JSS Mahavidyapeetha
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
JSS Technical Institutions' Campus, Mysuru – 570006

"Video Surveillance Anomaly Detection on Crime using Deep Learning approaches"

A Report submitted in partial fulfillment of curriculum prescribed for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCEINCE AND BUSINESS SYSTEMS
by

Dhundhubi A	*01JST21CB010*
Kushal Gowda K N	*01JST21CB020*
Srushti Gundappa Khavatkoppa	*01JST21CB041*
Yadunandan N M	*01JST21CB052*

Under the Guidance of

Dr R J Prathibha
Associate Professor

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
June, 2025

Educate Elevate Enlighten

JSS Mahavidyapeetha

JSS Science And Technology University

(Established Under JSS Science and Technology University Act No. 43 of 2013)

JSS SCIENCE AND TECHNOLOGY UNIVERSITY MYSURU

CERTIFICATE

This is to certify that the work entitled "VIDEO SURVEILLANCE ANOMALY DETECTION ON CRIME USING DEEP LEARNING APPROACHES" is a bonafied work carried out by Dhundhubi A (01JST21CB010), Kushal Gowda K N (01JST21CB020), Srushti Gundappa Khavatkoppa (01JST21CB041) and Yadunandan N M (01JST21CB052) in Partial fulfillment for the award of the Degree of BACHELOR OF ENGINEERING IN COMPUTER SCEINCE AND BUSINESS SYSTEMS of JSS Science and Technology University, Mysuru, during the year 2024-25. It is certified that all corrections / suggestions indicated during Continuous Internal Evaluation have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in concerning the project work prescribed for the Bachelor of Engineering degree.

Guide
Dr R J Prathibha,
Associate Professor
Dept. of ISE,
JSS STU, Mysuru.

Head of the Department
Dr. S P Shiva Prakash,
ISE and CSBS,
JSS STU, Mysuru.

Examiners:

	Name	**Signature with Date**
1		
2		
3		

DECLARATION

We hereby declare that the project entitled “Video Surveillance Anomaly Detection on Crime using Deep Learning approaches” has been carried out Dhundhubi A, Kushal Gowda K N, Srushti Gundappa Khavatkoppa and Yadunandan N M under the guidance of Dr. R J Prathibha, Associate Professor, Department of Information Science and Engineering, JSS Science and Technology University, Mysuru, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Business Systems during the academic year 2024–25.

We further declare that this project report is our original work and has not been submitted to any other university or institution for the award of any degree or diploma.

Name	**USN**	**Signature**
Dhundhubi A	01JST21CB010	
Kushal Gowda K N	01JST21CB020	
Srushti Gundappa Khavatkoppa	01JST21CB041	
Yadunandan N M	01JST21CB052	

Date: 03/06/2025

Place: Mysuru

ABSTRACT

Anomaly detection in video surveillance is vital for identifying unusual and potentially harmful events across monitored environments. However, manual review of video feeds is tedious and error prone. This paper presents a deep learning-based anomaly detection system that combines ResNet-18 for spatial feature extraction and Long Short-Term Memory (LSTM) networks for capturing temporal patterns. The system is trained and evaluated on a subset of the UCF-Crime dataset, focusing specifically on four anomaly classes: arson, explosion, fighting, and stealing. The proposed model achieves an overall test accuracy of 62.2%, with strong recall on high-impact classes like arson and stealing. A user-friendly graphical interface and real-time email alert mechanism are integrated to assist security personnel in monitoring and rapid response. In contrast to prior work employing deeper Res-Net variants or Simple Recurrent Units (SRUs), this approach emphasizes a lighter and more deployment-friendly architecture while maintaining competitive performance.

ACKNOWLEDGEMENT

An endeavour is successful only when it is carried out under proper guidance and blessings. I would like to thank a few people who helped me in carrying this work by lending invaluable assistance.

