the activation function, and the objective function. Proper selection and tuning of these hyperparameters are essential to optimize the model's performance for a given task.

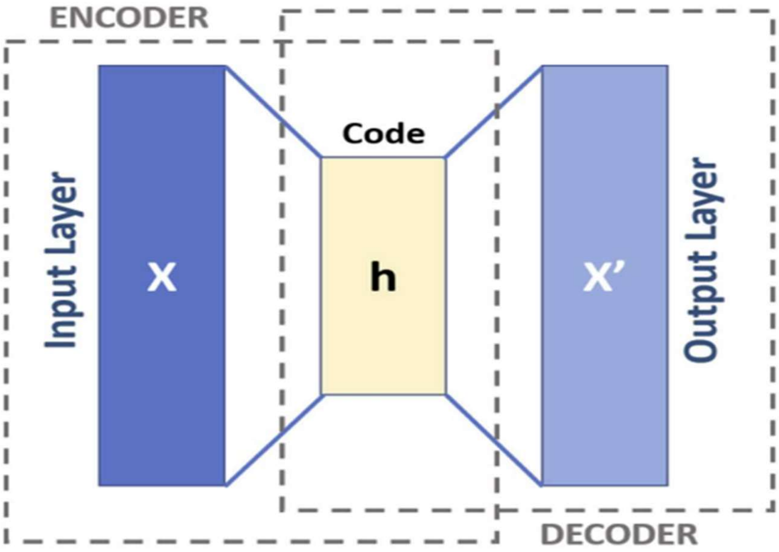


Figure 2.3 Autoencoder Diagram by [24]

The capacity to capture intricate features in an unsupervised manner. With careful design and parameter tuning, they hold promise for various applications in machine learning and data analysis.

2.2.1 Regularized AutoEncoders (RAE)

Regularized autoencoders are a class of autoencoders that incorporate regularization techniques during training to impose constraints on the model's learning process. These constraints help prevent overfitting, encourage the learning of meaningful representations, or impose specific structures on the learned latent space. Here are some common types of regularized autoencoders:

2.2.2 Sparse AutoEncoder (SAE)

Sparse Autoencoder (SAE) is known for its focus on learning a sparse representation of input data by constraining the number of active neural nodes simultaneously. Its primary goal is to minimize the difference between input data and reconstructed data, all while imposing limitations on the sparsity of the latent representation. In an SAE, the loss function consists of two key components: the reconstruction loss and the sparsity loss.

$$\boldsymbol{L_{SAE}} = min(\ ||X - X'||_F^2\ + \lambda KL(p\ ||\ q))$$

where $KL(p\ ||\ q)$ calculates the Kullback–Leibler divergence between a target sparsity parameter (p) and the estimated average activation of each neuron (q) during training, defined as

$$\sum p\ log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right)$$

This combined penalty term encourages the model to acquire a sparse representation, wherein only a limited number of neurons are active for each input.

The Kullback-Leibler (KL) divergence is a measure of the difference between two probability distributions. It quantifies how one probability distribution diverges from another. In the context of neural networks, KL divergence is often used as a regularization term to encourage certain properties in the learned representations, such as sparsity.

2.2.3 Contractive AutoEncoder

CAE [24] is an autoencoder that aims to produce similar representations for similar input data by adding a penalty term to the loss function. This penalty term, based on the Frobenius norm of the Jacobian matrix of the encoder concerning the input data, encourages local stability in the learned representation. The primary objective of the CAE is to minimize the difference between the input data and the reconstructed data while taking the penalty term into account, promoting similarity in representations for similar input data. The overall loss function of a SAE and a CAE include the reconstruction loss, and a penalty term as follows.

$$\boldsymbol{L_{SAE}} = min(||X - X'||_F^2 \ + \lambda KL(p \ || \ q)) \quad (1)$$

where $||X - X'||_F^2$This term represents the reconstruction error, which is the squared Frobenius norm measuring the difference between the original input X and the reconstructed output X'.

$KL(p \ || \ q)$ enforces sparsity in the encoded representation by measuring the difference between the desired average activation p and the actual average activation q.

For the CAE:

$$\boldsymbol{L_{CAE}} (X, X') = min(||X - X'||_F^2 + \lambda||\nabla_X E(X)||_F^2) \qquad (2)$$

where $||X - X'||_F^2$ This term represents the reconstruction error, which is the squared Frobenius norm measuring the difference between the original input X and the reconstructed output X'.

$||\nabla_X E(X)||_F^2$ This term represents the squared Frobenius norm of the Jacobian matrix of the encoded representation concerning the input data X. This norm measures the sensitivity of the encoded representation to small variations in the input data.

2.3 Convolutional autoencoder (ConvAE)

ConvAE employs convolutional layers instead of fully connected layers in both the encoder and decoder. The encoder uses these layers to create a compact representation from input images, while the decoder employs deconvolution layers for image reconstruction. CAEs are particularly effective for image data, as they excel at capturing spatial dependencies, which refer to the patterns and relationships among pixels or locations within individual images or data frames. They find wide-ranging applications in tasks such as image denoising, inpainting, segmentation, and super-resolution.

2.4 Recurrent Neural Network (RNN)

Traditional feedforward neural networks operate under the assumption that all inputs (and outputs) are independent of one another. However, many tasks, especially those involving sequences, require learning temporal dependencies between inputs. For example, in language modelling, the prediction of a word should be informed by the preceding words. RNNs address this by allowing outputs to be influenced not only by the current input but also by the entire sequence of previous inputs. RNNs have shown success in various applications, such as speech recognition and natural language processing [25]. While RNNs theoretically can handle dependencies over long sequences, in practice, they are often limited by the vanishing gradient problem, which restricts their ability to look back more than a few steps [26].

2.5 Long Short-Term Memory (LSTM)

To mitigate the vanishing gradient issue inherent in RNNs, LSTM networks were introduced. LSTMs incorporate a mechanism known as the forget gate, which helps in maintaining a constant error that can be backpropagated through time and layers, thus preserving information over longer sequences. This architecture allows LSTMs to effectively capture long-term dependencies and stack layers to learn higher-level temporal features. The typical LSTM unit consists of several gates and operations summarized in the following equations:

$$Forget\ gate:\ f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \qquad (3)$$

$$Input\ gate:\ i_t = \sigma\left(W_f[h_{t-1}, x_t] + b_i\right) \qquad (4)$$

$$Candidate\ cell\ state:\ \widehat{C_t} = tanh(W_c[h_{t-1}, x_t] + b_c) \qquad (5)$$

$$Cell\ state\ update:\ C_t = f_t \otimes C_{t-1} + i_t \otimes \hat{C}_t \qquad (6)$$

$$Output\ gate:\ o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \qquad (7)$$

$$Hidden\ state:\ h_t = o_t \otimes tanh(C_t) \qquad (8)$$

In the equations, x_t represents the represents the input at time step t, h_t is the hidden state, and C_t is the cell state. W denotes the weight matrices, b are the bias vectors, σ is the sigmoid function, and represents the Hadamard product. The forget gate (equation 3) determines what information to discard from the cell state. The input gate (equation 4) and the candidate cell state (equation 5) together decide what new information to store. The cell state (equation 6) combines old and new information, while the output gate (equation 7) and the final hidden state (equation 8) determine the output and what information to pass to the next time step.

LSTMs have been extensively used in various applications requiring sequence modelling such as natural language processing, speech recognition, and video analysis, demonstrating their ability to handle complex temporal dependencies [27] [28] [29]. A variant of the LSTM, convolutional LSTM is used for this project and is discussed in Chapter 3.

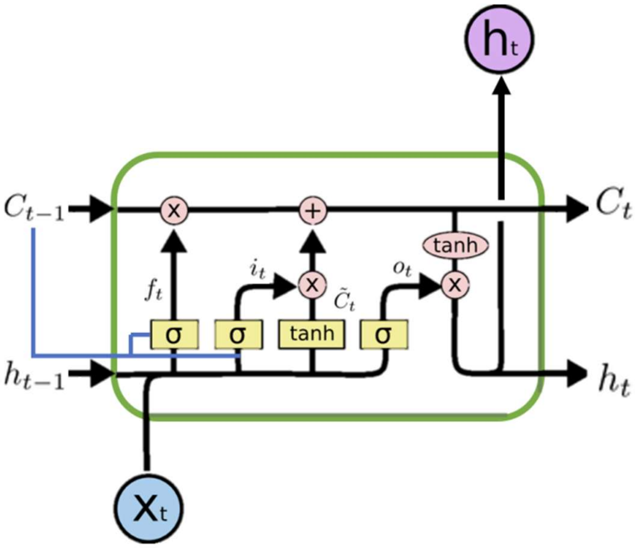


Figure 2.4 *The structure of a typical LSTM unit.*

The blue line indicates an optional peephole connection, enabling the internal state to reference the previous cell state C_{t-1} for improved decision-making [9].

2.6 Techniques for Human Anomaly Detection in Video Surveillance

Human anomaly detection in video surveillance has evolved significantly over the years, transitioning from traditional methods to sophisticated deep learning techniques. This literature review evaluates these various approaches, highlighting their strengths and limitations, and concludes by arguing in favour of spatiotemporal autoencoders as the most effective method.

Traditional Methods

Traditional methods for anomaly detection are primarily based on statistical and machine learning techniques, focusing on feature extraction and pattern recognition.

Background Subtraction (BS)

This is a foundational technique where each pixel of a video frame is compared against a background model to classify it as foreground or background. This method, while simple and effective in controlled environments, often struggles with dynamic backgrounds and illumination changes.

Statistical Models, such as Hidden Markov Models (HMMs) and dynamic Bayesian networks, have been used to model normal behaviour patterns and detect deviations.

[16] implemented an HMM-based approach for unusual event detection. These models, while useful, often fail in real-world scenarios where normal behaviours are highly variable and unpredictable.

Clustering techniques like k-means and DBSCAN can also be employed for anomaly detection by combining them with local motion features. While these methods can be effective, they are computationally intensive and sensitive to parameter selection, making them less practical for real-time applications.

Deep Learning Methods

Deep learning has revolutionized anomaly detection by enabling the automatic extraction of high-level features from raw video data, providing superior performance in handling complex and high-dimensional data.

CNNs have been widely used for feature extraction and classification. [30] demonstrated that CNNs could detect anomalous events by learning spatiotemporal features. However, CNNs require large, labelled datasets for training, which can be a significant limitation.

RNNs, particularly LSTM networks, are adept at modelling temporal dependencies in video sequences. [19] used predictive convolutional LSTMs for anomaly detection, showing improved performance over traditional methods. Despite their effectiveness, RNNs are computationally demanding and can suffer from vanishing gradient issues.

Autoencoders have become powerful tools for unsupervised anomaly detection. By learning to reconstruct input data and identifying anomalies based on reconstruction errors, autoencoders offer a robust solution. Variations like convolutional autoencoders and denoising autoencoders enhance robustness [20] [24] introduced contractive autoencoders, which improve the robustness of learned representations, making them more effective for anomaly detection tasks.

Hybrid approaches combine traditional and deep learning methods to leverage their complementary strengths. For instance, background subtraction can preprocess video frames before a deep learning model performs anomaly detection [31]. This combination can improve detection accuracy and reduce computational complexity.

