National College of Ireland

Optimized Convolutional-Recurrent Architecture for Detecting Diverse Crimes in Real-Time

MSc Research Project
Data Analytics

Zohaib Rasool
Student ID: 23256796

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Zohaib Rasool

Student ID: 23256796

Programme: Data Analytics **Year:** 2024

Module: MSc Research Project

Supervisor: Vladimir Milosavljevic

Submission Due Date: 24/01/2025

Project Title: Optimized Convolutional-Recurrent Architecture for Detecting Diverse Crimes in Real-Time

Word Count: 11021 **Page Count** 27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Zohaib Rasool

Date: 24/01/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Table of Contents

Optimized Convolutional-Recurrent Architecture for Detecting Diverse Crimes in Real-Time

Zohaib Rasool
23256796

Abstract

This research proposes a new solution to real-time crime detection by expanding the Convolutional Recurrent Auto Encoder (CR-AE) model to detect 12 different types of crimes and typical scenes based on video surveillance data. This work utilizes an incremental model development approach where Conv3D, ConvLSTM2D, and Conv3DTranspose are used to capture spatio-temporal features. Out of the four versions of the model developed the best performing model was Version 2 (V2) when the data was split into 80-10-10 data split. The proposed model testing accuracy was 84.34% with AUC and F1-score being 0.97 and 0.83 respectively, suggesting it would be useful for feature extraction and computational requirements. V2 showed an intermediate depth which allowed it to generalize well across different crime scenes as it outperformed models with higher depth like V4 and V5 that seemed to over-fit. The trade-off relationship found in this work between model complexity and available data is critical if the performance of models is to be maximized. There is an evident efficiency of the suggested system, yet, its weaknesses include the usage of video data only and occasional classification errors. Potential future work can include the addition of audio features, more data augmentation and the use of attention mechanisms to increase resilience and architecture flexibility. This research provides a number of improvements over previous work in automated crime detection and presents a solid basis for the application of intelligent surveillance in more realistic environments.

1 Introduction

1.1 Background

The Global Organized Crime Index 2023 tells that organized crime continues to grow around the world, with 83% of the world's population living in high-crime situations[1]. In order to protect people and public property, Closed-Circuit Television (CCTV) systems have been installed in almost every corner such as transportation zones, markets, residential areas, schools and hospitals. The main goal is to identify any abnormal incident whether it is an accident, theft, robbery, shooting and many other events in the most accurate and timely manner because social public safety is the priority of the surveillance systems. Unfortunately, these systems require humanly resources that is not only tedious but involves a number of shortcomings including wastage of time, increased error in anomaly detection and general inefficiency because one person has to monitor hundreds of screens at the same time and there are human lives at stake. To continuously monitor video surveillance systems at a faster rate with the most efficient computing resources, emerging technologies in light computation and deep learning algorithms including Convolutional Neural Network (CNN) and Long Short-Term Memory

[1] https://globalinitiative.net/analysis/ocindex-2023

Networks (LSTM) provide an opportunity for the identification of a range of crimes with higher accuracy and less biases as stated by (Annem & Pavuluri, 2023). These approaches are intended to solve important issues, including lack of clarity of the crime definitions, the paucity of the labelled crime datasets, and the non-stationary and intricate environment of the real world. Thus, this research aims at using lightweight yet sound computational approaches to investigate efficient solutions towards multiple crime identification.

1.2 Motivation

The motivation for this work comes from the increasing demand for real time crime detection surveillance systems that should detect criminal activities with high accuracy and which are capable to be tuned for various real-world problem contexts. The Office of National Statistics finds a notable crime increase in robbery, violence with injury and fraud and consumer and retail fraud[2]. With more than 1 billion security cameras all over the world[3], Conventional methods of surveillance are generally inadequate in responding to dynamic environments and therefore may fail to timely detect crucial events or may classify the events inaccurately. Moreover, reliance on computationally expensive hardware as a component of such systems may confine the implementation of such systems in regions with restrained infrastructure or processing power. As mentioned by (Bijoor, Alugubelly, & Aggarwal, 2023), These challenges along with the steady increase of various types of crimes underlines a need for research on light weight computing approaches which facilitate real time, accurate and efficient crime detection systems.

1.3 Research Question

How can diverse crime detection be more accurate and reliable with lightweight computing resources?

1.4 Research Objective

This work focuses on designing and implementing an Auto Encoder Convolutional Long Short Term Memory Networks (AE-ConvLSTM) for detecting 12 categories of crimes and differentiating normal situation in video footage. To that end, this study aims to systematically increase the layers in its architecture and test various splits of the dataset to achieve the best accuracy, temporal comprehension, and computational time. The specific objectives are as follows:

1. AE-ConvLSTM with an encoder-decoder structure with Conv3D, ConvLSTM2D, and Conv3DTranspose layers.
2. Incrementally build up on the different versions of the model (from V1 to V5, V5 being the final), each time incorporating layers of each major core type in order to test its performance implications.
3. Vary the ratio of training, validation and test data split in a wide range like 60-20-20, 70-15-15 and 80-10-10 to see the impact of data division.
4. Compare the results for all the versions of the model with respect to accuracy, precision, recall, F1-Score and AUC-ROC score.

[2] https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/bulletins/crimeinenglandandwales/yearendingjune2024
[3] https://www.cnbc.com/2019/12/06/one-billion-surveillance-cameras-will-be-watching-globally-in-2021.html

5. Determine which configuration yields the best results with differentiating the type of crime and best accuracy.

2 Related Work

The use of deep learning in particular and artificial intelligence in general for crime detection has transformed modern surveillance systems. CCTV networks are being installed more frequently as a result of the increased need for automated procedures that can keep an eye on criminal activity. The progress of such technologies is evaluated in this literature review, beginning with the application of transfer learning and pre-trained models made to address limited data problems. It also elaborates on the idea of more specialized deep learning models made to detect specific kinds of crimes, pointing out the specialized development procedure required to lower false positives and boost the model's overall effectiveness.

2.1 Leveraging Transfer Learning for Real-Time Surveillance Applications

Working with transfer learning allows to prevent lengthy training processes and achieve high accuracy rates due to the use of the models previously trained by other individuals. In their paper, (Ilyas, Obaid, & Bawany, 2023) discuss how to use transfer learning to improve aggressive activities in video surveillance. When working on videos, it's crucial to keep in mind that pre-existing models like Residual Neural Network (ResNet) and Visual Geometry Group (VGGNet) are used to extract features from the video's frames. These models are then trained for improved fine-tuning for specific purposes, including violence detection. The primary advantage is that these pre-trained networks perform effectively when fine-tuned on a new task with low quantities of data, requiring less specialized material from the domain. In a similar vein, (Mathur, Chintala, & Rajeswari, 2022) used audio signals and video footage to detect illegal activity. By adding scream detection, an additional warning that indicates a threat, this study broadens the application of deep learning. While both pieces aim to improve the effectiveness of real-time monitoring, the former is more focused on visual data presentation, while the later suggests an audio-visual approach to the issue of precise detection, especially in environments that are difficult to identify.

In the paper by (Viswanatha, et al., 2024), the authors who have proposed evaluating pre-trained CNN models, considered models like ResNet50, MobileNet, and EfficientNet for evaluation. All these models were trained on a total of 2,766 video samples; violent and non-violent; using transfer learning approaches. It can be seen that for fine-grained feature extraction and achieving the highest F1-score of 96% ResNet50 was utilized while for achieving the results with the constraint of resources MobileNet outperformed. (Xue, Chen, & Fang, 2020) suggested conceiving the spatial temporal feature maps as Video Energy Vectors (VEV), reducing the dimension to time series vectors. In addition to being highly accurate, this approach was integrated with Support Vector Machines (SVM) for classification which improved computational speed enabling real time anomaly detection on high-speed video feeds.

These studies demonstrate the growing availability of deep learning in video surveillance by demonstrating the transition from visual-only to visual-and-audio detection systems. The deep learning model that has been created for architecture for various criminal acts will be described in the following part.

2.2 Advanced Deep Learning Architectures for Crime Detection

A new CNN architecture for criminal recognition, more especially, theft and assault is presented in the study by (Nojor, et al., 2022). By focussing on particular traits associated with these actions, this study also highlights the necessity of designing the neural network in a unique manner that is suitable for the type of crime. Therefore, the model's flexibility and fine-tuning are crucial for increasing accuracy across a range of use scenarios. Then, by combining both 2D and 3D, (Jan & Khan, 2022) application expands on the current CNN usage pattern and provides extensive coverage of illegal activity. This is because the model can now include both static and dynamic aspects of the video data, which guarantees accurate detection of complex behaviours like assault and vandalism. Because the study's main focus is on the surveillance system's application, concerns like its resilience and flexibility must be the main focus.

Researchers proposed to design deep learning models for solving crime issues where advanced architectures and augmentation features are utilized to handle determined crime situations. Similarly, in (Vosta & Yow, 2022) the authors used ResNet50 for spatial analysis and ConvLSTM for temporal analysis. Training on video sequences, the model detected acts of violence; the binary classification accuracy was 81.71% based on AUC. For pose information, (Kilic & Tuceryan, 2024) used Google Mediapipe on pose detection and DeepFace on emotion detection. These features were input to the Vision Transformer (ViT) with 4 encoder layers and 8 attention heads with which it was able to recognize behavioral patterns commonly associated with shoplifting with a success rate of 95%. Building on this, (Kilic & Tuceryan, 2024) used activity information derived from optical flow to capture temporal activity movement, raising accuracy to 96%.