I express my sincere regards to His Holiness Jagadguru Sri Shivaratri Deshikendra Mahaswamiji who has showered his blessings on us for framing our career successfully. Deem We hereby thank Dr. C Nataraju, Principal (I/C) and Dean(Engineering and Technology), SJCE, JSSSTU, Mysuru and Dr. S P Shiva Prakash, HOD, Department of Information Science Engineering, SJCE, JSSSTU, Mysuru who encouraged me in this venture. It is my foremost duty to thank my project supervisor Dr R J Prathibha for her encouragement, effective guidance, and valuable suggestions right from the beginning of this project till its completion. I thank the panel members for their support and guidance throughout the project.

I also extend my gratitude to all the teaching and non-teaching staff of the Department of Information Science and Engineering who have helped me throughout the course of this project. Finally, I would also like to thank my family and friends for their constant support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 OVERVIEW

In high traffic environments, manual surveillance can be tedious and susceptible to errors because it often requires people to observe events from continuous video feeds. In this project, we introduced an automated anomaly detection solution, designed to automate the process of monitor video to help people accurately identify abnormal behavior. The automated anomaly detection solution leverages deep learning, to analyze video data, to either classify a given event as normal or suspicious, allowing human operators to reduce the time spent observing video feeds by only reacting to ideal events after they have been flagged.

The solution incorporates a hybrid model with a ResNet-18 backbone to extract spatial characteristics from video frames, and we use LSTM to learn/develop temporal associations across frame sequences. The model was trained on a subset of the UCF-Crime video dataset with a balance of the different classes of anomalous activities, depicting a range of activities such as arson, explosion, fighting, stealing, and normal activity. It was important for the model to maintain balanced performance across the different classes, but also to improve reliability when considering that people would be using the product in very different surveillance environments. Our architecture maintains solid performance and reliability, and is made to scalable and efficient for future systems integration. The system provides automated feature extraction and classification, allowing human operators prioritize of critical decision making from the identified classification to limit interaction perturbations in normal conditions that require manual surveillance. Overall, in a high traffic environment, the system will mitigate the amount of time human observers spend looking at video enormously and it enables more reliable and informed human interaction and decision making process when responding to video monitoring conditions.

1.2 ABOUT THE WORK

This effort entails an automated video analysis system that is designed to serve as a supplement to existing surveillance actions by acting as an early warning system for unusual activity not identifiable in a human generated video feed. The purpose of the study is to decrease the amount of work performed by human personnel and increase the reliability of surveillance through the use of deep learning to identify events depicted in video. The system categorizes video into types of events where human video surveillance may be difficult, such as arson, explosion, fighting, stealing, and normal activity. The objective is to speed up a video recording in combination with a human monitoring observer's ability to watch and react quicker then using human observation only. The model has two modules: ResNet-18 to extract spatial features from video frames, and LSTM to examine the time dynamics of the captured video events. The model learns what occurs in each frame and how the events transpire across frames. The data captured is protected and accessible to be used for alerts or another occasion's viewing, this supports early issue detection. The user interface is developed for usability, which allows use of the system in user-friendly environments that may include distance or equipment limitations. Essential to the model is the determination that its output through automation, the detect and classify process will support quicker decision-making and ultimately increase the efficiency of video surveillance.

1.3 PROBLEM STATEMENT

Various traditional and deep learning approaches to detecting anomalies in video surveillance systems have been explored, but no one approach can successfully accommodate large datasets, multiple anomaly types, and different types of anomaly variations. Furthermore, even with the existed methods of detection, the system administrator can lack proper software tools to monitor and review flagged anomalies. This project seeks to develop a fast and light automated system that can detect anomalies in video data as well as provide physical tools to monitor and assess the video surveillance system.