After reviewing various techniques, it is evident that each has its own set of advantages and challenges. Traditional methods are simple and fast but often fall short in complex, real-world scenarios. Deep learning methods, particularly autoencoders, offer superior performance and robustness but require extensive computational resources and large datasets.

Among the deep learning methods, spatio-temporal autoencoders stand out for their ability to capture both spatial and temporal features of video data, making them particularly well-suited for anomaly detection. [9] demonstrated the effectiveness of spatio-temporal autoencoders in detecting abnormal events by simultaneously learning normal motion and appearance patterns.

In conclusion, given their robust performance and ability to handle complex scenarios without requiring extensive labelled data, spatio-temporal autoencoders are the most effective technique for human anomaly detection in video surveillance. Their ability to learn and adapt to new and unseen data makes them a superior choice compared to other methods.

2.7 Image Regularization

Image regularization is a technique used in image processing and computer vision to improve the quality of images by reducing noise and other artifacts, enhancing important features, and restoring the underlying true image structure. The goal of regularization is to impose certain constraints or priors on the image to achieve a more desirable and stable solution. This is particularly useful in ill-posed problems where the solution may not be unique or stable without additional information.

2.7.1 Key Concepts in Image Regularization

1. **Noise Reduction:**

Regularization methods help to suppress random noise present in images while preserving important details. Techniques like Gaussian smoothing and median filtering are simple forms of noise reduction.

2. **Feature Preservation:**

While reducing noise, regularization methods aim to retain important image features such as edges, textures, and structural details. Techniques like Total Variation (TV) regularization is designed to preserve edges while reducing noise.

3. **Posed Problems:**

Many image processing problems, such as image denoising, deblurring, and super-resolution, are ill-posed, meaning that there isn't a unique solution, or the solution is highly sensitive to input data. Regularization helps to stabilize these problems by incorporating prior knowledge or constraints.

4. **Priors and Constraints:**

Regularization methods incorporate priors (assumptions about the image) or constraints to guide the solution. For example, a common prior is that natural images tend to have smooth regions with sharp edges.

2.7.2 Common Regularization Techniques

Tikhonov Regularization:

Also known as ridge regression in the context of statistical learning, Tikhonov regularization adds a penalty term to the loss function, typically the L2 norm of the image gradient. This helps to smooth the image and reduce noise.

Total Variation (TV) Regularization:

TV regularization minimizes the total variation of the image, which is the L1 norm of the image gradient. This method is effective at preserving edges while smoothing out noise in flat regions.

Laplacian Regularization:

This technique involves using the Laplacian operator to penalize large variations in the image, leading to smoother results. It is commonly used in solving inverse problems like image deblurring.

2.7.3 Applications of Image Regularization

Image Denoising: Removing random noise from images while preserving important details.

Image Deblurring: Restoring sharpness to images that have been blurred due to camera motion or out-of-focus capture.

Super-Resolution: Enhancing the resolution of images by reconstructing high-resolution details from low-resolution inputs.

Inpainting: Filling in missing or corrupted parts of an image by inferring the missing information from the surrounding context.

2.7 Review of Datasets Used in Anomaly Detection

The development and evaluation of systems that detect anomalies in videos depend on the availability of diverse and comprehensive datasets. These datasets provide essential data for training models to differentiate between normal and anomalous events. [32] provides an extensive review of prominent VAD datasets, categorizing them based on specific use cases and types of anomalies. This categorization aids researchers in selecting appropriate datasets that match their application requirements. Well-known datasets, such as the UCSD Pedestrian and CUHK Avenue datasets, are highlighted for their detailed annotations and diverse scenarios, making them pivotal for benchmarking and improving VAD models.

Quality and variety in datasets are crucial, as they should encompass a wide range of normal activities and potential anomalies to ensure models can generalize well to real-world scenarios. A common limitation is that models trained on limited data may fail to recognize anomalies in different contexts or environments. Additionally, the temporal complexity of video data is essential for robust video anomaly detection models. Datasets like the ShanghaiTech Campus dataset, which include longer sequences and more complex scenes, challenge models to capture temporal dependencies effectively, a necessity for real-time surveillance tasks.

[32] also discusses evaluation methods for VAD models, emphasizing frame-level and pixel-level accuracy. These metrics measure how well models identify anomalies in both broad and fine contexts, ensuring comprehensive assessment of their reliability and robustness.

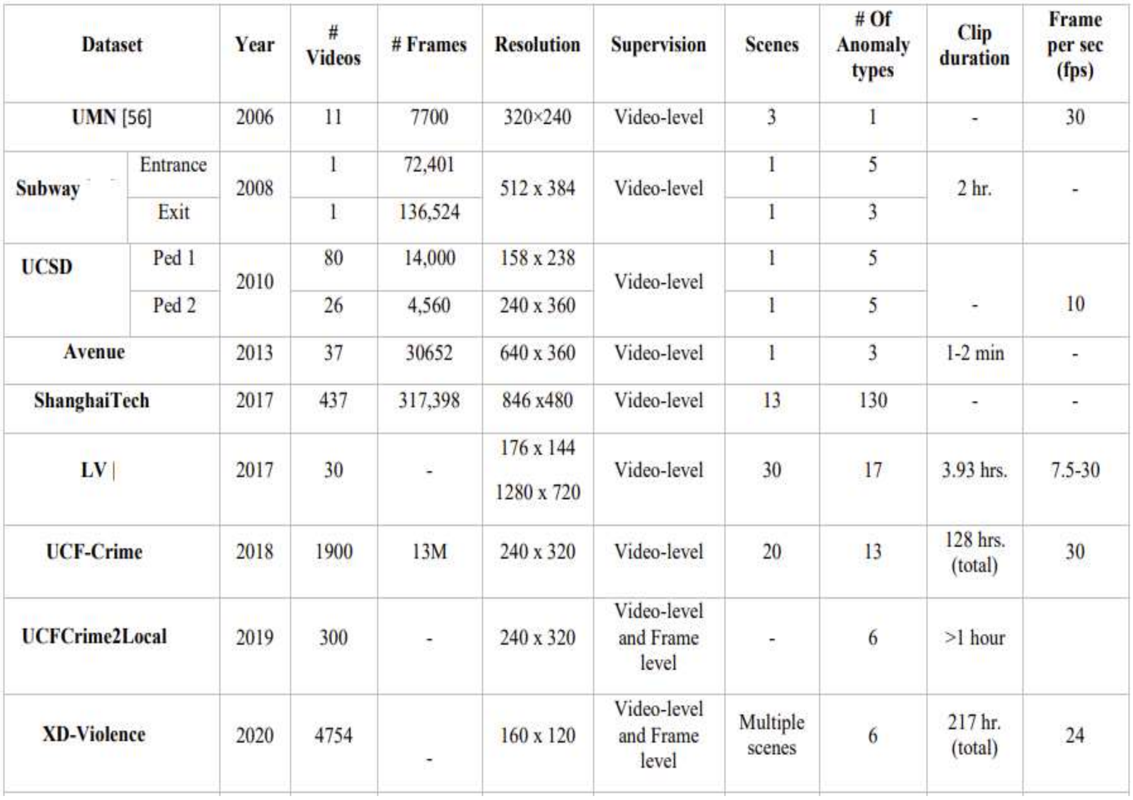
Despite the progress, the task of detecting anomalies in videos faces several challenges including but not limited to:

1. Exploring Abnormality: Defining abnormal moments is difficult due to the lack of a clear distinction between normal and abnormal events. Anomalies in videos are irregular, rare, and can vary depending on the environment, leading to false alarms.

2. Data Imbalance: Anomalies are rare compared to normal instances, leading to data imbalance. Collecting a large number of labelled abnormal instances is challenging.

3. Noise: Distinguishing between noise and real anomalies is a significant challenge, as noise can affect the model's accuracy.

4. Hardware Requirements: Real-time anomaly detection requires high computational power and infrastructure, posing a challenge in handling long and high-quality videos with the latest deep-learning models.



Dataset		Year	# Videos	# Frames	Resolution	Supervision	Scenes	# Of Anomaly types	Clip duration	Frame per sec (fps)
UMN [56]		2006	11	7700	320×240	Video-level	3	1	-	30
Subway	Entrance	2008	1	72,401	512 x 384	Video-level	1	5	2 hr.	-
	Exit		1	136,524			1	3		
UCSD	Ped 1	2010	80	14,000	158 x 238	Video-level	1	5	-	10
	Ped 2		26	4,560	240 x 360		1	5		
Avenue		2013	37	30652	640 x 360	Video-level	1	3	1-2 min	-
ShanghaiTech		2017	437	317,398	846 x480	Video-level	13	130	-	-
LV |		2017	30	-	176 x 144 1280 x 720	Video-level	30	17	3.93 hrs.	7.5-30
UCF-Crime		2018	1900	13M	240 x 320	Video-level	20	13	128 hrs. (total)	30
UCFCrime2Local		2019	300	-	240 x 320	Video-level and Frame level	-	6	>1 hour	
XD-Violence		2020	4754	-	160 x 120	Video-level and Frame level	Multiple scenes	6	217 hr. (total)	24

***Figure 2.5** Popular Benchmark Datasets for Video Anomaly Detection [32]*

Diverse and high-quality datasets are vital for advancing systems that detect anomalies in videos, addressing these challenges is essential for developing robust and reliable models capable of effective real-time anomaly detection.

Chapter 3

Methodology

In this methodology, the focus is on leveraging deep learning techniques to detect anomalies within video sequences. The approach revolves around the concept that abnormal occurrences induce significant deviations between recent frames and older frames within a video stream. To address this, a sophisticated architecture is crafted, comprising a spatial autoencoder for spatial feature extraction and an LSTM-based encoder-decoder for temporal analysis. This integrated setup empowers the model to effectively capture temporal dynamics embedded within the input frame sequence. During the training phase, only video volumes portraying normal scenes are utilized. The primary objective is to minimize the reconstruction error between the input video volume and the reconstructed output volume generated by the trained model.

Once the model is sufficiently trained, it is expected that video volumes depicting normal scenarios will exhibit low reconstruction errors, while volumes containing abnormal events will manifest higher errors. By establishing a threshold on the reconstruction error for each testing input volume, the system becomes adept at identifying anomalous events within video streams.

3.1 Preprocessing

The objective of this stage is to preprocess the raw data into a standardized and model-compatible format. Each frame is extracted from the raw videos and resized to 144 × 144 pixels. To ensure uniformity across input images, pixel values are scaled between 0 and 1. Additionally, each frame is normalized by subtracting it from a global mean image, calculated by averaging pixel values across all frames in the training dataset. Subsequently, the images are converted to grayscale, reducing dimensionality while preserving essential information.

Furthermore, the processed images undergo normalization to achieve a zero mean and unit variance. The input to the model consists of video volumes, each comprising 10 consecutive frames with varying skipping strides. Given the model's substantial parameter count, a sizable training dataset is imperative. Following established practices, data augmentation in the temporal dimension is performed to augment the training dataset's size.