However, in order to identify suspicious activity, (Singla & Chadha, 2023) used the object detection concepts that have been established using a combination of traditional machine learning techniques and advanced modern processes. This study demonstrates how to improve the detection of abnormalities in crowded environments by combining machine learning techniques with traditional methods like background subtraction. Because the dual technique checks the detected object with the database of behavioural patterns linked to criminals, it also addresses false positives.

The propensity to create better and more specialised detection models is demonstrated by the comparison of specialised models. This section illustrates the shift from general object detection to specific crime detection systems. This paper's following section will outline how these models are used in real-world situations and go into additional detail about the moral ramifications.

2.3 Challenges in Real-Time Systems and Computational Trade-Offs

The authors of the paper by (Sivakumar, V, R, & S, 2021) describe a number of features of real-time crime detection systems in relation to their real-world use. This study demonstrates the ability to manage live video streams and sound alarms about possible threats almost instantly. The paper also highlights the significance of the optimised algorithm, which combines the speed and accuracy of improved computer vision models and includes You Only Look Once (YOLO) and Single-Shot Detector (SSD). It highlights the reality that, frequently, attaining real-time detection performance lowers detection precision, particularly in situations where the environment is changing. Here, a counterargument about the morality of these technologies is provided by the study by (Pisati, Astya, & Chauhan, 2024). Among the noteworthy characteristics mentioned are the potential for selection bias and the necessity of developing standards for the use of AI systems. In their writings, the authors emphasise the value of transparent and unbiased AI models, particularly in fields like public security where

errors can have disastrous results. (S, M, T, & S, 2022) discuss how CNN architectures can be implemented to satisfy the requirements of specific surveillance scenarios. The goal of this research is to improve the model architecture so that there is as minimal delay as possible between the occurrence of a crime such as vandalism or unauthorised entry and its detection.

The problem of increasing the speed of operations while ensuring maximum accuracy is one of the critical issues in contemporary systems for criminal identification. (Jangam, Mohite, Nayak, & Nimkar, 2023) used algorithms like Video SUMMarization (VSUMM) and Scale-Invariant Feature Transform (SIFT) to extract keyframes from concrete video segments, compromising narrow number of frames by 90% and duration time by 67% of original result. The authors experimented with these methods using action detection models including SlowFast and UniformerV2, to achieve 84.53% of average accuracy. (Basthikodi, Vidya, Pinto, Basith, & Rao, 2024) used for violence detection while YOLO was used for crowd density estimation for dual use. The alerts were implemented in real-time through Telegram notifications and the system produced extremely high accuracy of 96%. These methods underscore the fact that reducing the computation time comes with a binding inverse relationship with the level of detection accuracy that often is rectified in AE-ConvLSTM model by employing spatial-temporal analysis alongside efficient forms of encoder-decoder architectures.

Regarding the efficacy of such systems, it should be noted that while they demonstrate the potential for real-time crime detection, there are some moral considerations that merit consideration. The ethical implications of using technology like surveillance and the possible bias of detection methods on privacy will be examined in the section that follows.

2.4 Dataset Limitations and the Integration of Feature-Rich Models

The work by (Purushotham, et al., 2024) gives important conclusions about the ethics of AI-driven surveillance in the context of the topic mentioned above. It emphasises how biased data can lead to unfair results, which in turn fuel prejudice in society. In order to prevent situations that could result in misuse of the available technology, the study lastly questions transparency in the design of these systems and in their implementation. Making sure that these technologies are not unduly designed to target particular demographics or violate privacy is an ethical concern. Furthermore, (Sheela, Balaji, Balaji, & Kumar, 2023) provide a thorough examination of the many data sets that have been employed by crime detection researchers. Unbalanced datasets that may accurately depict a range of scenarios and populations are one of the major issues that persist in machine learning, according to the report. For instance, the model may learn to handle a particular type of crime or set of conditions but perform badly in areas that are not as commonly represented in training. This can have the same kind of effect and result in bias in the training results. It suggests that larger arrays of data sets, encompassing various contexts and types of criminal incidents, must be included.

Handling datasets across space and time is always a difficult question in crime detection studies. (Rendón-Segador, Álvarez-García, Salazar-González, & Tommasi, 2023) eliminated this through adversarial training and Neural Structured Learning (NSL). The authors added graph-based relationships between adversarial neighbours to enhance generalization across datasets, resulting in Receiver-operating characteristic curve (ROC) Area Under the Curve (AUC) gains of up to 16.50%. Authors (Jyothi, et al., 2023) made the use of face and action detection for crime prevention by keeping in mind the ethical considerations which was maintained by using Haar-cascade detectors for face detection and MobileNetV2 for action classification. However, the authors also pointed to the inherent problems with bias in datasets and dependence on high-quality visuals for optimal performance. Enhanced real-time anomaly detection and feature analysis of surveillance video based on time series for privacy was done

by authors (Xue, Chen, & Fang, 2020) who used customized subsets of the UCF-Crime dataset, using Combined-Difference-Image (CDI) techniques to improve the extraction of motion features, thus improving the anomaly detection when there is limited resources.

In order to overcome these limitations, it is necessary to comprehend and address ethical concerns as well as ensure that the databases utilised include crimes of all kinds. The final section will outline the field of research and offer suggestions for how it might advance to better serve society.

2.5 Advanced Hybrid Architectures for Anomaly Detection

The framework of CR-AE was proposed by (Yang & Wang, 2022) where ConvLSTM has been integrated in a single end-to-end model for detecting anomalies in videos. This hybrid model performs well in spatial randomness and temporal correlation that allow for effective detection of abnormities in video sequences. This CR-AE uses a convolutional decoder for reconstructing video frames and computing the reconstruction loss for determining the anomalous points of view due to the fact that it is much more complex to reconstruct anomalous videos than normal ones. This approach helps avoid the necessity for carrying additional feature extraction, thus making the detection pipeline narrower and potentially more efficient.

Even though CR-AE achieved near to the state-of-the-art results in UCSD Ped2 and ShanghaiTech datasets where frame-level AUCs of 95.60% and 73.10% respectively has been obtained, it does not identify the type of anomalies or crimes. However, in doing so it fails to differentiate a crime such as theft, assault or vandalism as a sign of disorder rather than categorizing all three as deviations that. As a result, this limitation reduces its suitability for use where crime categories need to be categorised in detail. Its end-to-end design nonetheless provides high generalization capabilities with respect to different datasets and various situations. Based on the fact that the model aims at detecting anomalies instead of classifying instances or individuals, it posits itself as a generalized solution to real-world video surveillance, and this is accompanied by an opportunity for further enhancement to solve challenges related to crime type categorization.

2.6 Identification of Research Niche and Future Directions

The detection of crimes through deep learning has seen an improvement in all the following aspects: transfer learning, the architectures, real time systems, and datasets. However, some of the essential issues and research gaps prevails which shows that there is scope of more research in this area.

There is one significant limitation: In the realistic setting, transfer learning is used in real-time detection systems. Although models like ResNet50 and MobileNet are fast in utilizing pre-trained weights to reduce training time, a number of areas of application of these models are limited by resources and variability of the source datasets. Previous literature mainly tries to identify anomaly but does not give much attention to investigate context-specific behavior such as type of crime categorization. Thus, the following questions is proposed for future studies that how should more flexible architectures be designed in order to incorporate crime detection problem with multiple classes without compromising the running time?

Regarding specific models in deep learning at the current stage, ViT and hybrid models of CNN-LSTM are highly effective in identifying certain types of crime, for example, theft in the store or vandalism. Nevertheless, these architectures are frequently fine-tuned for specific kinds of crimes and therefore can hardly be applied to other types of crimes. Moreover, pose, emotion, and activity analysis improve the detection capabilities but are not often incorporated into a single framework to encompass a broad range of crimes classifications. It is inevitable

that future work should focus on merging such contextual features with spatial-temporal modeling for developing manageable and transferable work.

Even though heavy development has been done in the real-time systems, computation is still a trade-off. Milestone-based methods such as keyframe extraction and fast object detection (e.g., YOLO, SlowFast) enhance processing rates but degrade recognition performance, especially in crowded scenes with multiple activities. It is high time for architectures that can be implemented in real-time application while providing a high level of multiple class crime detection. Furthermore, these systems need to be scalable to support large scale integration into existing public safety infrastructures, a problem due to the resource constraints in many of these networks.

Data set limitation also persists as a serious issue in the examination of crime detection. The consequences of having an imbalance dataset are that it may bring bias when training which has poor generality in a variety of situations or different type of crime. Despite or perhaps due to these high-level approaches, new methods such as adversarial training and neural structured learning provide a much-needed improvement in generalisation but often with lower levels of fine-grained control needed for inherently vibrant and multi-class crime datasets. These shortcomings will of course persist and will require diverse and feature rich datasets to mitigate and develop models that work efficiently in real world scenarios.

Finally, despite the innovative structure of the more complex hybrid models, such as the CR-AE, the models are developed for spatial-temporal predictions and none is effective in crime type classification. This limitation makes it important for the hybrid architectures to not only recognize the existence of a crime event but also categorize it. Solutions to such models should include efficient spatial-temporal mechanisms, effective features depending on their contexts and the ability to classify through a vehicular ad hoc network.

These gaps are handled by AE-ConvLSTM model due to the combination of transfer learning, the enhanced architectures and the spatial temporal modeling. It solves the problems of the existing approaches by using Conv3D, ConvLSTM, and Conv3DTranspose in an incremental setup that can be expanded to accommodate new layers and future developments. Thus, with simultaneous detection and classification of 12 crime types, the need in detailed crime typification is satisfied simultaneously with meeting the requirement of generalization across datasets with systematic performance validation with respect to varying data splits. Such a basis has been established by the current work at a level characterized by scalability, efficiency, and flexibility, making further development possible in the future.