1.4 CHALLENGES

1. Handling and processing a mass amount of video data while maintaining system performance is very difficult with limited computational resources.
2. Due to the slight differences between anomalies and deviations of normal events that can occur over more complex, real-world scenarios, accurate anomaly detection is extremely difficult.
3. While attempting to capture both the visual features occurring from individual frames while simultaneously accounting for the changes that occurred over time in relation to time, a well-balanced and structured deep learning model can be difficult to implement from scratch, while also taking into consideration the need to optimize it.
4. The fact there is a lack of available labelled and balanced datasets to train on, limits the generalization ability of any model across different anomalies and environments.
5. Though the final solution would benefit from providing a user-friendly interface for reviewing detected anomalies, as well as assisting users in the understanding of flagged events, this remains challenging to effectively implement within a lightweight solution.

1.5 OBJECTIVES

1. To develop a deep learning model which is computationally lightweight and powerful to accept pre-recorded video data, and recognize important events, without needing to operate necessarily in real-time.

2. To create a true balance of normal behaviour and abnormal behaviour training dataset, to give the system opportunity to discriminate between types of anomalies.

3. Using ResNet-18 for spatial feature extraction for later use temporally to identify behaviour-related anomalies occurring through time within video sequences.

4. To have the system operating efficiently and effectively at the proper speed for processing and assessing extremely large video datasets, while not having an extremely powerful hardware to implement it on, and needing little human involvement.

5. Design a user-friendly alert interface that helps security personnel review, interpret, and respond to flagged anomalies in a timely and organized manner.

Chapter 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

1. The research titled **"Video Anomaly Detection System Using Deep Convolutional and Recurrent Models"** [1] combines CNN and Simple Recurrent Unit (SRU) architectures with various ResNet backbones (18, 34, 50) to detect violence in UCF Crime videos. The study highlights that incorporating attention mechanisms alongside CNN and SRU enhances anomaly detection accuracy, pointing towards the significance of temporal attention for better contextual understanding.

2. **"Conditioned Cooperative Training for Semi-Supervised Weapon Detection"** [2] utilizes pseudo-labelling driven by a teacher model with a novel threshold search, leveraging ResNet50 for weapon detection on YouTube GDD videos. The semi-supervised framework struggled to improve beyond certain cycles, primarily because of imprecise annotations on unknown images, suggesting that label quality is a crucial bottleneck in training semi-supervised detection models.

3. The paper **"RareAnom: A Benchmark Video Dataset for Rare Type Anomalies"** [3] introduces a dataset specialized in rare anomalies such as violence, abuse, kidnapping, and accidents. It applies 3D convolutional autoencoders (CAE), latent feature extraction, and unsupervised classifiers like Isolation Forest and One-Class SVM. The study discusses challenges due to the closed-set nature of the dataset and suggests that deep learning integration in the autoencoder stage could improve detection accuracy. Additionally, supervised methods may improve initial detection against their unsupervised twin.

4. **“Abnormal Behaviour Detection in Uncrowded Videos Using Two-Stream 3D Convolutional Neural Networks”** [4] introduces a deep learning-based framework for detecting abnormal behaviour in the context of videos with low crowd density. The method employs a two-stream3D CNN (3D-CNN), with one stream taking as input spatial data and the other the temporal motion cues. It processes the inputs of RGB and optical flow separately and combines the outputs together to improve the classification of anomalies. The study stresses the power of 3D convolutions to learn spatiotemporal characteristics and stresses that the improved performance is most applicable when the abnormal behaviours are more a subtle indication, and the majority of its application consisted of low density environments. However, it is also important to note some limitations that the method faces which included high computational costs due to the architecture with two streams of data and being sensitive to cases of camera motion and background clutter. Moreover, it can also expect to see decreased performance in complex real-world environments with the definitions of abnormal behaviours being as contextual as they are not visually differentiated.

5. Similarly, **"Real-World Malicious Event Recognition in CCTV Recording Using Quasi-3D Network"** [5] is another deep learning-based method using 2D and 3D convolution filters with ResNet to measure the recognition of events in videos such as shootings or fighting in separate custom crime scene and hockey fight datasets. The study notes that there was very limited anomaly diversity, and proposed that the findings would likely decrease even more when redundancy is eliminated and frame processing rates can be maximized in real time.