To generate these volumes, frames are concatenated with stride-1, stride-2, and stride-3. For instance, the initial stride-1 sequence comprises frames {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, while the first stride-2 sequence includes frames {1, 3, 5, 7, 9, 11, 13, 15, 17, 19}. Similarly, the first stride-3 sequence encompasses frames {1, 4, 7, 10, 13, 16, 19, 22, 25, 28}. With these preparations, the input is now primed for model training.

3.2 Data Augmentation

Data augmentation in the temporal dimension is a vital technique employed to enrich datasets in the temporal dimension. By generating sequences of frames with varying strides, this augmentation method enhances dataset diversity and facilitates effective capture of temporal information. For each video sequence in the dataset, frames are sampled with different strides to construct input sequences. This strategic approach not only diversifies the training data but also aids in effectively encapsulating temporal dynamics, thereby bolstering model robustness and performance in tasks such as action recognition and anomaly detection.

3.3. Autoencoders

The structure of an autoencoder, revolves around the mapping of an input vector x to an output vector ***r*** **= g(h)** (reconstruction) via an internal representation - a latent space vector h =***f(x)***. It consists of two main components: the **encoder** mapping x to ***h*** and the decoder g mapping ***h*** to r The learning process aims to minimize a loss function **L**, where ***x*** represents an input of the autoencoder and ***g(f(x))*** denotes reconstruction through an internal representation. Essentially, the objective is to reconstruct the original image after undergoing a generalized non-linear compression, as outlined in [33]. While a simple architecture of the autoencoder sufficed for our purpose, handling more complex datasets necessitates incorporating a more sophisticated autoencoder architecture. Autoencoders can be likened to principal component analysis (PCA) under a specific condition. When employing linear activation functions within the autoencoder, the resulting latent variables directly resemble the principal components obtained through PCA. However, it's essential to note that autoencoders typically utilize non-linear activation functions, which are pivotal in enhancing their performance. These non-linear functions enable autoencoders to capture intricate data patterns and relationships more effectively, distinguishing them from the linear transformation approach of PCA.

$$L(\boldsymbol{x},\, g(f(\boldsymbol{x})))$$

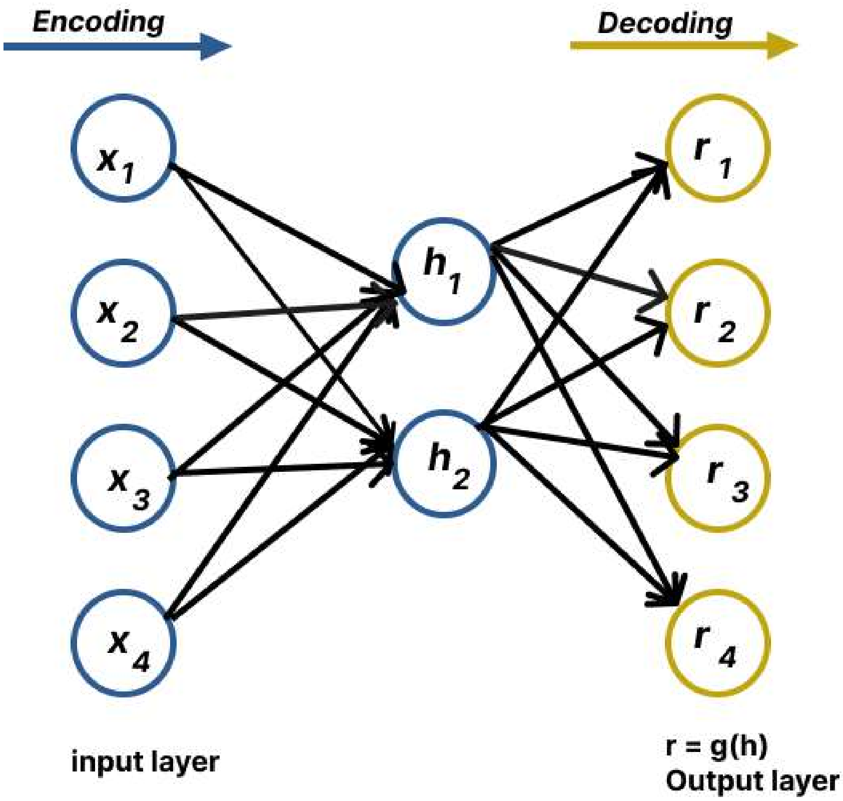


Figure 3.0-1 *Scheme of an Autoencoder*

3.3.1 Spatial Autoencoder (SAE)

The architecture comprises an encoder and a decoder, each containing multiple layers to process the input and generate the reconstructed output.

The encoder starts with a convolutional layer (conv1) with a kernel size of 7x7, which applies 128 filters to the input image, preserving its spatial dimensions. This is followed by a rectified linear unit (ReLU) activation function to introduce non-linearity. Subsequently, an average pooling layer is applied to downsample the feature maps by a factor of 2, reducing the spatial dimensions while preserving essential information. The process continues with additional convolutional layers (conv2 and conv3), each followed by ReLU activation and average pooling, further extracting hierarchical features from the input image and reducing its spatial dimensions progressively.

On the decoder side, the architecture mirrors the encoder's structure, albeit with deconvolutional layers (deconv1 to deconv4) instead of convolutional layers. Each deconvolutional layer upsamples the feature maps using nearest-neighbour spatial upsampling, gradually increasing the spatial dimensions to match those of the original input.

The deconvolutional layers apply a ReLU activation function, similar to the encoder, to introduce non-linearity. Finally, a convolutional layer with a sigmoid activation function is applied to generate the final output, which represents the reconstructed image. The architecture effectively captures spatial information from the input image, compresses it into a latent representation through the encoder, and then reconstructs the original image through the decoder.

1. **Encoder (E):**

- The output of each convolutional layer can be represented as:

$$conv(x) = ReLU(W * x + b)$$

Where W denotes convolutional filter weights, b represents the biases and (*) denotes the convolution operation.

- The average pooling operation reduces the spatial dimensions by a factor of 2, such that;

$$AvgPool(x) = \frac{1}{2}\sum_{i=0}^{1}\sum_{j=0}^{1} x(2i, 2j)$$

2. **Decoder (D):**

The upsampling operation increases the spatial dimensions by a factor of 2, effectively repeating each element:

$$Upsample(x) = \begin{bmatrix} x(\frac{i}{2}, \frac{j}{2}) & 0 \\ 0 & 0 \end{bmatrix}$$

Where ***x(i/2, j/2)*** denotes the ***(i/2, j/2)*** element of ***x***

- The output of each deconvolutional layer can be represented similarly to the encoder's convolutional layer.

$$Deconv(x) = ReLU(W * x + b)$$

3. **Final Output:**
   The final convolutional layer with sigmoid activation generates the reconstructed image:
   $$Output(x) = Sigmoid(W * x + b)$$

4. **Mathematical Notation**
   - Let $\boldsymbol{x}$ represent the input image tensor.

   - The output of the encoder $\boldsymbol{E}$ can be represented as:

   $$E(x) = AvgPool(ReLU(Conv(x)))$$

   - Similarly, the output of the decoder D can be represented as:

   $$D(x) = Deconv\left(Upsample\big(ReLU(Conv(x))\big)\right)$$

   - The final reconstructed output can be represented as:

   $$Reconstruction(x) = Sigmoid\big(Conv(x)\big)$$

3.4 Convolutional LSTM (ConvLSTM)

While spatial features extracted by the SAE provide valuable insights into the static content of individual frames, they fall short in capturing the dynamic temporal nuances inherent in videos. Spatial features alone lack the necessary temporal context for understanding scene evolution and detecting anomalies effectively. This is where the ConvLSTM layers come into play. By integrating ConvLSTM layers into the architecture, the model gains the ability to learn and encode temporal dependencies within the video sequence. The ConvLSTM layers complement the SAE by infusing the model with temporal insight, enabling it to capture temporal patterns, dynamics, and abnormalities inherent in the video data. Together, the SAE and ConvLSTM layers form a symbiotic relationship, with the SAE focusing on extracting spatial features and the ConvLSTM layers specializing in learning temporal dependencies. By combining spatial and temporal features, the model achieves a holistic understanding of the video data, enhancing its ability to detect anomalies accurately and robustly. Thus, while spatial features provide a foundation for understanding the content of individual frames, temporal features are indispensable for capturing the temporal context and dynamics essential for effective anomaly detection in videos. Additionally, a variant of the LSTM architecture, namely ConvLSTM model, was introduced by [34]. ConvLSTM has its matrix operations replaced with convolutions, resulting in fewer weights and yielding better spatial feature maps. The formulation of the ConvLSTM unit can be summarized with equations (7) through (12).

$$A = \sigma\left(W_f * \ [h_{t-1}, x_t, C_{t-1}] + b_f\right) \qquad (9)$$

$$i_t = \sigma(W_i * \ [h_{t-1}, x_t, C_{t-1}] + \ b_i) \qquad (10)$$

$$\hat{C}_t = \tanh(W_c * [h_{t-1}, x_t, C_{t-1}] + b_c \qquad (11)$$

$$C_t = f_1 \otimes C_{t-1} + i_t \otimes \hat{C}_t \qquad (12)$$

$$o_t = \sigma(W_o * \ [h_{t-1}, x_t, C_{t-1}] + \ b_o) \qquad (13)$$

$$h_t = o_t \otimes tanh(C_t) \qquad (14)$$

In this approach, although the equations resemble those numbered (3) through (8) of chapter 2, the inputs are images, and the weight matrices for each connection are substituted with convolutional filters, indicated by the symbol $*$ representing convolution operations. This modification enables ConvLSTM to perform more effectively with images compared to traditional FC-LSTM networks. The key advantage of ConvLSTM lies in its ability to maintain and transfer spatial information over time through each ConvLSTM state, thereby enhancing its suitability for image-based tasks.

3.5 Model Architecture

The architecture takes a sequence of *t* frames as input. Initially, a spatial encoder processes each frame individually. After *t* frames, the features from these frames are concatenated and passed to a temporal encoder for motion analysis. The decoders then reverse this process to reconstruct the original video sequence. See Figure 3.2 for a visual representation.