3 Research Methodology

This section of the paper includes a detailed elaboration of the research method used to design and test the deep learning model for crime detection. This is the most important part and it incorporates a number of steps and approaches and the hybrid artificial neural networks features with special focus on the CR-AE. This paper depicts the stages of the methodology to include data selection and pre-processing, model design, training and validation and justifiable use of evaluation metrics. *Figure 1* depicts an overview of the complete methodology followed in the research, highlighting the major steps taken which are described in the following sections.

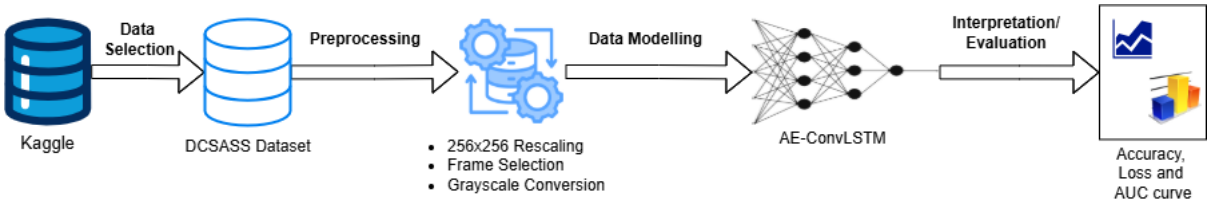

Figure 1 Overview of Research Methodology

3.1 Data Selection and Preparation

The dataset utilized for this research is the DCSASS[4] dataset, which includes videos categorized into 13 classes: five major crimes and seven minor crimes, together with a normal type. This variety also means that the dataset corresponds to the needs of practical observation, where various types of accidents need to be detected.

The **Normal** category rate best this list with a total of 5,846 frames; this might be because a lot of legitimate acts go unnoticed in surveillance videos. Analyzing the crime categories in *Figure 2*, the most frequent one is **Burglary** and the frequency totals 523 frames, while the second place belongs to **Arrest** with 497 frames and **Robbery** category has 489 frames. Other important categories include **Abuse** which has 372 frames, **Shooting** with 304 frames and **Stealing** with 322 frames and many others. Similarly, rare are event categories that include **Road Accidents** containing 97 frames and **Explosion** containing 158 frames that show that the dataset covers a variety of criminal events.

This distribution reinforces the coverage of our dataset to construct a model capable of identifying and classify various kinds of crimes in video surveillance systems.

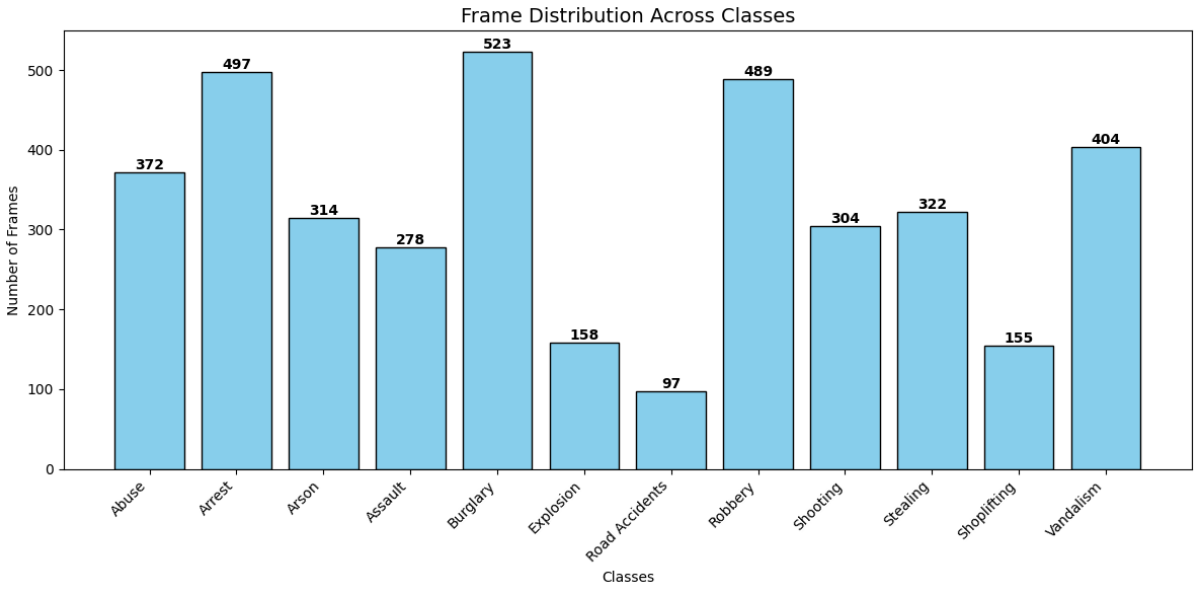

Figure 2 Distribution of classes

This dataset was selected based on its broad spectrum of crime situations; this means the model can be tested on various kinds of crime situations. To ensure the consistency and reliability of the data:

- Crime videos were sorted into directories by crime category, and labels in Comma Separated Value (CSV) files corresponded to the folders.
- Clips were taken from full videos and then converted to frames which were then modified to have the same size of 256x256 resolution. Blending was removed in order to decrease computational burden, nevertheless, significant outlines were preserved by channelling picture into grayscale.
- In the shorter videos, if a few frames were observed to be skimmed, the final frame is to be used, while in the longer videos, the videos were first subsampled to have an equivalent number of frames as the rest of the videos.

[4] https://www.kaggle.com/datasets/mateohervas/dcsass-dataset

A sample of the output of the step is illustrated in *Figure 3* thereby guaranteeing the qualitative nature of the raw data to be used for further processing.

Figure 3 Sample of each frame from each category of crime

3.2 Model Architecture

The hybrid model is based on the CR-AE which has been presented for the first time from (Yang & Wang, 2022). The architecture incorporates:

- **Conv3D Layers**: Analyzing video frames, these extract spatial features from video frames.
- **ConvLSTM2D Layers**: These preserve temporal dependencies and further reshape temporal patterns.
- **Conv3DTranspose Layers**: These upsample the feature maps making reconstruction as well as classification easy.

The incremental enhancement approach, proposed in this research for the first time, involved step by step addition of layers to measure its effectiveness in terms of classification accuracy and time complexity. A typically used methodology was employed in this model where the model was trained to look for anomalies but the range of the work was broadened, meaning it aimed to identify certain crime types instead of that is there a crime or not only in the AE-ConvLSTM framework which is illustrated in *Figure 4*.

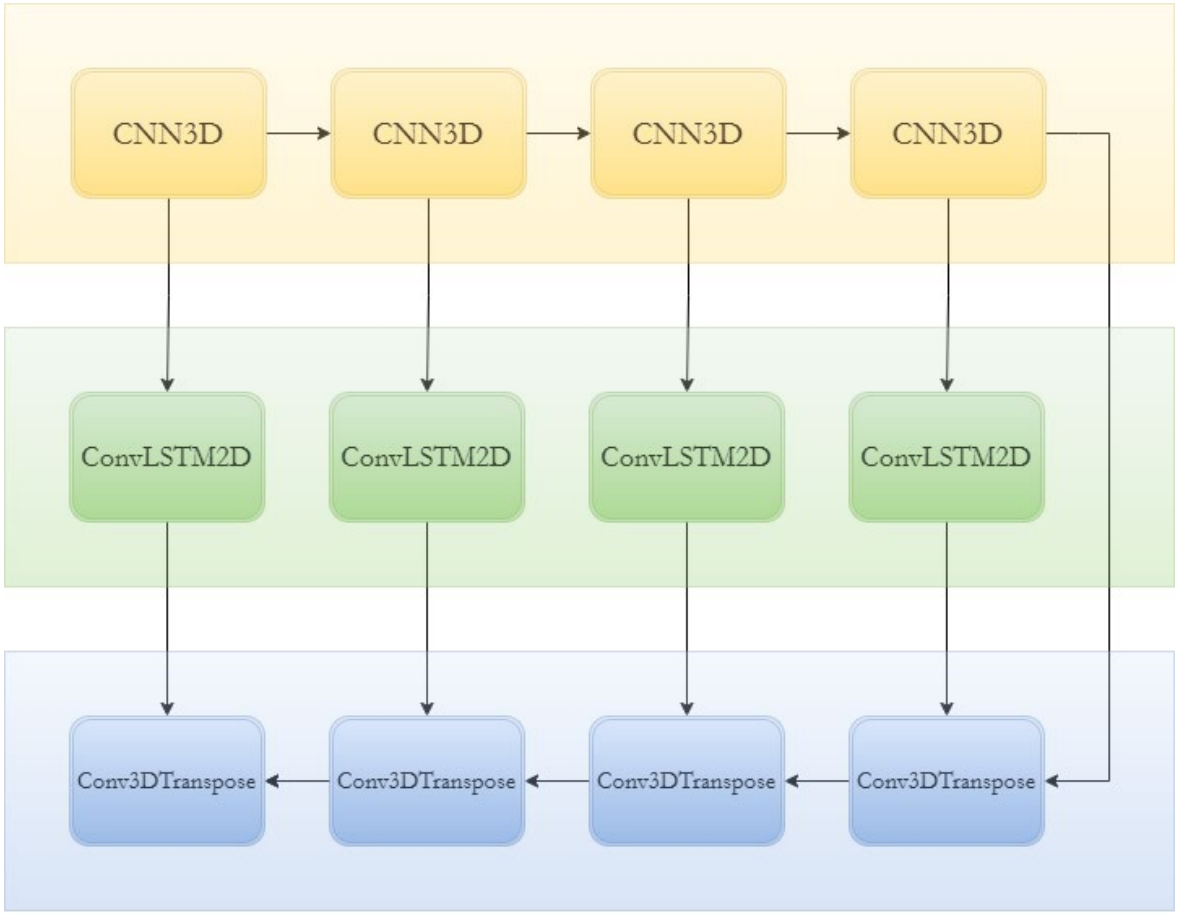

Figure 4 Overall architecture of the proposed AE-ConvLSTM model

3.3 Training and Validation

The dataset was split into training, validation, and test sets using stratified sampling to preserve class distribution:

- Configuration 1: 60% Training, 20% Testing, 20% Validation
- Configuration 2: 70% Training, 15% Testing, 15% Validation
- Configuration 3: 80% Training, 10% Testing, 10% Validation

The target variables were its labels and since multi-class classification was to be performed on it, the labels were one-hot encoded. The optimizer used here is the Adam optimizer that updates its learning rate to 0.001 due to the possibility of adaptive learning. In training, 10 epochs were used (only V5 which is the same architecture as of V4 was trained for 100 epochs), with batch size of 8 and the use of early stopping to minimize over-learning.