6. **"Anomaly Recognition from Surveillance Videos Using 3D Convolution Neural Network"** [6], also uses 3D CNNs and a pre-crime behaviour in the UCF Crime dataset, analyzes detection of suspicious behaviours such as snatching. The study is limited in that it only used the UCF crime dataset and cannot clearly identify more suspicious behaviours such as burglary or arson. This study demonstrates that the study can be further developed and is limited

in using a higher diversity of crime types which can improve the utility of this type of system.

7. The framework **"Fighting for a Future Free from Violence: Real-Time Detection of 'Signal for Help'"** [7] employs CNN models such as MobileNet for recognition, ResNet50 for detection, and ResNeXt101 for classification on a custom dataset designed to detect distress signals. Although it achieved lower accuracy than state-of-the-art methods, it offers significant improvements in computational efficiency with faster inference and lower resource consumption. The authors stress the need for more comprehensive and better-trained datasets to improve real-time performance.

8. The work **"Fighting Against Terrorism: A Real-Time CCTV Autonomous Weapons Detection Based on Improved YOLO v4"** [8] focuses on real-time weapon detection using an improved YOLOv4 with ResNet and multi-scale dilation modules. It is trained on synthetic datasets, but the study notes that the effects and robustness of using synthetic data for such critical detection tasks remain under-explored, indicating a gap in validating model performance on real-world data.

9. In **"Vision Transformer Attention with Multi-Reservoir Echo State Network for Anomaly Recognition"** [9], a novel approach integrates vision transformers with sequential learning via multiple reservoirs in an echo state network (ESN). The method was tested on datasets like UCF Crime and LAD-2000, targeting anomalies such as vandalism and road accidents. Despite improved feature extraction and temporal modeling, challenges such as intra-class similarity, occlusion, poor illumination, and background clutter led to misclassification in some anomaly categories, emphasizing the need for robust feature discrimination under adverse conditions.

10. The paper **"Sequential Attention Mechanism for Weakly Supervised Video Anomaly Detection"** [10] presents a hybrid model combining deep convolutional neural networks (DCNN), temporal convolutional networks (TCN), and an attention mechanism based on BotNet-152 architecture.

Evaluated on multiple datasets including UCF Crime2local and Crowd Violence, it showed effectiveness in detecting a variety of anomalous events. However, mispredictions occurred due to occlusions, fixed camera angles, and the strong similarity between different action classes, highlighting challenges in spatial-temporal context understanding.

11. The **"Intelligent Dual Stream CNN and Echo State Network for Anomaly Detection"** [11] presents a dual-stream architecture combining 2D CNNs with echo state networks and feature extraction via EfficientNetB7 autoencoders. Evaluated on multiple datasets including UCF Crime and Surveillance Fight, the approach faced limitations in accurately modeling object relationships. The use of optimized features instead of all features caused some loss of important spatial and pattern information, resulting in reduced performance in complex scenes.

12. The study **"Violence Detection by Pretrained Modules with Different Deep Learning Approaches"** [12] explores the use of deep learning architectures such as LSTM combined with VGG16/19 and ResNet50 for detecting violent and panic behaviours in movie scenes. While the models showed promise, the research highlighted limitations in missing certain types of violence-triggering behaviour. Additionally, the study reported that the relatively small dataset used resulted in marginal overfitting, indicating the need for larger and more diverse datasets for better generalization.

13. The research **"Extracting Pickpocketing Information Implied in the Built Environment by Treating it as Anomalies"** [13] uses ResNet50 with SHAP (Shapley Additive Explanations) on a customized Point of Interest (POI) dataset to analyze pickpocketing-related crime risks. The approach focuses more on risk and behaviour analysis than direct crime detection. The study recommends detailed and comprehensive datasets involving census, travel, and trajectory data for improved modeling of unusual behaviours and their spatiotemporal correlations at micro levels.