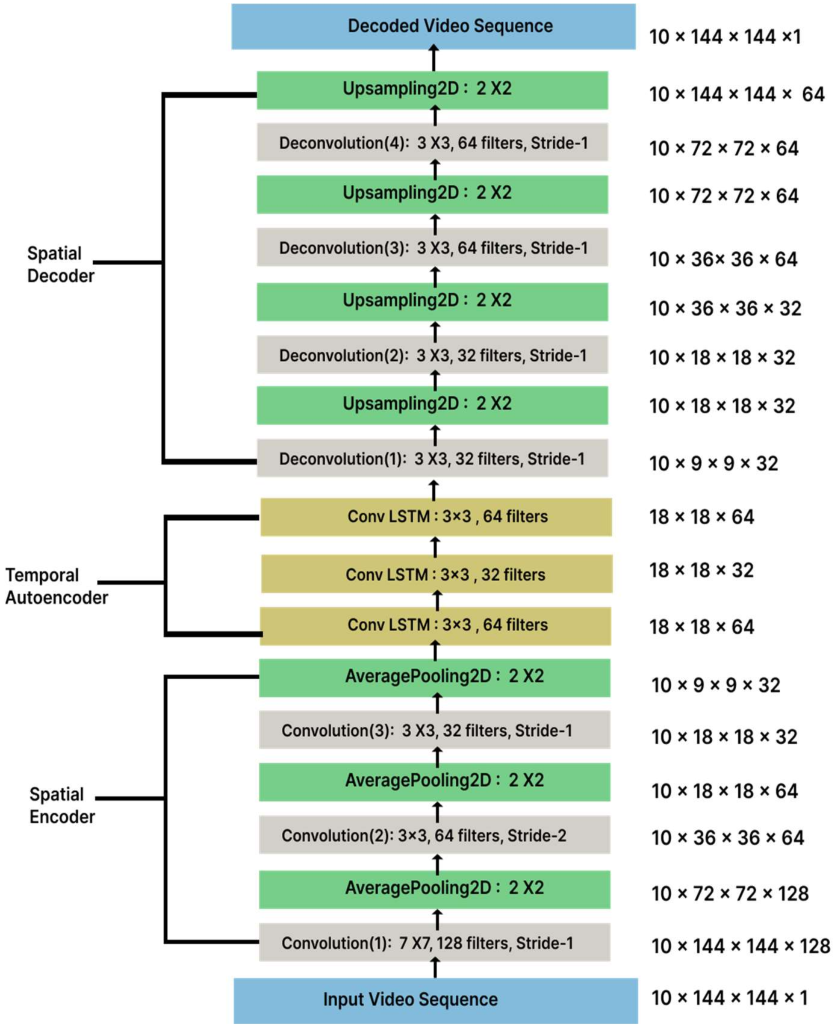


Figure 3.2 *Representation of the architecture of the model*

3.6 Regularity Score

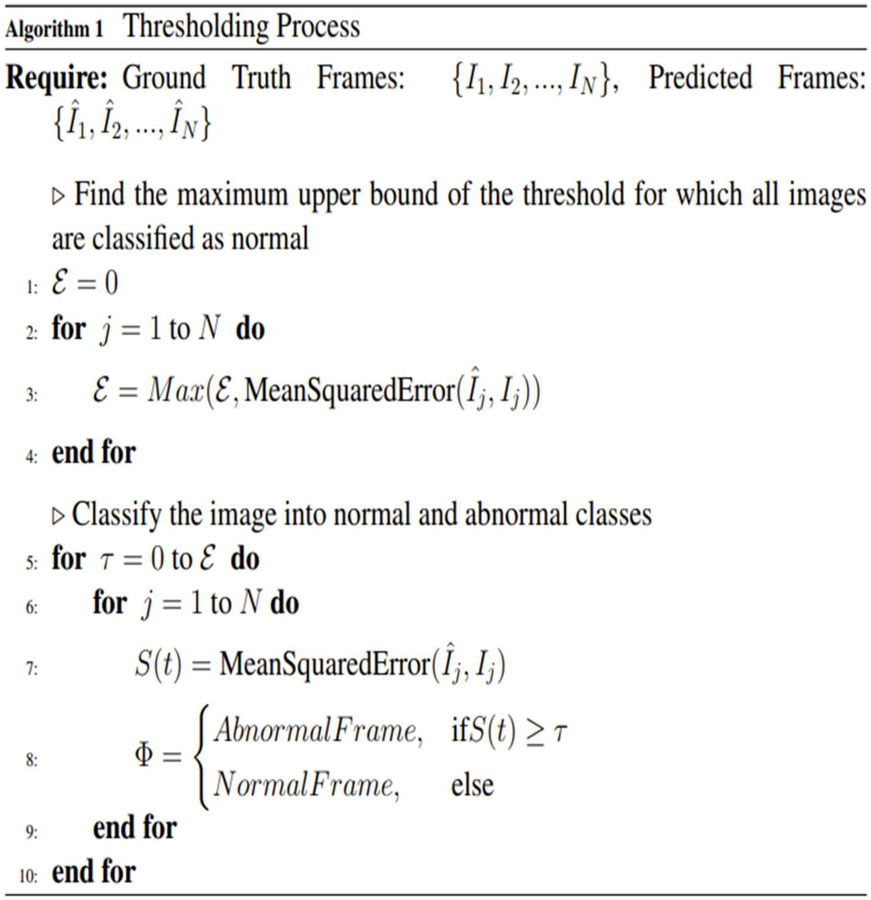
In the pursuit of evaluating model performance, ensuring reliability and effectiveness remains paramount. How do we ascertain if our model meets the task requirements? This question underscores the crucial need for robust evaluation metrics extending beyond mere accuracy assessments. Assessing consistency and stability across diverse scenarios is pivotal. Once the model is trained, its capability to detect abnormal events while maintaining a low false alarm rate is tested using testing data. As a way computing the 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 utilizing learned weights from the spatiotemporal model as discussed in.

$$e(t) = \|x(t) - f_w(x(t))\|_2 \qquad (15)$$

where $\boldsymbol{f_w}$ is the learned weights by the spatiotemporal model. The abnormality score $\boldsymbol{S_a(t)}$ is computed by scaling between 0 and 1. Subsequently, regularity score $\boldsymbol{S_r(t)}$ can be simply derived by subtracting abnormality score from 1.

3.7 Thresholding

The simplicity and effectiveness of determining whether a video frame is normal or anomalous using reconstruction error are remarkable. By comparing the reconstruction error of each frame to a predefined threshold, the system can efficiently classify frames as normal or anomalous. The threshold plays a pivotal role in determining the sensitivity of the detection system: a lower threshold makes the system more sensitive to scene dynamics, potentially triggering more alarms. As a common practice, to quantitatively assess the performance of the detection system, analysing the true positive and false positive rates across different error thresholds to calculate the area under the ROC (AUC). The EER provides a valuable metric, indicating the threshold at which the false positive rate equals the false negative rate, signifying a balanced trade-off between sensitivity and specificity. This comprehensive evaluation framework enhances our understanding of the detection system's performance, enabling fine-tuning and optimization for specific application requirements.



Algorithm 1 Thresholding Process

```
Require: Ground Truth Frames: ${I_1, I_2, ..., I_N}$, Predicted Frames: ${\hat{I}_1, \hat{I}_2, ..., \hat{I}_N}$
    ▷ Find the maximum upper bound of the threshold for which all images are classified as normal
$\mathcal{E} = 0$
for $j = 1$ to $N$ do
    $\mathcal{E} = Max(\mathcal{E}, \text{MeanSquaredError}(\hat{I}_j, I_j))$
end for
    ▷ Classify the image into normal and abnormal classes
for $\tau = 0$ to $\mathcal{E}$ do
    for $j = 1$ to $N$ do
        $S(t) = \text{MeanSquaredError}(\hat{I}_j, I_j)$
        $\Phi = \begin{cases} AbnormalFrame, & \text{if} S(t) \geq \tau \\ NormalFrame, & \text{else} \end{cases}$
    end for
end for
```

The model's output is Φ, the set of normal and abnormal frames.

Chapter 4

Experimental Walkthrough

This chapter details the datasets utilized, the experimental setup, the model parameters, and the hardware and software employed. An overview of the datasets used to train and test the proposed method is provided. Additionally, the model parameters are detailed, along with the specifications of the hardware and software used in the experiments.

4.1 Datasets

This section outlines the datasets used for training and testing the proposed architecture. Three widely recognized benchmarking datasets were utilized to evaluate the method comprehensively: UCSD Ped1 and Ped2 datasets, CUHK Avenue. All videos in these datasets are captured from fixed positions. The training videos exclusively contain normal events, while the testing videos include both normal and abnormal events, allowing for robust evaluation of the model's anomaly detection capabilities. These datasets are discussed briefly as follows.

1. **UCSD Pedestrian 1 (Ped1)**

   The UCSD Ped1 dataset features video clips from pedestrian walkways, capturing anomalies involving unknown objects such as bikes and small cars. It comprises 6,800 training frames and 7,200 testing frames at a resolution of 238×158 pixels and is available only in grayscale. Acquired with stationary cameras mounted at an elevation overlooking pedestrian walkways, the dataset captures variable crowd densities ranging from sparse to crowded, with normal settings showcasing only pedestrians. Anomalies arise from non-pedestrian entities circulating in walkways or anomalous pedestrian motion patterns, including bikers, skaters, small carts, and pedestrians traversing grassy areas. The naturally occurring dataset, without staging, is divided into subsets representing distinct scenes, with video footage segmented into clips of around 200 frames. Ped1 clips depict groups of people walking towards and away from the camera, often with perspective distortion, and include 34 training and 36 testing video samples. Ground truth annotation for each clip includes binary flags per frame indicating anomaly presence, with some clips accompanied by manually generated pixel-level binary masks identifying anomaly regions to facilitate algorithmic performance evaluation regarding anomaly localization. Figure 4.1 shows a type of anomaly in the UCSD Ped1 dataset.

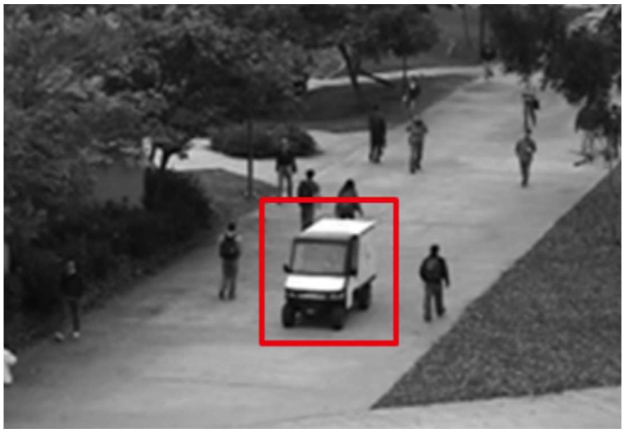


Figure 4.1 *Anomaly (cart) in UCSD Ped1*

2. **UCSD Ped2**

The UCSD Ped2 dataset consists of 2,550 training frames and 2,010 testing frames with a resolution of 360×240 pixels and is also available only in grayscale. Similar to Ped1, the video footage is captured using stationary cameras mounted at an elevation, overlooking pedestrian walkways, and captures variable crowd densities ranging from sparse to crowded. Normal settings showcase only pedestrians, while anomalies are characterized by non-pedestrian entities in walkways or anomalous pedestrian motion patterns, such as bikers, skaters, small carts, and pedestrians traversing grassy areas. The dataset is naturally occurring, without staging, and is divided into subsets representing distinct scenes, with video footage segmented into clips of around 200 frames. Ped2 scenes feature pedestrian movement parallel to the camera plane and comprise 16 training and 12 testing video samples. Ground truth annotation for each clip includes binary flags per frame indicating anomaly presence, with a subset of clips accompanied by manually generated pixel-level binary masks identifying anomaly regions to facilitate algorithmic performance evaluation regarding anomaly localization.