3.4 Evaluation and Statistical Analysis

To evaluate the model's performance:

- Metrics such as accuracy, precision, recall, and F1-score were computed for each class.
- AUC-ROC curves were plotted to assess the model’s discrimination ability across classes.

Additionally, comparisons were drawn with the state-of-the-art methods discussed in the related work section, highlighting improvements in accuracy and computational efficiency achieved through the incremental design approach.

4 Design Specification

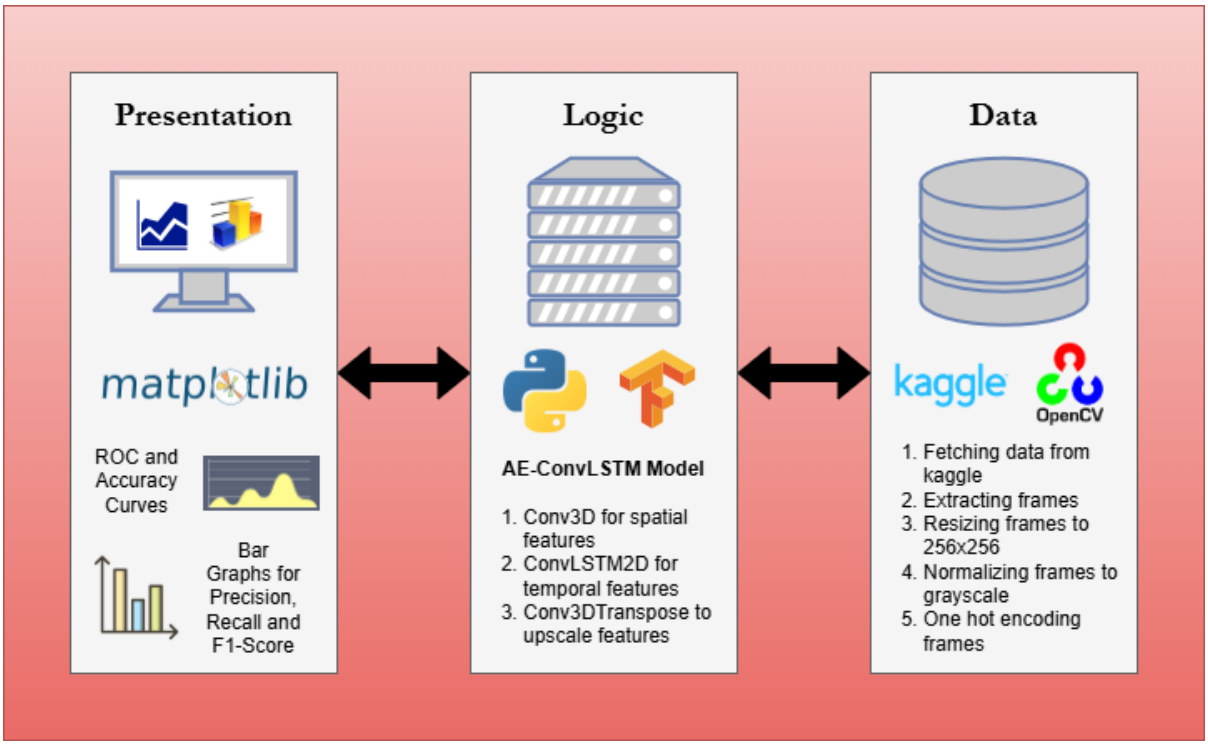

Figure 5 Design Process Flow for Crime Detection

This proposed system in *Figure 5* aims at a three tier architecture model which includes the **Data Tier**, **Logic Tier** and **Presentation Tier**. This architecture provides a good distributed, flexibly scalable, and efficient way to implement and manage the crime detection system. Every level has specific tasks; by doing so, it provides solutions to data preparation, model computations, and results delivery.

The first tier is the **data tier** with specific roles include sourcing, collecting, preprocessing and arranging of raw video data to be analyzed. **Data preprocessing** is the process of cleaning the data and essential steps are crucial for normalizing the inputs. First, the frames of videos are resized to 256 x 256 spatial dimension to reduce computational burden and improve consistency of the spatial dimension. The pixel intensity values are scaled to be between 0 and 1 it makes the training process more stable and faster.

This feature is achieved through some temporal information preservation by sampling same number if frames from each of the video sequences while at the same time avoiding over-complexity and poor manufacturing of features. The labels of each video sequence are then one-hot encoded for the data to be in a ready format for multi-class classification. This preprocessing helps make the data clean, more structured and most importantly preparing the layers for the analysis done in the next layers.

The **logic tier** consists of the essential aspects of the system, and contains the execution of the deep learning model by using Tensorflow. The architecture is tailored to process space-time features of video data. The input layer takes sequences of preprocess videos in a format of `(frames, height, width, channels)`.

Spatial feature extraction is done through a set of **Conv3D**, which perform patterns like movement and object shapes from individual frames. These layers gradually decrease in spatial sampling while enhancing its feature map to be meaningful. The dependency between time frames in videos is also captured by **ConvLSTM2D**, which captures sequences typical of specific crimes. The temporal and spatial analysis combine to allow the system to identify dynamic patterns in space over time.

Layers are used for reconstruction and upscaling of the feature maps of the extracted spatial-temporal features. This reconstruction helps altering 'interpretability' and in fact, augments the model capacity to bring accurate predictions. Last but not least, the reconstructed features are flattened and fed to dense layers; the softmax activation function transforms the features to the 13 output classes which give probabilistic values to each type of crime.

The model is trained using different configurations for the training, validation, and the testing of the model. The training process involves the categorical cross-entropy as the loss function and the Adam optimizer for faster update of parameters. The Logic Tier is the core layer of the system responsible for complex spatial-temporal computing and providing crime type prognosis.

The Presentation Tier is on the preparation and presentation of the outcome of the analysis in a form that is understandable by the user. The outputs of the system are aligned and interpreted on this tier to make sense and be useful to its users.

In the **presentation tier** the results are illustrated using matplotlib by using different illustrative methods such as **ROC** for all crime categories and **accuracy** for all classes to show classification performance of the model. Other measurement tools such as **precision, recall and F1 Score** give a more detailed understanding of the when the model is right and when it is wrong. Next to each input video sequence, the predicted class as well as the confidence scores for each class are then displayed, providing an informative view on the models output.

Indeed, the presentation tier can be further developed in real-world applications to include real-time alerting systems in response to criminal activities based on the generated model. This tier makes not only the result of the system reflection the degree of reliability and usefulness in criminology field, but also its preparedness for being an effective tool in crime detection and prevention.

Python is the programming language used throughout and even throughout the real-time crime prevention in surveillance videos, this three-tier architecture promotes a proper and orderly manner to follow. Due to the modularity of the approach it is flexible, possible to state that it has a clear division of labor, and new changes can be integrated easily, so it seems to be very reliable in terms of meeting the difficulties of crime detection.

5 Implementation

This section incorporates a set of significant steps in the process of the system's implementation – data preparation, including data arrangement and pre-processing, and the development, evaluation, and testing of the model. This section captures how the data was pre-processed, how the model was built incrementally as well as how the established evaluation metric was used to evaluate the performance of the model. The Integrated Development Environment (IDE) used during the development of the complete system was Kaggle's provided code editor because of the fact that Kaggle provides the most hardware resources with the best computation and usage capacity and not only that but the libraries used during the development are also discussed.

5.1 Data Arrangement and Reading

For this implementation, the dataset which consisted of videos grouped into crime category with each category included label information in CSV format files. The implementation started with the use of these CSV files to gain filenames and labels; which form a mapping between videos and categories. Employing some of Python libraries such as Pandas and OS, each video clip was recognized, and its path was built dynamically starting with the category folder and the specific filename of the video.

For better processing the implementation involved additional features of error checking to effectively deal with mistakenly missing or wrongly placed files from the CSV structures and the actual folder structures. The systematic arrangement ensured that all data could be read and processed by the model without the model realising the difference.

5.2 Frame Extraction and Labelling

After the video files were identified from folders as shown in *Figure 6*, frames were extracted from each of the videos using OpenCV. Each video was then split into a one number of frames so that across the entire sample timestamps were equally represented. The number of frames was set to be optimal on what can provide good temporal characterization without being much computational expensive.