Chapter 3

SYSTEM REQUIREMENTS AND SPECIFICATIONS

3.1 FUNCTIONAL REQUIREMENTS

Functional requirements state what basic features the system has to include for the application or the developed system to work.

1. **Video Frame Extraction and Preparation:**
    1. The system takes videos as input and selects 100 frames from each video, evenly spaced apart.
    2. These frames are converted to black-and-white (grayscale) and resized to a smaller size to reduce processing time.
    3. Each video is grouped under a category like Arson, Explosion, Fighting, or Stealing.
2. **Learning from Image Details using CNN:**
    1. The system looks at each frame and learns important details like objects, shapes, and patterns.
    2. This helps the system understand what is happening in each image.
3. **Learning from Movement and Actions using LSTM:**
    1. The system checks how the scene or actions change from one frame to the next.
    2. This helps in understanding if something unusual is happening over time.
4. **Detecting and Categorizing Anomalies:**
    1. The system combines what it learned from the images and their movement to decide if something wrong or suspicious is happening.
    2. It then classifies the event into one of the four categories: Arson, Explosion, Fighting, or Stealing.
5. **Using a Clean and Balanced Dataset:**
    1. The system uses a selected portion of a large video dataset that only contains clear examples of the above four categories.
    2. This helps the system learn better and avoid confusion from unclear or unrelated videos.

6. **Providing Results and Reports:**

    1. The system gives the final output by showing which type of anomaly was detected.
    2. It can also show some key frames from the video to explain its prediction.

3.2 NON-FUNCTIONAL REQUIREMENTS

Non-Functional Requirements define how the system should operate., Ensure it is reliable, scalable, and user-friendly.

1. **Performance Requirements**

    1. The system should analyze video clips and provide anomaly results within a reasonable time without long delays.
    2. It should be able to process videos of different sizes and qualities without slowing down.
    3. Multiple videos should be handled one after another efficiently without crashing the system.

2. **Accuracy and Reliability Requirements**

    1. The system should detect and classify anomalies correctly, aiming to reduce incorrect or missed detections.
    2. It should run smoothly and consistently without unexpected shutdowns or major errors.
    3. There should be mechanisms in place to restart or recover if the system fails during video analysis.

3. **Scalability Requirements**

    1. The system should be able to handle more video data in the future without losing performance.
    2. It should work well even if more users or video files are added to the system.

3. Data storage and processing should scale up easily with the help of cloud-based platforms when required.

4. **Usability Requirements**
    1. The system's interface should be easy to navigate, even for non-technical users.
    2. It should clearly display the analysis results and make it simple to understand which type of anomaly was detected.
    3. The design should focus on clean visuals and logical flow to help users work with minimal training.
5. **Maintainability Requirements**
    1. The system should be built in a way that future updates and bug fixes can be done easily.
    2. It should have clear documentation for developers to refer to while updating or maintaining the system.
    3. Logs should be maintained to track any issues or errors during execution for quicker resolution.

3.3 HARDWARE REQUIREMENTS:

The system requires specific hardware components to ensure smooth video processing, model training, and efficient system operation. Below are the required hardware specifications:

1. **Laptop/PC Configuration:**
    1. **Processor:** Intel i5 (10th Gen) / Ryzen 5 or higher.
    2. **RAM:** Minimum 8 GB (Recommended: 16 GB for smoother model testing).
    3. **Storage:** SSD 256 GB+ (to manage datasets and code effectively).
    4. **OS:** Windows 10/11 or Ubuntu 20.04+.
    5. Required for local development, testing, and running the Python-based interface.

2. **Display and Visualization Devices:**

    1. A monitor with minimum 1080p resolution for viewing outputs clearly.
    2. Optional external display for comparing frame-by-frame anomaly detection outputs.