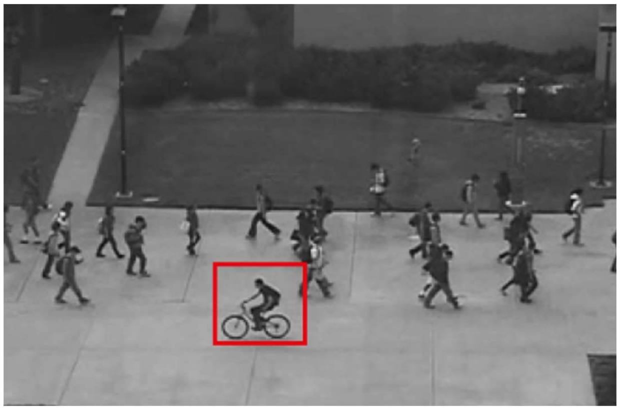


Figure 4.2 *Anomaly (cart) in UCSD Ped2*

3. **CUHK Avenue**

The CUHK Avenue dataset presents a pedestrian-centric focus, emphasizing unexpected pedestrian behaviour and the presence of unknown objects like bicycles as anomalies. With a resolution of 640 x 360 pixels, it comprises 15,328 training frames and 15,324 testing frames across 16 training and 21 test videos. Anomalies in this dataset are primarily characterized by aberrant pedestrian actions such as scrambling, running, and instances involving unidentified objects. The dataset encompasses normal scenes depicting pedestrian movement between staircases and subway entrances, juxtaposed with abnormal events like running, walking in opposite directions, and loitering. Challenges within Avenue include camera shakes, outliers in the training data, and infrequent appearances of certain normal patterns. Despite these challenges, it serves as a valuable resource for developing algorithms capable of detecting and analysing pedestrian behaviour in complex urban environments.

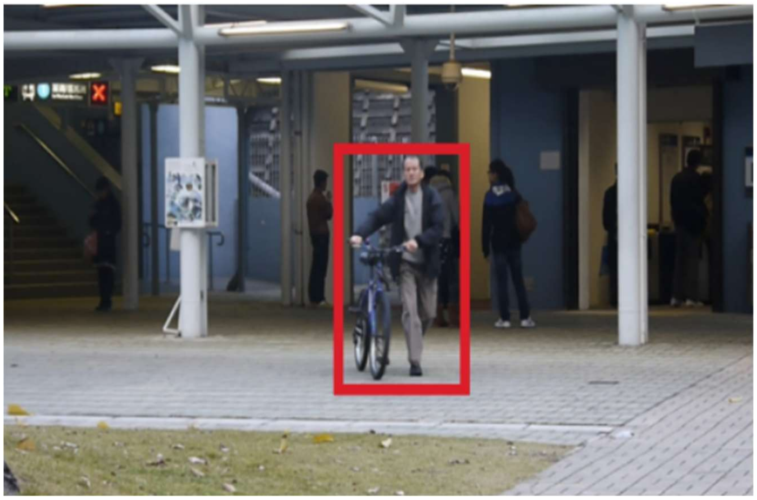


Figure 4.3 *Anomaly in CUHK Avenue (bicycle)*

Table 1 *Characteristics of Benchmark Datasets Used*

Dataset	Year	#Training	#Testing	#Scenario	Source	Resolution
UCSD Ped1	2010	34	36	1	Surveillance	238 x158
UCSD Ped2	2010	16	12	1	Surveillance	238 x 158
CUHK Avenue	2013	16	21	1	Surveillance	640 x 360

4.2 Data Preprocessing

This section outlines the systematic approach adopted to preprocess the datasets, encompassing frame extraction, image regularization, tensor storage, and data augmentation.

4.2.1 Frame Extraction

The datasets under consideration are predominantly presented in video (.avi) and also .tif format. To facilitate anomaly detection at the frame level, frames were extracted from the videos using OpenCV, aligning with the spatio-temporal nature of the autoencoder architecture. The OpenCV module's video reader was employed for efficient extraction, maintaining the original frames-per-second (fps) rate of both UCSD and CUHK avenue datasets, as detailed in Table 4.2.

Table 2 *FPS rate of Datasets*

Dataset	FPS
UCSD Ped1 and Ped 2	10
Avenue	15

4.2.2 Image Regularization

To ensure uniformity and compatibility across datasets with varying frame dimensions, extracted frames were resized to a consistent resolution of 144x144 pixels. This regularization process not only standardizes the input dimensions but also aligns with the requirements of the model architecture, which operates optimally with uniform-sized inputs.

4.2.3 Data Augmentation

In pursuit of refining model performance, data augmentation techniques were employed, including the introduction of noise into the training data. This augmentation strategy serves a dual purpose: enhancing dataset variability to better represent real-world scenarios and mitigating overfitting by exposing the model to a wider range of situations. By integrating noise augmentation, the architecture’s, demonstrates improved efficacy in identifying anomalies within surveillance videos, thereby advancing the anomaly detection.

4.2.3 Tensor Storage

Following image regularization, frames were transformed into tensors, a data storage format optimized for GPU computation. Leveraging the parallel processing capabilities of GPUs, tensors facilitate expedited model training and inference while minimizing memory overhead. This streamlined data format empowers researchers to tackle complex anomaly detection tasks with enhanced computational efficiency.

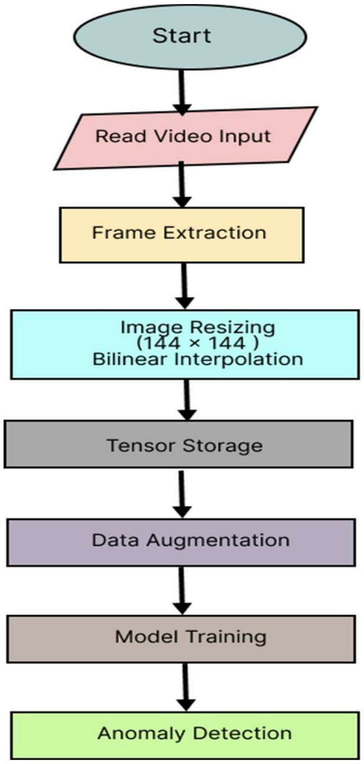


Figure 4.4 *Pipeline of the Proposed Architecture*

4.3 Model Parameters

The model is trained to minimize the reconstruction error of input volumes. We utilize the Adam optimizer, allowing it to dynamically adjust the learning rate based on the model's weight update history. Training is performed using mini batches of 64, with each training volume undergoing a maximum of 50 epochs. Alternatively, training halts if the reconstruction loss of validation data fails to decrease for 10 consecutive epochs. The hyperbolic tangent function is chosen as the activation function for both the spatial encoder and decoder. We refrain from using the RELU activation due to its unbounded activated values, which could disrupt the symmetry between the encoding and decoding functions.

4.4 Hardware and Software Specifications

The machine used for experimentation was equipped with 16GB of RAM and operated on Windows 12 with an Intel Evo processor. Due to the large number of frames and the limitations of the machine's CPU, which was unable to handle the processing load and would crash, Google Colab was utilized to take advantage of the T4 GPU. The entire codebase was written and tested using the TensorFlow framework. The datasets were organized into two folders: the training folder, containing only normal images, and the testing folder, containing the test videos. The complete code was developed and tested using the TensorFlow framework. The datasets were extracted and organized into two folders: the training folder, comprising solely of normal images, and the testing folder, housing the test videos. Following training, the models were saved and exported in the .h5 format. Chapter 5 will delve into the analysis of results, accompanied by an exploration of the chosen hyperparameters.

4.5 Hyperparameter Tuning

Hyperparameter tuning is one important aspect of developing an effective machine learning model, as it involves selecting the optimal set of hyperparameters that govern the learning process and model architecture. In this study, hyperparameter tuning was performed systematically to enhance the performance of the ConvLSTM network and the Spatial AE used for video anomaly detection. The primary hyperparameters considered for tuning included the number of ConvLSTM layers, the number of filters in each layer, the filter size, the stride size, the type of activation function, the learning rate, the batch size, and the dropout rate. An extensive grid search approach was employed to explore the hyperparameter space, testing various combinations to identify the configuration that yields the best performance on the validation set.

The tuning process involved:

1. **Layer Configuration:** Experimenting with different numbers of ConvLSTM layers and AE layers to determine the optimal depth for capturing spatial and temporal features. [34] highlight that deeper networks can capture more complex patterns but also risk overfitting.

2. **Filter Parameters:** Adjusting the number of filters and filter sizes in ConvLSTM layers to balance model complexity and computational efficiency. Filters determine the level of detail the model can capture; more filters can capture finer details but increase computational load, as discussed by [35].

3. **Activation Functions:** Evaluating the performance of different activation functions, with a particular focus on the tanh activation function for ConvLSTM layers. Activation functions

like tanh help in controlling the flow of information and maintaining the stability of gradients during training, as noted by [4].

4. Learning Rate: Fine-tuning the learning rate to ensure stable and efficient convergence of the model. The learning rate impacts how quickly the model learns; too high can lead to instability, too low can result in slow convergence, as illustrated by [36].

5. Batch Size: Testing various batch sizes to optimize training time and model accuracy. Batch size affects the model's ability to generalize; larger batches provide more stable updates, but smaller batches can offer better generalization, as explained by [37].

6. Dropout Rate: Introducing dropout layers and adjusting dropout rates to prevent overfitting and improve generalization. Dropout regularizes the model by preventing co-adaptation of neurons.

The hyperparameter tuning was guided by performance metrics such as accuracy, precision, recall, and F1-score on the validation set. The final set of hyperparameters was selected based on achieving the best trade-off between model complexity and performance.

4.6 Model Optimization

Model optimization focuses on refining the trained model to achieve optimal performance and efficiency in detecting anomalies in video data. In this study, several optimization techniques were employed to enhance the ConvLSTM and Spatial AE models' performance.

1. Regularization Techniques: Regularization methods, such as L2 regularization and dropout, were applied to prevent overfitting. Dropout layers were incorporated at various points in the network to randomly deactivate neurons during training, encouraging the model to develop more robust and generalized features.

2. Learning Rate Schedulers: Adaptive learning rate schedulers, such as ReduceLROnPlateau, were utilized to dynamically adjust the learning rate based on the model's performance during training. This approach helped in achieving faster convergence and avoiding local minima, as shown by [36].

3. Early Stopping: Implementing early stopping criteria based on validation loss allowed the training process to halt when the model's performance ceased to improve, thus preventing overfitting and saving computational resources.

4. Gradient Clipping: To address the issue of exploding gradients often encountered in deep recurrent networks, gradient clipping was employed. By capping the gradients during backpropagation, the model maintained stable and efficient learning.

5.Batch Normalization: Batch normalization layers were added to the model to normalize the input features of each layer, which helped in accelerating the training process and improving overall model stability. Batch normalization reduces internal covariate shift and stabilizes learning, as discussed by [38].

6. Data Augmentation: Data augmentation techniques, such as random cropping, flipping, and rotation, were applied to the training data to artificially increase the dataset size and diversity, thus improving the model's ability to generalize to unseen data.

7. Optimization Algorithms: Advanced optimization algorithms, including Adam and RMSprop, were utilized to optimize the model's weights. These algorithms adaptively adjusted the learning rate for each parameter, ensuring efficient convergence, as illustrated by [36].