Every frame obtained was given a label equalling the category of its associated video according to the CSV mapping. This approach allowed every frame to be labelled and processed by the model collectively with its associated categories for individual frames. If the number of frames in the video was not enough to reach the fixed frame number, the last frame was used several times for the needed number of frames and in longer videos, frames were sampled at equal time intervals.

```
/kaggle/input/thesis/DCSASS Dataset/RoadAccidents
/kaggle/input/thesis/DCSASS Dataset/Assault
/kaggle/input/thesis/DCSASS Dataset/Vandalism
/kaggle/input/thesis/DCSASS Dataset/Arrest
/kaggle/input/thesis/DCSASS Dataset/Shooting
/kaggle/input/thesis/DCSASS Dataset/Arson
/kaggle/input/thesis/DCSASS Dataset/Explosion
/kaggle/input/thesis/DCSASS Dataset/Shoplifting
/kaggle/input/thesis/DCSASS Dataset/Robbery
/kaggle/input/thesis/DCSASS Dataset/Stealing
/kaggle/input/thesis/DCSASS Dataset/Burglary
/kaggle/input/thesis/DCSASS Dataset/Abuse
```

Figure 6 Folder structure of DCSASS dataset

5.3 Grayscale Conversion

In order to lower the dimensionality of the feature space and also avert computational load all the frames were changed to black and white. This step also helped in preserving some of the spatial features needed for crime detection and at the same time reducing on the amount of data which resulted in low computation cost. The frames were also resized to 256x256 pixels to standardize input dimensions, ensuring compatibility with the Conv3D layers used in the model.

The grayscale conversion was done using OpenCV and the **cv2.cvtColor()** function was used for the same. This made the data less large while preserving features such as object boundaries and motion profiles which are vital in crime tracking.

5.4 Incremental Model Development

As for the design and optimization of the architecture for detecting different types of crime, an incremental approach was chosen as a more successful strategy. Through constructing four versions of the model, the precise methodology of the work allowed not only to keep the acceptable computational parameters and accommodate the detailed spatial-temporal properties of the given dataset. This section elaborates why and how each version was constructed and what can be learnt from an iterative approach.

5.4.1 Version 1 (V1): Foundational Framework

The first version with the breakdown of architecture shown in *Table 1* provided a basic structure which consists of only one layer to perform space-time feature extraction and synthesis. V1 included Conv3D Layer which extracted minimum spatial characteristics of the input video frames like movements and stationary objects. There is also the ConvLSTM2D Layer that captured temporal relationships of frames in order to identify temporal patterns in dynamic action. Finally, the Conv3DTranspose layer reconstructed the spatial-temporal features into a higher-dimensional space, making it easier to interpret and classify.

This version gave a starting point of how the basic layers behave and how they engage themselves on the dataset. Despite being less complex, V1 had low computational demands necessary for detecting intricate actions or detailed behaviour, especially where variations prevailed in videos.

Table 1 Architecture breakdown of Version 1

Layer	Input	Kernel Size	Stride	Output	Connected To
Input	1 x 256 x 256	-	-	1 x 256 x 256	Conv3D
Conv3D	1 x 256 x 256	3 x 3 x 3	1x2x2	128 x 128 x 64	ConvLSTM2D
ConvLSTM2D	128 x 128 x 64	3 x 3	-	128 x 128 x 64	Conv3DTranspose
Conv3DTranspose	128 x 128 x 64	3 x 3 x 3	1x2x2	256 x 256 x 64	Flatten

5.4.2 Version 2 (V2): Improved Modelling for Spatial-Temporal Data

In V2, as shown in *Table 2*, a second layer was added to each of the core components and this enhancement helped the model to capture better the level of spatial details and more resilient temporal relationships. Thus more abstract spatial pyramids like overlapping parts or fine movement could be detected by deepening the network with the Conv3D layers. The ConvLSTM2D layers became capable of recognizing longer sequences, enhancing their function for extended or recurrent movements. Conv3DTranspose layers improved the reconstruction process and made it possible to capture fine grained detail of spatial-temporal interactions.

The performance increase was evident in the model and especially for crime types involving motions such as assaults. While it expanded the depth it also brought with it the potential for overfitting which needs more careful tuning and regularization and validation.

Table 2 Architecture breakdown of Version 2

Layer	Input	Kernel Size	Stride	Output	Connected To
Input	1 x 256 x 256	-	-	1 x 256 x 256	Conv3D
Conv3D	1 x 256 x 256	3 x 3 x 3	1x2x2	128 x 128 x 64	Conv3D_1 + ConvLSTM2D
Conv3D_1	128 x 128 x 64	3 x 3 x 3	1x2x2	64 x 64 x32	ConvLSTM2D_1 + Conv3DTranspose
ConvLSTM2D	128 x 128 x 64	3 x 3	-	128 x 128 x 64	Conv3DTranspose_1
ConvLSTM2D_1	64 x 64 x32	3 x 3	-	64 x 64 x32	Conv3DTranspose
Conv3DTranspose	64 x 64 x32	3 x 3 x 3	1x2x2	128 x 128 x 64	Conv3DTranspose_1
Conv3DTranspose_1	128 x 128 x 64	3 x 3 x 3	1x2x2	256 x 256 x 64	Flatten

5.4.3 Version 3 (V3): Complex Feature Hierarchies

V3 added The third level for every primary component predominately enhancing the depth of the model from V2 as displayed in *Table 3*. V3 seek to overcome some of the weaknesses spotted on this version to allow the network to capture complex spatial-temporal relationships existing in the dataset. Conv3D Layers with three layers, the network is capable to learn the spatial hierarchies and recognize the details differences of the shape or movement of the objects in a video. New to ConvLSTM2D layer, these layers were now able to understand multi-step temporal patterns which increase the rate of correct action detection span over time. Concerning the Conv3DTranspose Layers, the added depth was useful in more precise reconstruction of high level features hence improving on the models spatial-temporal mapping features to classes.

V3 showed enhanced accuracy and performance, although the training process was proportional to the computational overhead. Thus, the complexity grew higher, and this required a far more rigorous analysis to check the applicability on other data.

Table 3 Architecture breakdown of Version 3

Layer	Input	Kernel Size	Stride	Output	Connected To
Input	1 x 256 x 256	-	-	1 x 256 x 256	Conv3D
Conv3D	1 x 256 x 256	3 x 3 x 3	1x2x2	128 x 128 x 64	Conv3D_1 + ConvLSTM2D
Conv3D_1	128 x 128 x 64	3 x 3 x 3	1x2x2	64 x 64 x32	Conv3D_2 + ConvLSTM2D_1
Conv3D_2	64 x 64 x32	3 x 3 x 3	1x2x2	32 x 32 x 16	ConvLSTM2D_2 + Conv3DTranspose
ConvLSTM2D	128 x 128 x 64	3 x 3	-	128 x 128 x 64	Conv3DTranspose_2
ConvLSTM2D_1	64 x 64 x32	3 x 3	-	64 x 64 x32	Conv3DTranspose_1
ConvLSTM2D_2	32 x 32 x 16	3 x 3	-	32 x 32 x 16	Conv3DTranspose
Conv3DTranspose	32 x 32 x 16	3 x 3 x 3	1x2x2	64 x 64 x32	Conv3DTranspose_1
Conv3DTranspose_1	64 x 64 x32	3 x 3 x 3	1x2x2	128 x 128 x 64	Conv3DTranspose_2
Conv3DTranspose_2	128 x 128 x 64	3 x 3 x 3	1x2x2	256 x 256 x 64	Flatten

5.4.4 Version 4 (V4): Spatial-Temporal Pattern: Exploring the Last Inch

In the last version, the V4, clearly proved to be the most developed version of the model, as the four layers were provided for each of the core components illustrated in *Table 4*. This depth was intended to engage the spatial-temporal characteristics in the dataset in a manner that, the model can solve even the most complicated patterns of crime. As for the special features, the four Conv3D actually constructed dense and multi-scale representations of the spatial regimes, ranging from low-level components, such as edges and contours, to high-level organization units of objects and scenes. The temporal dynamics are now being handled by the four ConvLSTM2D layers that offered enriched depth in following sequenced patterns to identify minor temporal patterns in dynamic actions. After the reconstruction where the Conv3DTranspose layers help in enhancing and enlarging features, it is guaranteed that necessary cooperation between extracted patterns and their related crime categories was established.

As it can be noted, a model for V4 had better performance in most evaluation criteria but much costlier to compute. The concept was intentionally more complex than necessary for simplicity's sake but provided higher resolution, commonly needed for live monitoring.

Table 4 Architecture breakdown of Version 4

Layer	Input	Kernel Size	Stride	Output	Connected To
Input	1 x 256 x 256	-	-	1 x 256 x 256	Conv3D
Conv3D	1 x 256 x 256	3 x 3 x 3	1x2x2	128 x 128 x 64	Conv3D_1 + ConvLSTM2D
Conv3D_1	128 x 128 x 64	3 x 3 x 3	1x2x2	64 x 64 x32	Conv3D_2 + ConvLSTM2D_1
Conv3D_2	64 x 64 x32	3 x 3 x 3	1x2x2	32 x 32 x 32	Conv3D_3 + ConvLSTM2D_2
Conv3D_3	32 x 32 x 32	3 x 3 x 3	1x1x1	32 x 32 x 16	ConvLSTM2D_3 + Conv3DTranspose
ConvLSTM2D	128 x 128 x 64	3 x 3	-	128 x 128 x 64	Conv3DTranspose_3
ConvLSTM2D_1	64 x 64 x32	3 x 3	-	64 x 64 x32	Conv3DTranspose_2
ConvLSTM2D_2	32 x 32 x 32	3 x 3	-	32 x 32 x 32	Conv3DTranspose_1
ConvLSTM2D_3	32 x 32 x 16	3 x 3	-	32 x 32 x 32	Conv3DTranspose
Conv3DTranspose	32 x 32 x 16	3 x 3 x 3	1x1x1	32 x 32 x 32	Conv3DTranspose_1
Conv3DTranspose_1	32 x 32 x 32	3 x 3 x 3	1x2x2	64 x 64 x32	Conv3DTranspose_2
Conv3DTranspose_2	64 x 64 x32	3 x 3 x 3	1x2x2	128 x 128 x 64	Conv3DTranspose_3
Conv3DTranspose_3	128 x 128 x 64	3 x 3 x 3	1x2x2	256 x 256 x 64	Flatten

5.4.5 Rationale for Incremental Development

This paved way to the incremental approach because by implementing architectural depth systematically, this was able to uncover both the strengths and weaknesses of the model in extracting, modeling and reconstructing features. All the versions were useful as they gave an understanding of the fact which aspects of the model were strong, and which parts may in fact be costly in terms of time and resources. This approach meant that the final architecture, V4, was grounded in data and actually implementable in the real world.