3.4 SOFTWARE REQUIREMENTS

The system uses several software components for model development, video processing, and user interaction. Below are the required software tools:

1. **Programming Language:**

    Python 3: Used for implementing the entire system including model integration and interface development.

2. **Deep Learning Framework:**

    TensorFlow: Used for building and training the CNN-LSTM model for video anomaly detection.

3. **Libraries and Dependencies:**

    1. OpenCV: For video frame processing and manipulation.
    2. NumPy: For handling numerical operations and array manipulations.
    3. OS: For managing file paths and accessing system directories.

4. **User Interface:**

    Python-based Interface: Simple interface created using Python for users to view flagged anomalies and visualize results. No third-party web frameworks used.

5. **Model Training Platform:**

    Kaggle Notebook (Cloud-based with integrated GPU support): It provides temporary file storage and resource-limited execution environment for model development. It also provides two Nvidia T4 GPU, free-of-cost for 30 hours, which are crucial for machine learning applications.

6. **Development Tools:**

    1. Jupyter Notebook: Used within Kaggle for code development and model training.
    2. Anaconda: Optional for managing the Python environment during local testing.

Chapter 4

SYSTEM DESIGN

4.1 HIGH-LEVEL DESIGN

The proposed Video Anomaly Detection System is designed to automatically identify unusual or suspicious activities in surveillance footage using a deep learning-based approach. The system designed in figure 4.1 begins by extracting frames from the input video, which are then processed using a convolutional neural network to capture spatial features from each frame. These features are passed into a recurrent neural network that analyzes the sequence of frames to understand temporal relationships and detect abnormal patterns. A classification layer determines whether an anomaly is present, and if so, categorizes the type of anomaly, such as theft, fight, or fire Anomalies detected are clearly marked in the video with outputs displayed on an easy-to-use interface for users to easily follow the outputs. This complete end-to-end solution improves the surveillance capturing intelligent analysis of the feed to assist security personnel in the identification and response to critical events.