Through these optimization techniques, the ConvLSTM and Spatial AE models were fine-tuned to achieve superior performance in detecting anomalies in video data. The combination of hyperparameter tuning and model optimization resulted in a robust and efficient model capable of capturing both spatial and temporal features essential for accurate anomaly detection.

Chapter 5

Experimental Results and Discussion

This section discusses the results of the model and tests against other modern video anomaly detection models using a quantitative approach. The chosen evaluation metrics are explained, and a detailed overview of the experimental setup is provided, including the constants and learning parameters used to obtain the results.

5.1 Performance Evaluation Metrics

The anomaly detection performance of the model was assessed using a frame-level criterion. The evaluation metrics employed, namely the AUROC score and the EER, offer comprehensive insights into the model's classification capabilities, regardless of specific threshold settings. The analysis began with the visualization of the ROC curve, which entails examining the TPR and FPR at different threshold levels. Higher AUC and lower EER are better.

$$TPR = \frac{Number\ of\ true\ \ positives}{Number\ of\ actual\ positives}$$

$$FPR = \frac{Number\ of\ false\ \ positives}{Number\ of\ actual\ negatives}$$

5.1.1 AUROC

The AUROC score serves as a valuable measure for assessing a classifier's ability to distinguish between different classes. Visualized as the area under the ROC curve, this metric offers insights into the classifier's performance. A higher AUROC score, approaching 1.0, signifies a strong classifier that effectively separates classes. Conversely, a score close to 0.5 indicates that the classifier performs no better than random guessing, while a score nearing 0.0 suggests that the classifier predicts results opposite to the actual outcome. These distinctions are vividly illustrated in ROC plots, offering a clear understanding of the classifier's predictive capabilities.

5.1.2 EER

This measure offers a comprehensive overview of the model's classification prowess. It focuses on the ratio of misclassified frames within the model. Specifically, in terms of frame-level evaluation, this corresponds to the point where TPR = **I-** FPR. The determination of this score involves interpolation techniques. Typically, a lower Equal Error Rate (EER) is favoured as it indicates greater accuracy.

5.2 Experimental Configuration

Employing a spatiotemporal autoencoder architecture for model training, we utilized the Adam optimizer with hyperparameters including a learning rate of $1\text{x}10^{-4}$, a decay of $1\text{x}10^{-5}$, and an epsilon of $1\text{x}10^{-6}$ to optimize the training process. To assess the model's performance on test data, a sliding window technique with a sequence length of 10 was applied. Subsequently, the reconstruction of these sequences was predicted using the trained model, and the reconstruction cost of each sequence was computed. Regularity scores were then derived based on the deviation of each sequence's reconstruction cost from the norm, facilitating the identification of anomalies within the test data. Figure 5.1 gives an overview of how it all works.

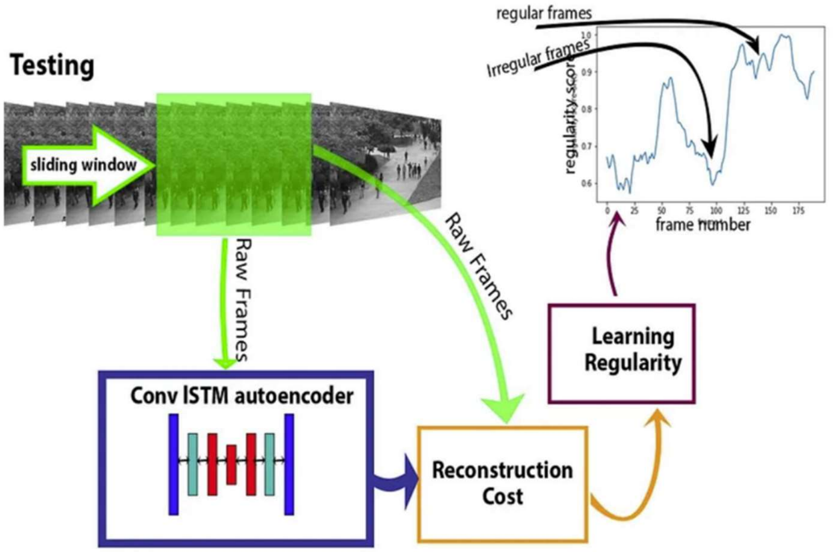


Figure 5.1 *Sliding Window Technique*

5.3 Regularity Score

At the testing phase, each testing video consists of 200 frames in the case of ped1 and 150 in ped 2. Unlike ped1, in the testing phase for, videos from the dataset undergo processing to extract consecutive 10-frame sequences using the sliding window technique. Each video is assessed individually. If a video comprises fewer than 150 frames, the frames are padded by repeating the last frame until there are 150 frames. Conversely, if a video contains more than 150 frames, the frames are truncated to the first 150 frames. Notably, some videos may consist of 150 frames, while others may have 180 frames. The sliding window technique is employed the extract all consecutive 10-frame sequences. Specifically, for each time step

$\boldsymbol{t}$ ranging from *0 to 190*, we calculate the regularity score $\boldsymbol{S_r(t)}$ for the sequence starting at frame $\boldsymbol{t}$ and ending at frame $\boldsymbol{t} + 9$. The regularity score is calculated as follows.

1. **Pixel-wise Reconstruction Error**
   The reconstruction error for a pixel's intensity value ***I*** at the location (x, y) in frame t of the video is computed using the L2 norm:

$$A = \left\| I(x, y, t) - F_w\big(I(x, y, t)\big) \right\|_2$$

   Here F_w represents the learned model by the LSTM convolutional autoencoder.

2. **Frame-wise Reconstruction Error:**
   The reconstruction error for frame t is calculated by summing up all the pixel-wise errors:

$$e(t) = \sum_{x,y} e(x, y, t)$$

3. **Sequence Reconstruction Cost:**
   The reconstruction cost for a 10-frame sequence starting at ***t*** is obtained by summing the frame-wise errors for the sequence:

$$seq\_reconstruction\_cost(t) = \sum_{t'=t}^{t+10} e(t')$$

4. **Normalization of Reconstruction Costs:**
   To normalize the reconstruction costs, we compute the normalized abnormality score $S_a(t)$:

$$A = \frac{seq\_reconstruction\_cost(t) \; - \; \min(seq\_reconstruction\_cost)}{\max(seq\_reconstrution\_cost)}$$

5. **Regularity Score:**
   The regularity score (t) is derived by subtracting the normalized abnormality score from 1:

$$S_r(t) = 1 - \; S_a(t)$$

After computing the regularity score $\boldsymbol{S_r(t)}$ for each t in the range [0, 190], we visualize $\boldsymbol{S_r(t)}$ to identify and analyse anomalies. A higher regularity score indicates a sequence that closely follows normal patterns, while a lower score highlights potential anomalies. This approach enables effective real-time anomaly detection by leveraging the spatiotemporal autoencoder's ability to reconstruct sequences and identify deviations from the norm.

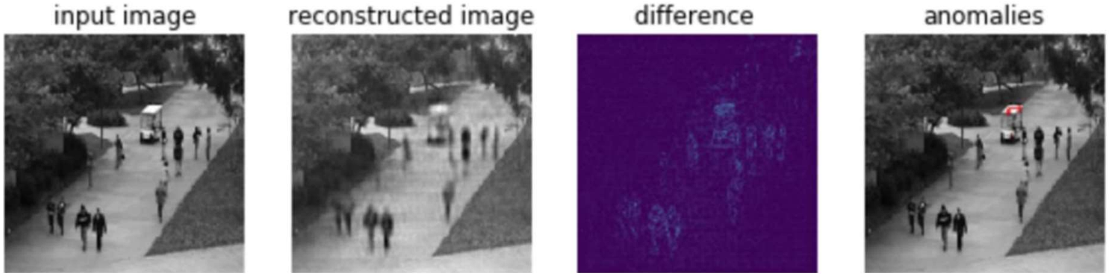


***Figure 5.2** Anomaly Detection in UCSD Ped1 (cart) by model*

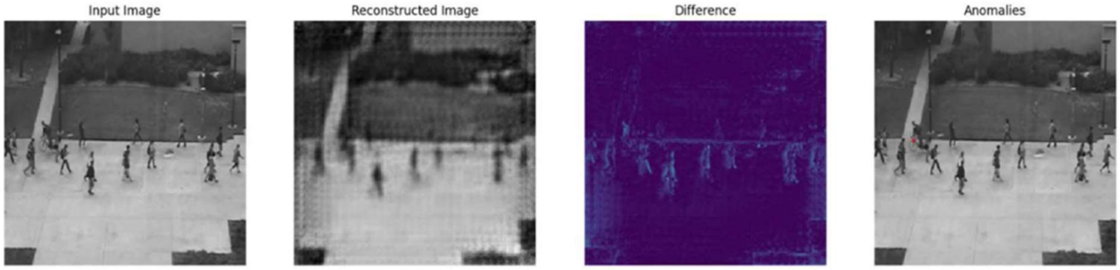


Figure 5.3 *Anomaly Detection in UCSD Ped2 (bicycle) by model*

5.4 Visualisation

The figures below show the plot of the anomaly score (S(t)) across some testing videos in the ped1 and ped2 dataset. Figure 5.4 has a person on a bike at the beginning of the video and hence a drop in the regularity score, it then gets to normal and drops again as in when another person enters the scene on a bicycle as well. In Figure 5.5 a skater is seen during the early part of the video. Towards the end, a person is seen walking on the grass hence the drop in regularity score. Figure 5.6 is a graph of a video from the UCSD ped2 dataset. A person is seen riding a bicycle. Even though he enters the scene around frame 40, There is a drop in regularity score at the very beginning, till the anomaly is evidently seen between frame 60 and 80.