5.5 Data Splitting Configurations

To evaluate the model's performance under different conditions, three configurations of data splits were experimented with: 70-15-15, 60-20-20, 80-10-10. Different proportions of data were allocated for training/validation/testing for each configuration in order to perform a comprehensive study of the model's capability of learning and generalizing.

5.5.1 60-20-20 Split: Balanced Training and Testing

In the first configuration 60% of the data was used for training, 20% for validation, and 20% for testing. With this split, a good amount of data was reserved for validation and testing and this configuration was particularly useful for:

- How well can the model generalize to new data.
- A way to reduce the risk of overfitting since validating on a relatively large validation set.
- To provide a robust test set for final performance evaluation.

This configuration allowed a robust evaluation of a model's generalization, but limited training set size reduced the model's exposure to novel patterns and thus would impede its ability to learn more sophisticated features.

5.5.2 70-15-15 Split: Emphasizing Training

In the second case, 70% of the data was for training with 15% percent each for validation and testing. This was mostly a split to give the model more data to learn from, so that its spatial temporal patterns could be captured. The benefits of this configuration included:

- Greater exposure to diverse samples leading to increased training accuracy.
- For rare crime types there were enhanced feature extraction capabilities.

Although this configuration was able to cope with the reduced validation and test sets, generalization was not properly evaluated with this configuration; this would be more suitable for scenarios that require maximal training accuracy.

5.5.3 80-10-10 Split: Maximizing Learning Capacity

Finally, the data was divided into 80% for training and 20% for validation and testing each taking 10%. Such a split favoured the model's learning capability as it spent a large bulk of the dataset in form of feature extraction and temporal modelling. This configuration was ideal for:

- Preventing overfitting of the deeper models (i.e. V4) as it needed huge data for training.
- To address the data requirement for the rare crime categories, which the model hasn't seen during training.

This configuration improved learning by the model, however the small validation and test sets rendered it impossible to fully evaluate the model's performance. The extra validation techniques, such as cross validation, that were needed to validate the robustness of this set of parameters were highlighted by this trade-off.

5.5.4 Rationale for Multiple Configurations

To learn how the training set size affects the model's performance, the implementation tested training set size by experimenting with different data splits. Results showed how model's ability to generalize was impacted by balance between training and validation/testing data and provided guidelines for allocating data in similar tasks.

6 Results and Evaluation

This section of the paper presents the evaluation outcomes of the different model versions for the three data split setting levels (60-20-20, 70-15-15, and 80-10-10). The presented analysis helps recognize the most effective models in each configuration and understand their effectiveness compared to others. The architectures are versioned 1 to 5 where V5 has the same configuration as the V4 but this architecture has been trained for 100 epochs. The metrics used for evaluation are accuracy $\left(\frac{TP+TN}{Total\ Sample}\right)$, precision $\left(\frac{TP}{TP+FP}\right)$, recall $\left(\frac{TP}{TP+FN}\right)$, F1-Score $\left(\frac{2*Precision*Recall}{Precision+Recall}\right)$ and ROC AUC score where ROC curve is plotted with True Positive Rate (TPR) $\left(\frac{TP}{TP+FN}\right)$ on y axis and False Positive Rate (FPR) $\left(\frac{FP}{FP+TN}\right)$ on x axis and AUC score is calculated $\left(\sum_{i=1}^{n-1}(FPRi+1-FPRi)\cdot\frac{(TPRi+1+TPRi)}{2}\right)$ where n is number of points in ROC curve and $FPRi, TPRi$ are coordinates of ROC curve at i-th threshold. TP, TN, FP and FN are True Positive, True Negative, False Positive and False Negative respectively.

6.1 Configuration 1: 60% Training, 20% Testing, 20% Validation

Table 5 Evaluation metrics results in Configuration 1

Version	Training Accuracy	Testing Accuracy	Validation Accuracy	AUC	Precision	Recall	F1-Score
V1	0.9026	0.8138	0.7961	0.96	0.8003	0.8074	0.7943
V2	0.8859	0.7896	0.7838	0.95	0.7892	0.7859	0.7704
V3	0.9022	0.8149	0.7982	0.96	0.8052	0.8058	0.7946
V4	0.5987	0.6139	0.5994	0.50	0.3586	0.5989	0.4486
V5	0.9909	0.8101	0.8017	0.92	0.7949	0.7966	0.7844

Version 3 (V3) performed well in the 60-20-20 configuration as the results for all the models can be seen in *Table 5*. V3 has the testing accuracy of 81.49%, the validation accuracy of 79.82%, AUC of 0.96 and the F1-score of 0.79. In V3, the three-layer structure enabled the features of spatial and temporal simultaneously, making V3 exhibit consistent performance with the training, validation, and testing set.

Although V1 achieved competitive performance with slightly lower AUC (0.96) and testing accuracy (81.38%) and the F1-score (0.79), V1 indicated its inefficiency in comparison to V3. On the other hand, when the networks are configured in this way, V4 obtained only 61.39% of accuracy on the learning set, most probably because of the deeper architecture of this net which was over-trained on the small learning sample. V5 has an outstanding training accuracy at 99.09% but it has not generalizable well, this is evident from the F1-score of 0.78.

Since previously V3 showed very good performance relatively to other networks, we can state that due to the split the necessary depth of the network was achieved to extract features without overfitting. That made V3 the most efficient for the 60-20-20 configuration.

6.2 Configuration 2: 70% Training, 15% Testing, 15% Validation

Table 6 Evaluation metrics results in Configuration 2

Version	Training Accuracy	Testing Accuracy	Validation Accuracy	AUC	Precision	Recall	F1-Score
V1	0.9187	0.7737	0.8060	0.94	0.7814	0.7794	0.7671
V2	0.7757	0.6739	0.6646	0.77	0.6528	0.6578	0.6047
V3	0.8937	0.7796	0.8320	0.96	0.8001	0.8012	0.7898
V4	0.8915	0.7909	0.8176	0.97	0.7983	0.8026	0.7975
V5	0.9923	0.7883	0.8238	0.92	0.7991	0.8033	0.7958

Among all the presented configurations as shown in *Table 6* of the proposed 70-15-15 architecture, it is identifiable that Version 4 (V4) allowed for reaching the highest results in testing – 79.09%, validation accuracy of 81.76%, AUC of 0.97 and F1-score of 0.80. Incorporated in a deeper architecture, V4 was able to utilise the extra training data properly and thereafter retrieve a variety sophisticated spatial-temporal features and outstanding performance across all indexes of assessment.

Although V3 achieved a testing accuracy of 77.96% and an F1-score of 0.79, V4 was slightly better and offered superior specific class performance in terms of precision and recall which are 0.79 and 0.80 respectively. Indeed, the testing accuracy of V1 turned out to be 77.37%; however, lacking depth, the network failed to perform better than the models under consideration. V5 responded equally well to training records with a 99.23% training accuracy, but the testing and validation results suggest overfitting.

The reason of V4's good result in this configuration was attributed to the benefits of the expanded training dataset with the richer structure of the network. From its performance, the relationship between the model's complexity level and the division of data set used for learning and testing is brought out clearly.

6.3 Configuration 3: 80% Training, 10% Testing, 10% Validation

Table 7 Evaluation metrics results in Configuration 3

Version	Training Accuracy	Testing Accuracy	Validation Accuracy	AUC	Precision	Recall	F1-Score
V1	0.9098	0.8337	0.8207	0.98	0.8358	0.8340	0.8243
V2	0.8831	0.8434	0.8371	0.97	0.8372	0.8402	0.8309
V3	0.8960	0.8263	0.8207	0.97	0.8258	0.8279	0.8208
V4	0.8851	0.8350	0.8289	0.98	0.8328	0.8350	0.8272
V5	0.9848	0.8015	0.8012	0.92	0.7848	0.7848	0.7751

Analysing the results of container throughput distribution between terminals in the 80-10-10 configuration in *Table 7*, Version 2 (V2) performed the best of all. They obtained a testing accuracy of 84.34%, the validation accuracy of 83.71%, an AUC of 0.97 and an F1-score of 0.83. The relatively low depth of V2 ensured that spatial-temporal patterns essential for predicting could be unveiled effectively without incurring a risk of overfitting common with deeper models.

V1 made a high testing accuracy of 83.37% and an F1 score of 0.82 but due to its less complex structure it was unable to identify some broad panorama of crime. Likewise, to the first version, the testing accuracy and F1 score of the V3 were 82.63% and 0.82, respectively. As seen from the results, V4 and V5, which it is deeper than V2, underperformed in this split because of overfitting, where the F1-score and the Validation accuracy are slightly lower than V2.