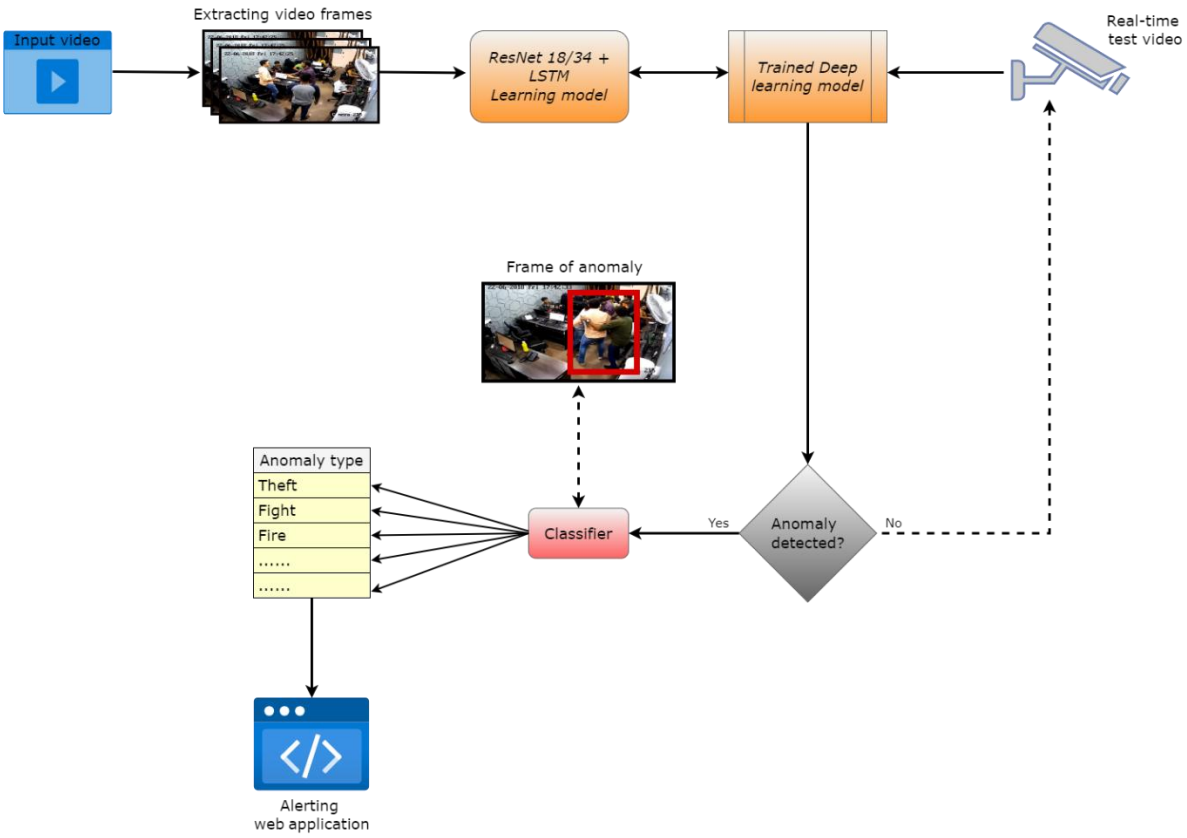

Figure 4.1: High-Level Design.

4.2 LOW-LEVEL DESIGN

The low-level architecture of the VSAD (Video Surveillance Anomaly Detection) system (in figure 4.2) processes video streams to detect anomalous activities via a hybrid deep learning pipeline. The inputs to the system come from video surveillance cameras that are located in the environment, and the video streams are processed as a series of frames, or individual images. The preprocessing module applies the processing and segmentation to each consecutive video frames - for example, converting color to grayscale, resizing images and sampling frames to minimize redundancy and support computational efficiency.

Each pre-processed frame is passed to a pre-trained ResNet (Residual Network) model that serves as the backbone for the extraction of spatial features. ResNet has skip connections that help avoid vanishing gradients by utilizing deep networks with residual points in order to extract a rich hierarchy of features like edges, shapes, object-level semantics, etc., and does this for each frame independently. Each of the extracted features are then summarized based on the sequence of frames in the original video.

To model temporal dependencies and motion dynamics among the sequences of feature vectors, the features sequence is fed into an LSTM network. The LSTM has been shown to learn long-term dependencies and temporal correlations, making it a suitable choice for detecting patterns over time such as whether a person is loitering, stealing, or fighting. The LSTM module's output will be a hidden state with the learned temporal context representation.

The final classification layer consists of fully connected layers generating a softmax output layer to predict a probability distribution over the classes: Arson, Explosion, Fighting, Stealing, or Normal behaviour. The predicted output label will be the class with the highest predicted probability. Moreover, the architecture includes several thresholds, or post-processing, for high-risk behaviors to trigger an alert.

Training-time additions such as dropout and batch normalization, respectively, can also mitigate overfitting and enable convergence to be sped up. All of this can happen in real-time (or near real-time) depending on hardware configurations and optimization strategies like model quantization or frame front skipping. Thus, this low-level

architecture provides a strong basis for deep learning-based intelligent video surveillance applications.