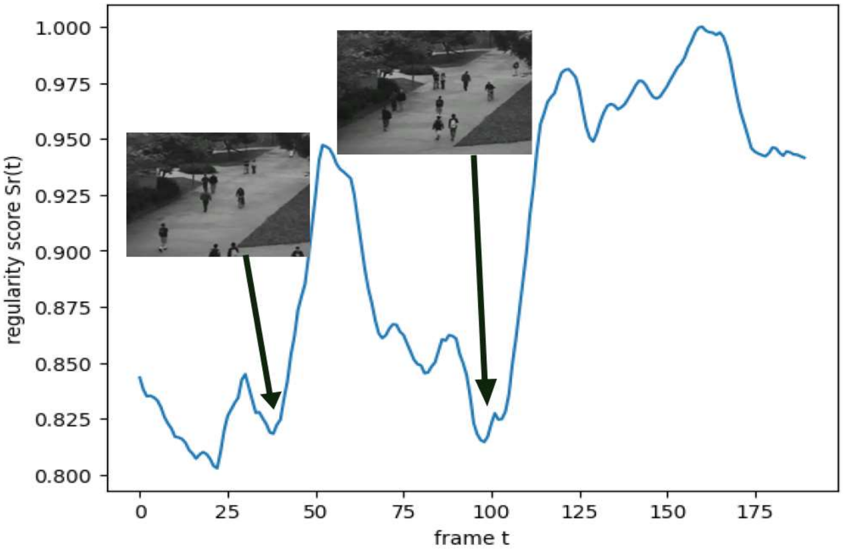


Figure 5.4 *Frame level anomaly UCSD ped1 (bicycle)*

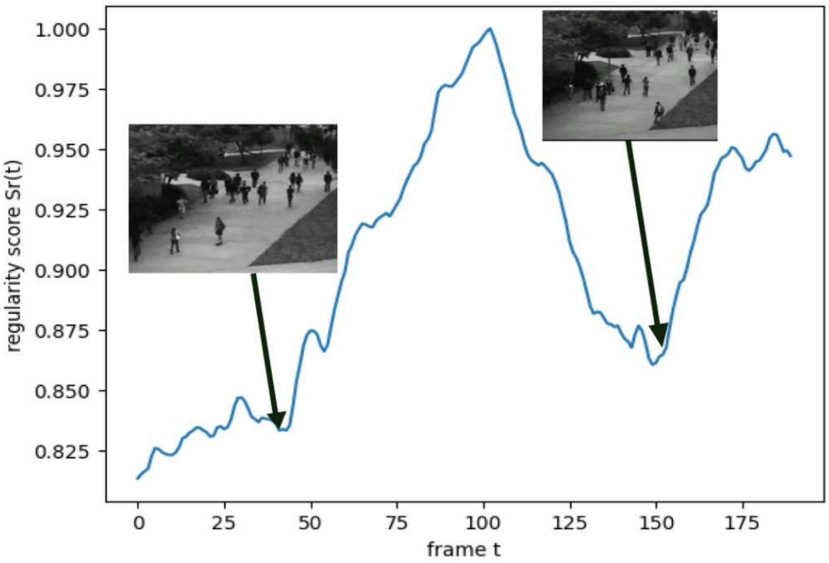


***Figure 5.5** Frame level anomaly UCSD ped1 (skateboard) and (walking on the grass)*

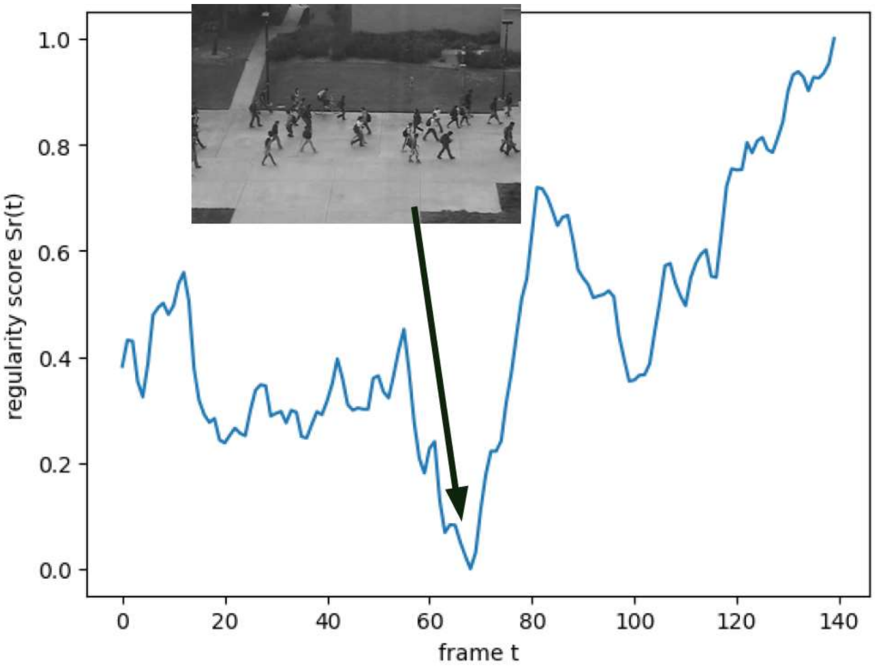


***Figure 5.6** Frame level anomaly UCSD ped 2 (bicycle)*

5.5 Quantitative Analysis

The comparison study, detailed below, demonstrates that our model performs better compared to other models, highlighting its efficiency and capability in real-time anomaly detection tasks. Notably, our model achieves a good performance in ped1 dataset and does considerably well against other methods. Table 5.1: and 5.2 Comparison of the AUROC score for both ped 1 and ped2 dataset.

***Table 3** Comparison of the AUROC score for ped1 and other methods.*

Method	**AUROC (%)**	**EER**
	Ped1	
(Adam et al.) [16]	77.1	38.0
HOFME (Wang et al.) [44]	72.7	33.1
(Mehran et al.) [45]	96.0	-
(Chong et al.) [9]	89.9	12.5
ConvAE (Hasan et al.) [11]	81. 0	27.9
Ours (proposed model)	**84.5**	**19.5**

***Table 4** Comparison of the AUROC score for ped2 and other methods.*

Method	**AUROC (%)**	**EER**
	Ped2	
ConvAE (Hasan et al.) [11]	90.0	21.7
HOFME (Wang et al.) [44]	87.5	20.0
(Chong et al.) [9]	87.4	12.0
(Adam et al.) [16]	-	42.0
(Nawarante et al.) [46]	**91. 1**	**8.9**
Ours (proposed model)	**74.9**	**27.2**

Table 5 *Comparison of the AUROC score for CUHK Avenue and other methods.*

Method	**AUROC (%)**	**EER**
	CUHK	
ConvAE (Hasan et al.) [11]	70.2	25.1
(Chong et al.) [9]	80.3	20.7
(Liu et al.) [20]	85.1	-
Ours (proposed model)	**72.4**	**29.2**

5.6 Analysis of Challenges Faced

In the development and implementation of the spatiotemporal autoencoder for anomaly detection, several challenges were encountered. These challenges spanned across data preprocessing, model architecture design, training, and evaluation. Here is a detailed analysis of these challenges:

1. **Data Preprocessing:**
   Ensuring high-quality and consistent data was a significant challenge. Variability in video frames, such as differences in lighting, resolution, and background noise, impacted the model's performance. Standardizing the input data through normalization and resizing was essential but challenging due to the diverse nature of the datasets.

2. **Hyperparameter Tuning:**
   Selecting optimal hyperparameters, such as learning rate, decay, epsilon, and sequence length, was a time-consuming process. Hyperparameter tuning required extensive experimentation and validation to ensure the model’s robustness and accuracy.

3. **Computational Resources:**
   Training the model on large video datasets required substantial computational resources. Efficiently utilizing these resources while maintaining high training speed and accuracy was a persistent challenge. Specifically, the system's 12.7 GB of RAM and the T4 GPU with 15.0 GB of memory on Google Colab often reached full capacity due to the high volume of frames and the intensive processing demands. To mitigate session crashes, techniques such as reducing the resolution of input frames and using smaller batch sizes were employed. Despite these measures, the model's performance could have been significantly enhanced with sufficient resources, allowing for larger batch sizes, higher resolution frames, and more complex model architectures.

Specifically, the LSTM convolutional autoencoder, due to its sophisticated architecture, required substantial computational resources for both training and inference.

To address the issues and improve the system's stability and performance in my specific case, several strategies were implemented. For data management, downsampling the resolution of input frames significantly reduced memory usage without drastically impacting performance, and frame skipping was employed to lower the computational load by processing a subset of frames instead of every frame. Model optimization involved model pruning to remove redundant neurons and connections, reducing complexity and resource requirements, and parameter tuning to optimize hyperparameters such as batch size, learning rate, and sequence length, balancing performance and resource utilization. Efficient resource usage was enhanced by implementing incremental learning techniques to prevent memory overload and using data generators to load data in smaller batches, effectively managing memory usage. Leveraging the T4 GPU on Google Colab significantly sped up processing and reduced the strain on CPU and RAM. However, with sufficient resources, the model could have been further improved by allowing for larger batch sizes, higher resolution frames, and more complex model architectures, ultimately enhancing its performance.

Chapter 6

Conclusion

6.1 Possible Enhancements to the Proposed Methodology

The proposed spatiotemporal autoencoder methodology has demonstrated considerable efficacy in anomaly detection within video sequences. However, there are several potential enhancements that could further improve the performance and robustness of the model:

Incorporation of Attention Mechanisms: Attention mechanisms have proven effective in improving the performance of various neural network architectures by allowing the model to focus on relevant parts of the input data [47].

Multi-Scale Feature Extraction: Incorporating multi-scale feature extraction techniques can allow the model to capture anomalies occurring at different scales, enhancing the sensitivity of the model to various types of anomalies [39].

Adversarial Training: Utilizing generative adversarial networks (GANs) for training the autoencoder can help in generating more realistic reconstructions, thus improving the anomaly detection performance [21].

Hybrid Models: Combining the strengths of different architectures, such as CNNs for spatial feature extraction and RNNs for temporal feature learning, can potentially improve the model's performance [40].

Enhanced Regularization Techniques: Implementing advanced regularization techniques such as dropout, batch normalization, and L2 regularization can help in preventing overfitting, thereby improving the generalization capabilities of the model.

Data Augmentation: Employing advanced data augmentation techniques can help in generating a more diverse training dataset, which can improve the model's robustness and performance [41].

6.2 Potential Research Directions in Human Anomaly Detection

Human anomaly detection is a rapidly evolving field with numerous research opportunities. Some potential research directions include:

Real-Time Anomaly Detection: Developing models that can detect anomalies in real-time with minimal latency is crucial for applications such as surveillance and safety monitoring.

Cross-Dataset Generalization: Ensuring that models trained on one dataset generalize well to other datasets is a significant challenge. Research could explore transfer learning techniques and domain adaptation strategies to improve cross-dataset generalization.

Integration with IoT Devices: Integrating anomaly detection models with IoT devices could enable widespread deployment of real-time monitoring systems. This integration poses challenges related to computational constraints and energy efficiency, which need to be addressed [42].

Explainable AI: Developing models that not only detect anomalies but also provide explanations for their decisions is essential for gaining user trust and improving the interpretability of the models [43].

Robustness to Adversarial Attacks: Ensuring that anomaly detection models are robust to adversarial attacks is critical for their deployment in security-sensitive applications [21].

6.3 Emerging Technologies and Trends

Several emerging technologies and trends are poised to influence the field of human anomaly detection:

Edge Computing: Incorporating edge computing in anomaly detection systems can enable real-time analysis and decision-making [42].

5G Networks: The advent of 5G technology promises higher data transfer speeds and lower latency, facilitating the deployment of real-time anomaly detection systems.

Advanced Sensor Technologies: Advanced sensor technologies can provide richer data for anomaly detection models, potentially improving their accuracy and reliability.

Quantum Computing: Research in quantum machine learning could lead to the development of more powerful anomaly detection models.

Blockchain Technology: Blockchain technology can enhance the security and transparency of anomaly detection systems.

6.4 Summary of Key Findings

The key findings of this study are summarized as follows:

Effectiveness of Spatiotemporal Autoencoder: The spatiotemporal autoencoder demonstrated significant efficacy in detecting anomalies within video sequences.

Impact of Regularization and Optimization: Implementing advanced regularization techniques and optimizing hyperparameters were important in improving the model’s performance.

Challenges in Computational Resources: The study highlighted the importance of sufficient computational resources for training complex models on large datasets.

Potential for Real-Time Applications: The model shows potential for real-time anomaly detection applications.