The outcome of this split shows that while relative model complexity is a good thing, equal model complexities are even better on V2. It gave a high accuracy in generalization between

the validation and testing datasets while having high precision and recall and therefore was the most reliable in this configuration.

6.4 Best-Performing Model Analysis

In all the configurations, the highest performance was recorded in Version 2 (V2) using 80-10-10 data division. V2's success can be attributed to several factors:

1. **Moderate Depth:** The balanced architecture of V2 precept and model these salient features were extracted without compromising on generalization.
2. **Sufficient Training Data:** The 80-10-10 split offered a sufficient number of training examples, which allowed V2 to be trained on how to group the data and recognize multiple patterns.
3. **Class-Specific Performance:** Precision, recall, and F1-score remained above the threshold level for all types of crime types, which testified to the stable and accurate predictions of V2.
4. **Efficient Resource Utilization:** Thus, for a given problem, V2 was as accurate, or even more accurate, than deeper models including V4 and V5, with less computational cost.

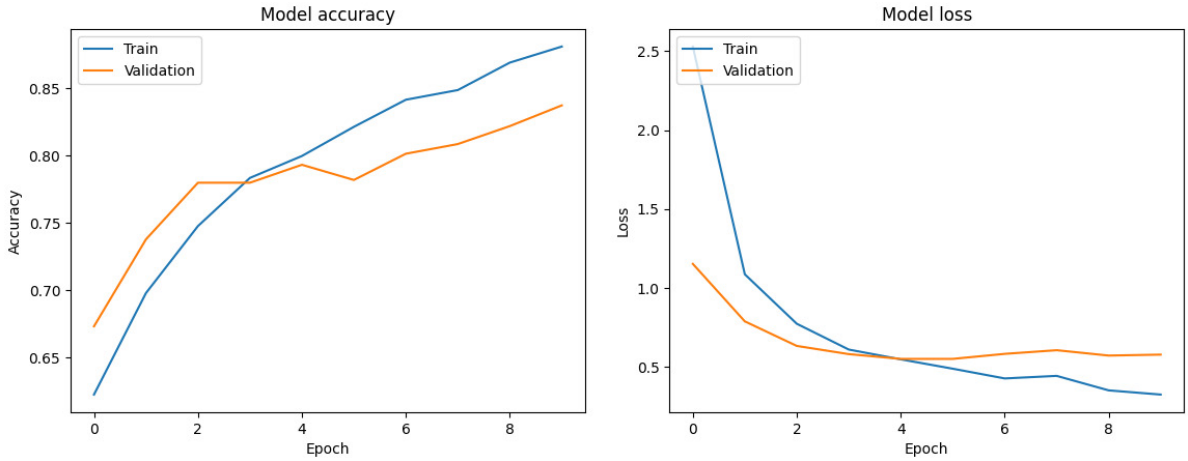

Figure 7 Accuracy and Loss with each epoch

From *Figure 7*, the accuracy graph representing the epoch contains the training and validation values and it shows that the validation accuracy follows the training curve very well without much overfitting. The loss graph also supports the model's stability by illustrating the downwards trend in training and validation loss.

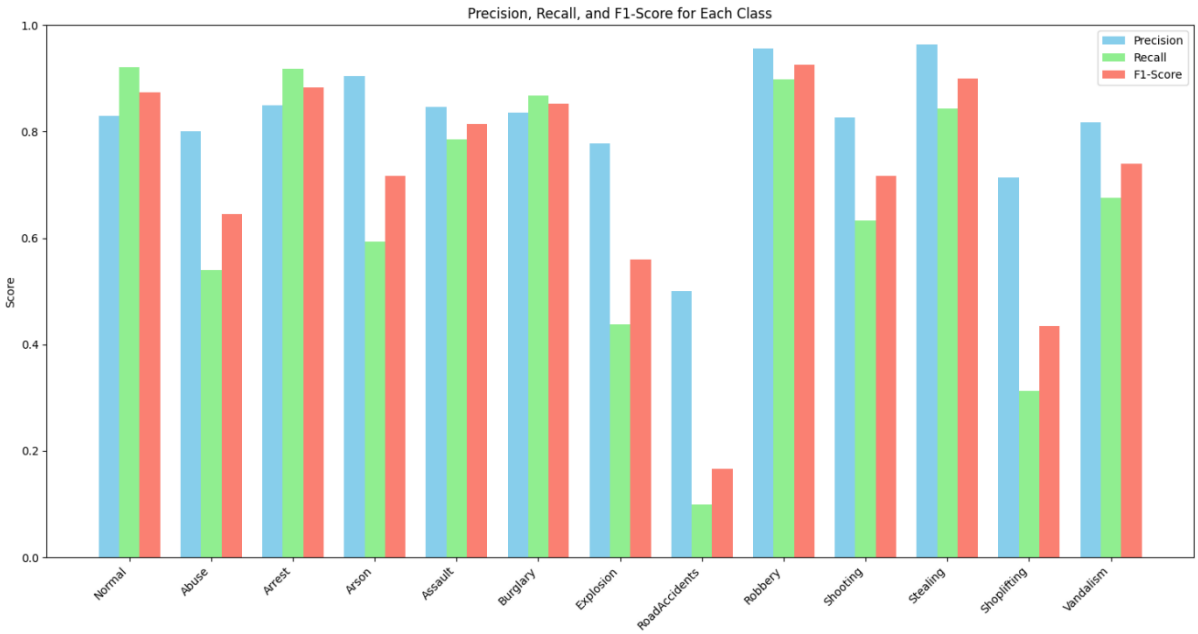

Figure 8 Precision, Recall and F1-Score against each class

The bar chart of precision, recall, and F1-score from *Figure 8* gives an insight into the model's per class classification performance. As with most classes, the precision and recall values are approximately balanced, however, recall for "Shoplifting" and "Road Accidents" is slightly lower indicating possible difficulties in detecting these activities under some circumstances.

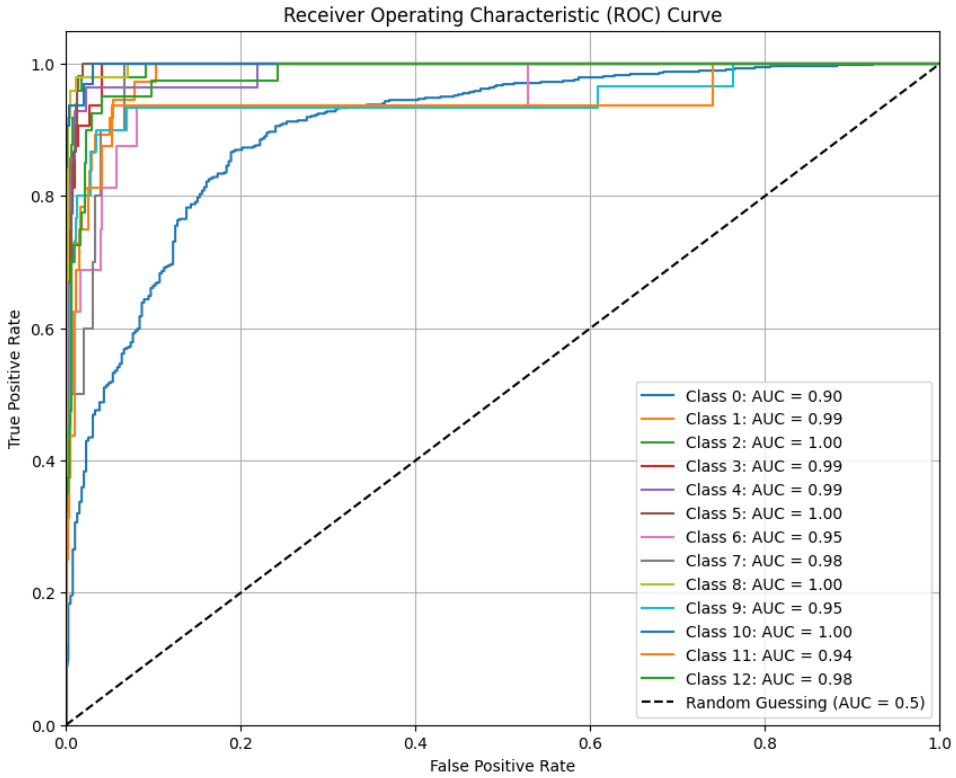

Figure 9 AUC-ROC Curve

Lastly, ROC analysis shows very high classification performance measure by AUC for all classes The AUC values range from 0.90 to imply high discriminant capability. For instance,

Arson, Burglary and Shooting classes, the model achieved AUC scores close to 1 meaning that the model is closely aligned with the its capacity in detecting pattems of these specific classes in the spatio-temporal plane.

Figure 10 True/Predicted crime labels by Version 2

The results from *Figure 10* point out that the model architecture and data splits should correspond to the needs of the given problem. The evaluation of V2 has shown that it is rather resistant and flexible ensuring the most accurate crime identification in real-time video surveillance.

7 Discussion

The findings adapted from the model development in increments and different splits of the data are informative to evaluate the effectiveness and inefficiencies of the suggested crime detection system. The obtained results were discussed in the context of the overall research area and in

particular in relation to the CR-AE model of (Yang & Wang, 2022). Though this study is an extension of their spatial temporal modelling framework, many modifications as well as several special issues were noted while applying these methods for multi class crime detection.