6.5 Contributions of the Study

This study makes several contributions to the field of human anomaly detection:

1. Novel Methodology: The study presents a novel spatiotemporal autoencoder methodology for detecting anomalies in video sequences.

2. Comprehensive Evaluation: The methodology was evaluated on multiple benchmark datasets, demonstrating its efficacy and robustness across different scenarios and types of anomalies.

3. Practical Insights: The study provides practical insights into the challenges and solutions related to computational resource constraints.

6.6 Implications for Future Research and Practical Applications

The findings and contributions of this study have several implications for future research and practical applications:

1. Enhanced Model Development: Future research can build upon the proposed methodology by incorporating advanced techniques and optimizing for deployment in real-world scenarios.

2. Scalability and Deployment: Optimizing the model for deployment in resource-constrained environments such as IoT devices and edge computing platforms.

3. Cross-Domain Applications: Adapting the proposed methodology to various domains beyond human anomaly detection.

4. **Interdisciplinary Collaboration:** Advancing the field of anomaly detection will benefit from interdisciplinary collaboration.

5. Ethical Considerations: Addressing ethical considerations such as privacy, fairness, and transparency in anomaly detection systems.

In conclusion, the proposed spatiotemporal autoencoder methodology represents a significant advancement in the field of human anomaly detection within video sequences. By combining deep learning techniques with innovative model architectures, this research has demonstrated promising results in detecting anomalies and offers valuable insights for future research and practical applications. The study's contributions, including the novel methodology, comprehensive evaluation, and practical insights, underscore its significance in advancing anomaly detection technology. Moving forward, continued exploration of enhancements, integration with emerging technologies, and interdisciplinary collaboration will further propel the field towards more efficient, accurate, and scalable anomaly detection solutions.

References

[1] H. T. Duong, V. T. Le, and V. T. Hoang, "Deep Learning-Based Anomaly Detection in Video Surveillance: A Survey," *Sensors (Basel)*, vol. 23, no. 11, p. 5024, May 2023. doi: 10.3390/s23115024.”.

[2] G. Sreenu and M. A. Durai, "Intelligent video surveillance: a review through deep learning techniques for crowd analysis," *Journal of Big Data*, vol. 6, p. 48, 2019. doi: 10.1186/s40537-019-0212-5.

[3] A. Bahnsen, “AC Bahnsen. Building ai applications using deep learning. Building AI Applications Using Deep Learning – albahnsen (wordpress.com)”.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436-444, 2015. doi: 10.1038/nature14539.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 41, 2009. doi: 10.1145/1541880.1541882.

[6] Venkatesh Saligrama and Zhu Chen. Video anomaly detection based on local statistical aggregates. In 2012 IEEE *Conference on computer vision and pattern recognition*, pages 2112–2119. IEEE, 2012.

[7] M. Ahmed, A. Mahmood, and J. Hu, "A Survey of Network Anomaly Detection Techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, 2015. doi: 10.1016/j.jnca.2015.11.016.

[8] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes," *Computer Vision and Image Understanding*, vol. 172, 2018. doi: 10.1016/j.cviu.2018.02.007..

[9] Y. S. Chong and T. Y. H. Tay, "Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder," in *Proceedings of the 5th International Conference on Internet of Things and Cloud Computing*, 2017, pp. 189-196. doi: 10.1007/978-3-319-59081-3_23.

[10] W. Luo, W. Liu, and S. Gao, "A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework," in Proceedings of the *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 341-349. doi: 10.1109/ICCV.2017.45.

[11] M. Hasan, J. Choi, J. Neumann, A. Roy-Chowdhury, and L. Davis, "Learning Temporal Regularity in Video Sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 733-742, 2016. doi: 10.1109/CVPR.2016.86.

[12] C. Aggarwal, “Aggarwal, Charu. (2013). Outlier Analysis. 10.1007/978-1-4614-6396-2.”.

[13] B. Ramachandra, M. Jones, and R. Vatsavai, "A Survey of Single-Scene Video Anomaly Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1-1, 2020. doi: 10.1109/TPAMI.2020.3040591.

[14] F. Jiang, J. Yuan, S. Tsaftaris, and A. Katsaggelos, "Anomalous video event detection using spatiotemporal context," *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 323-333, Mar. 2011. doi: 10.1016/j.cviu.2010.10.008.

[15] S. Calderara, U. Heinemann, A. Prati, R. Cucchiara, and N. Tishby, "Detecting anomalies in people's trajectories using spectral graph analysis," *Computer Vision and Image Understanding*, vol. 115, no. 8, pp. 1099-1111, Aug. 2011. doi: 10.1016/j.cviu.2011.03.003.

[16] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 555-560, Apr. 2008. doi: 10.1109/TPAMI.2007.70825.

[17] E. Zhu, J. Yin, and F. Porikli, "Video Anomaly Detection and Localization by Local Motion based Joint Video Representation and OCELM," *Neurocomputing*, vol. 277, pp. xxx-xxx, 2017. doi: 10.1016/j.neucom.2016.08.156.

[18] L. Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha and Anton van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[19] J. R. M. a. A. Savakis, “Medel, Jefferson & Savakis, Andreas. (2016). Anomaly Detection in Video Using Predictive Convolutional Long Short-Term Memory Networks.”.

[20] W. Liu, W. Luo, D. Lian, and S. Gao, "Future Frame Prediction for Anomaly Detection - A New Baseline," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6536-6545. doi: 10.1109/CVPR.2018.00684.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, W.-F. David, O. Sherjil, C. Aaron, and Y. Bengio, "Generative Adversarial Networks," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International*

Conference on Machine Learning (ICML), 2008, pp. 1096-1103. doi: 10.1145/1390156.1390.

[23] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[24] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive Auto-Encoders: Explicit Invariance During Feature Extraction," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 201/

1.

[25] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2010.

[26] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

[27] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM Neural Networks for Language Modeling," in *Proceedings of Interspeech*, 2012. doi: 10.21437/Interspeech.2012-65.

[28] A. Graves, "Generating Sequences With Recurrent Neural Networks," 2013. [Online]. Available: https://arxiv.org/abs/1308.0850

[29] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2625-2634. doi: 10.1109/CVPR.2015.729.

[30] F. Jiang, J. Yuan, S. Tsaftaris, and A. Katsaggelos, "Anomalous video event detection using spatiotemporal context," *Computer Vision and Image Understanding*, vol. 115, pp. 323-333, 2011. doi: 10.1016/j.cviu.2010.10.008.

[31] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference*, 2019.

[32] M. Yossef, "A Review on Video Anomaly Detection Datasets," *Volume 1, Issue 2, July 2023, Pages 1-9*.

[33] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," 2016. [Online]. Available: https://www.deeplearningbook.org/contents/autoencoders.html

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778. doi: 10.1109/CVPR.2016.90.

[35] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[37] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[38] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[39] C. Zhang, K. Cao, L. Lu, and T. Deng, "A multi-scale feature extraction fusion model for human activity recognition," *Scientific Reports*, vol. 12, no. 1, p. 20620, Nov. 2022. doi: 10.1038/s41598-022-24887-y.

[40] X. Shi, Z. Chen, H. Wang, D. -Y. Yeung, W. K. Wong, and W. -c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS)*, 2015.

[41] C. Shorten and T. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, 2019. doi: 10.1186/s40537-019-0197-0.

[42] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, 2010. doi: 10.1016/j.comnet.2010.05.010.

[43] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," *arXiv preprint arXiv:1702.08608*, 2017.

[44] T. Wang and H. Snoussi, "Histograms of Optical Flow Orientation for Visual Abnormal Events Detection," in *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2012, pp. 13-18. doi: 10.1109/AVSS.2012.39.

[45] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2009, pp. 935-942. doi: 10.1109/CVPR.2009.5206641.

[46] N. R. Nawaratne, D. Alahakoon, D. Silva, and X. Yu, "Spatiotemporal Anomaly Detection Using Deep Learning for Real-Time Video Surveillance," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 1, pp. 1-1, 2019. doi: 10.1109/TII.2019.2938527.

[47] A. M. Hafiz, S. Parah, and R. Bhat, "Attention mechanisms and deep learning for machine vision: A survey of the state of the art," *Research Square*, 2021. doi: 10.21203/rs.3.rs-510910/v1.

Abstract

To complete my Master's degree in Intelligent and Decision-Making Models, a deep learning method was developed for video anomaly detection using a spatial autoencoder combined with Convolutional Long Short-Term Memory (ConvLSTM) networks. In a unified framework, this approach captures both spatial and temporal features for detecting anomalies based on significant reconstruction loss. ConvLSTM networks learn the temporal dependencies in video data, while a clip-based video processing method enhances training efficiency. This combination detects unusual patterns and dependencies in video sequences, making it effective for identifying anomalies across diverse video sources.

Keywords: *Anomaly Detection, Deep Learning, Spatial Autoencoder, ConvLSTM, Temporal Dependencies, Video Surveillance, Clip-Based Processing, Spatiotemporal Features*

Résumé

Pour compléter mon Master en Modèles Intelligents et Décision, une méthode d'apprentissage profond a été développée pour la détection d'anomalies vidéo en utilisant un autoencodeur spatial combiné avec des réseaux de mémoire à long court terme convolutifs (ConvLSTM). Dans un cadre unifié, cette approche capture à la fois les caractéristiques spatiales et temporelles pour détecter les anomalies basées sur une perte de reconstruction significative. Les réseaux ConvLSTM apprennent les dépendances temporelles dans les données vidéo, tandis qu'une méthode de traitement vidéo par clips améliore l'efficacité de l'entraînement. Cette combinaison permet de détecter des schémas et des dépendances inhabituels dans les séquences vidéo, la rendant efficace pour identifier les anomalies à travers diverses sources vidéo.

Mots-clés : *Détection d'anomalies, Apprentissage profond, Autoencodeur spatial, ConvLSTM, Dépendances temporelles, Surveillance vidéo, Traitement par clips, Caractéristiques spatiotemporelles*

خلاصة

لإكمال درجة الماجستير في النماذج الذكية ونماذج صنع القرار، تم تطوير طريقة تعلم عميقة للكشف عن شذوذ الفيديو (ConvLSTM). جنبًا إلى جنب مع شبكات الذاكرة التلافيفية طويلة المدى "autoencodeur spatial SAE" باستخدام في إطار موحد، يلتقط هذا النهج الميزات المكانية والزمانية للكشف عن الحالات الشاذة بناءً على خسارة كبيرة في إعادة التبعيات الزمنية في بيانات الفيديو، بينما تعمل طريقة معالجة الفيديو القائمة على ConvLSTM الإعمار. تتعلم شبكات المقاطع على تحسين كفاءة التدريب. يكتشف هذا المزيج الأنماط والتبعيات غير العادية في تسلسلات الفيديو، مما يجعله فعالاً في تحديد الحالات الشاذة عبر مصادر الفيديو المتنوعة.

الكلمات الدالة:

التبعيات الزمنية، المراقبة بالفيديو، المعالجة ،ConvLSTM ،اكتشاف الشذوذ، التعلم العميق، التشفير التلقائي المكاني القائمة على المقاطع، الميزات الزمانية المكانية