By far, the most enlightening aspect of the proposed model development approach was the incremental model development for analysing the levels of complexity involved in architectures that delivered the best performance possible. Versions 2 (V2) and 3 (V3) again showed better performance, as per the results observed by (Yang & Wang, 2022) in their study, the integration of ConvLSTM layers is useful for anomaly detection as it also captures the temporal patterns. Nonetheless, the more complex ones such as the Versions 4 (V4) and the Versions 5 (V5) seemed to overfit especially with smaller training sets. This support the fact that depth and data availability are vital dimensions to consider with an aim of improving generalization.

However, some of these areas of strength were illuminated as opportunities for further development in the experimental design. Three splits as shown in *Table 8*, further showed that as the training set increases or the test set decreases, the capacity of generalization reduces. The improvement of the performance in the 80-10-10 split can therefore be attributed to large training data sets, although the small sizes of the validation and testing sets raised issues to the stability of the evaluation. The aforementioned limitation could be solved by the use of cross-validation methods where at least part of transformed data is used in cross validation. Furthermore, the chosen evaluation criteria could be expanded to perform class-oriented analysis in order to detect the patterns of misclassification, which is particularly relevant for the less frequent types of crime, such as shoplifting or arson.

Table 8 Breakdown of frames in different data splits

Configuration	Training	Testing	Validation
60 – 20 – 20	5855	1952	1952
70 – 15 – 15	6831	1464	1464
80 – 10 – 10	7807	976	976

Due to the modularity of the architecture of deeper models such as V4 and V5 this is an area of research that could lead to a significant scaling of the models while reducing their complexity and generally maintaining their performance.

It is possible to state that further changes may be made to enhance data diversity by using data augmentation techniques such as frame rotation, scaling and temporal jittering. Even integration of some transfer learning approaches with more trained models such as ResNet or EfficientNet could also minimize the training time and foster the performance especially for the classes that have few data representations. Lastly, the use of contextual inputs such as audio data proposed by (Mathur, Chintala, & Rajeswari, 2022) could also improve the system's insight as well as its reliability.

8 Conclusion and Future Work

Conclusively, research is provided on incremental development for the model based on Convolutional-LSTM autoencoder to enhance spatial-temporal features and classification results. The analysis examined depths of architecture on four iterations of the model for performance improvements under different configurations of data splits. The specific research questions and goals were that what is the most effective model for the simultaneous detection of multiple crimes, based on the available data and different evaluation scenarios and What configuration of the proposed model should be used for real-time implementation.

The results show that this research achieved its goals and answered the posed research question effectively. Specifically, all the tested models, such as V2 and V3, have reported high accuracy, AUC and F1-score for each configuration, especially where a large dataset was used for training. The incremental development approach was equally successful in achieving the goal of finding the right levels of the model that balances complexity with generalization capabilities, with the best candidates of the models achieving the level of performance of reaching up to 12 crime types and normal scenarios levels. These findings therefore offer a great advance in enhancing real time crime identification for surveillance purposes.

However, the present study has several limitations despite the positive findings. Since the presented approach focuses solely on videos and does not introduce other input modalities, the model may fail when there are occlusions or when the lighting conditions are low. Moreover, the sophisticated deeper models such as V4 and V5 showed overfitting during training in the configuration with smaller training sets, and it is likely that better regularization algorithms or multimodal data blending could be useful in the future. Despite the clear and comprehensive evaluation of the individual components, some aspects could have been approached with cross-validation or extending with more real-world datasets testing.

There are several tremendous implications as can be evidenced from this research. Therefore, this work is useful to the development of more improved and efficient real time crime detection methodologies and overall goals of mega city policing enhanced automated surveillance systems where the general public benefit by reduction of security threats. The scalability of the model with respect to the various crime categories also emphasizes how the model could be implemented in smart city surveillance and security, policing, and public safety.

Subsequently, future research will consist of the addition of audio features to the framework, alongside multimedia approaches that use video alongside IoT sensor data or other contextual data. Examining attention mechanisms as part of the architecture's design might yield enhancements to spatial-temporal feature extraction and overall performance in challenging situations. Moreover, it can be quite useful for commercial purposes of using the model this could be optimized to work on edge devices with low resource constraint environments. Besides the improvement of the methods, aspects of ethical nature, like the concern with privacy or approaches for overcoming bias, should become more important in further work on the development of systems for crime detection.

Therefore, this research has established an efficient and relevant strategy in crime detection by applying this innovative technique. Thus, strengthening its premises and widening the range of its perspective, the consequent research and development actions can make use of this study as a starting point for further enlarging the capacities and possibilities of the automated surveillance and contributing to the common benefit of the society.

References

Annem, G., & Pavuluri, G. (2023). A Deep Learning Approach to Video Anomaly Detection using Convolutional Autoencoders. *arXiv*.

Basthikodi, M., Vidya, B., Pinto, E. M., Basith, M., & Rao, S. A. (2024). AI Based Automated Framework for Crime Detection and Crowd Management. *2024 Second International Conference on Advances in Information Technology (ICAIT)* (pp. 1-6). Chikkamagaluru, Karnataka, India: IEEE.

Bijoor, S., Alugubelly, M., & Aggarwal, S. (2023). Light Weight Real-Time Burglary Detection and Inspection in Low Light Surveillance Videos. *2023 16th International Conference on Developments in eSystems Engineering (DeSE).* Istanbul, Turkiye.

Ilyas, A., Obaid, S., & Bawany, N. Z. (2023). Deep Learning for Violence Detection in Surveillance: The Role of Transfer Learning and Pre-Trained Models. *2023 24th International Arab Conference on Information Technology (ACIT)* (pp. 1-8). Ajman, United Arab Emirates: IEEE.

Jan, A., & Khan, G. M. (2022). Deep Vigilante: : A deep learning network for real-world crime detection. *Journal of Intelligent & Fuzzy Systems, Volume 42, Issue 3*, 1949 - 1961.

Jangam, A. G., Mohite, A. P., Nayak, D. U., & Nimkar, A. V. (2023). Keyframe Extraction assisted Crime Detection. *2023 7th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA)* (pp. 1-6). Roorkee, India: IEEE.

Jyothi, A. P., Kumar, M., Pravallika, V., Saha, R., Sanjana, K., Varshitha, V., . . . Narayan, A. (2023). Face and Action Detection for Crime Prevention: Balancing Promise and Ethical Considerations. *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)* (pp. 1-7). Kalaburagi, India: IEEE.

Kilic, S., & Tuceryan, M. (2024). Crime Detection from Pre-crime Video Analysis with Augmented Pose and Emotion Information. *2024 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)* (pp. 97-100). Santa Fe, NM, USA : IEEE.

Mathur, R., Chintala, T., & Rajeswari, D. (2022). Identification of Illicit Activities & Scream Detection using Computer Vision & Deep Learning. *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1243-1250). Madurai, India: IEEE.

Nojor, V. V., Austria, J. A., Galit, A. A., Guevarra, J. T., Jogno, K. Q., Venal, M. C., & Somao-i, M. J. (2022). Design of a Deep Learning-based Detection System for Criminal Activities. *2022 3rd International Informatics and Software Engineering Conference (IISEC)* (pp. 1-5). Ankara, Turkey: IEEE.

Pisati, R., Astya, R., & Chauhan, P. (2024). A Profound Review of AI-Driven Crime Detection in CCTV Videos. *2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT)* (pp. 193-199). Sonepat, India: IEEE.

Purushotham, P., Srividya, G., Chitty, A., Kurakula, A. K., Silparaj, M., & Kiran, A. (2024). Detection of Criminal Activity Using Deep Learning. *2024 International Conference on Science Technology Engineering and Management (ICSTEM)* (pp. 1-6). Coimbatore, India: IEEE.

Rendón-Segador, F. J., Álvarez-García, J. A., Salazar-González, J. L., & Tommasi, T. (2023). CrimeNet: Neural Structured Learning using Vision Transformer for violence detection. *Neural Networks*, 318-329, Volume 161.

S, S., M, H., T, D., & S, S. (2022). Real-time Crime Detection Using Customized CNN. *2022 1st International Conference on Computational Science and Technology (ICCST)* (pp. 416-419). Chennai, India: IEEE.

Sheela, A. J., Balaji, S., Balaji, B., & Kumar, U. H. (2023). A Survey on Crime Detection using CCTV Systems. *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 254-261). Bengaluru, India: IEEE.

Singla, S., & Chadha, R. (2023). Detecting Criminal Activities From CCTV by using Object Detection and machine Learning Algorithms. *2023 3rd International Conference on Intelligent Technologies (CONIT)* (pp. 1-6). Hubli, India: IEEE.

Sivakumar, P., V, J., R, R., & S, K. (2021). Real Time Crime Detection Using Deep Learning Algorithm. *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)* (pp. 1-5). Puducherry, India: IEEE.

Viswanatha, V., Ramachandra, A. C., Parameshachari, B. D., Thota, S. M., Nalluri, S. D., & Thota, A. (2024). Next-Generation Crime Detection and Transmitting: Evaluating Pre-Trained CNN Models. *2024 Second International Conference on Data Science and Information System (ICDSIS)* (pp. 1-6). Hassan, India: IEEE.

Vosta, S., & Yow, K.-C. (2022). A CNN-RNN Combined Structure for Real-World Violence. *Applied Sciences. 2022*, 12(3), 1021.

Xue, R., Chen, J., & Fang, Y. (2020). Real-Time Anomaly Detection and Feature Analysis Based on Time Series for Surveillance Video. *2020 5th International Conference on Universal Village (UV)* (pp. 1-7). Boston, MA, USA: IEEE.

Yang, C., & Wang, B. (2022). Video Anomaly Detection Based on Convolutional Recurrent AutoEncoder. *Sensors*, 22(12), 4647.