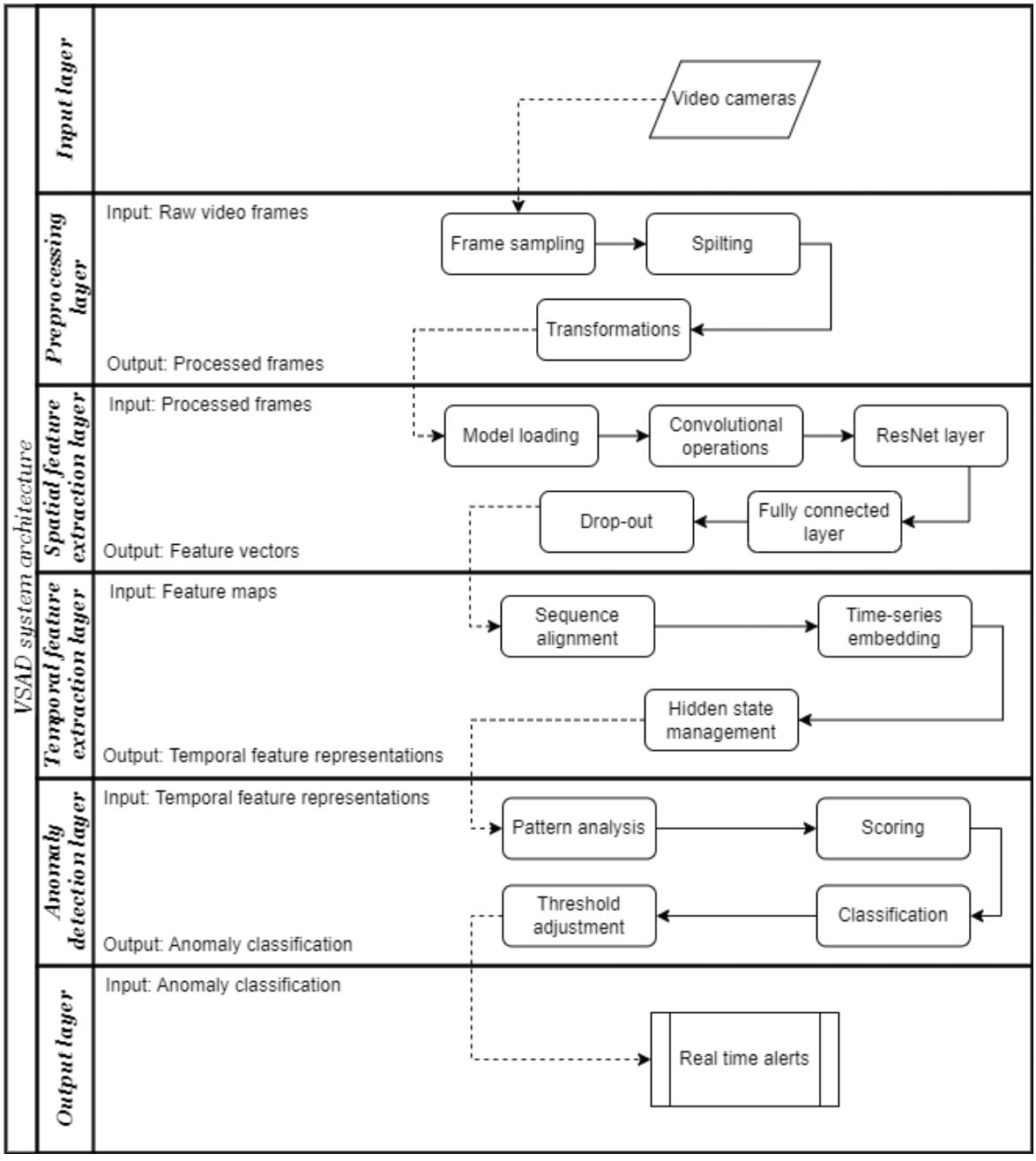

Figure 4.2: Low level design.

Chapter 5

IMPLEMENTATION

5.1 METHODOLOGY

The Video Anomaly Detection System proposed will use a deep-learning-based architecture of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The presented pipeline involves video pre-processing, where surveillance videos are sampled into frames that are evenly spaced in grayscale (usually 100 per clip). To reduce the computational load of future learning, the frames are then resized to 64×64 pixels. Since only the segments of videos where action is relevant are used (from the UCF-Crime dataset), only these smaller segments will be sampled into frames and then multiple anomalies will be labelled into the categories: Arson, Explosion, Fighting and Stealing.This focused selection enhances training accuracy and ensures that the model learns to identify key anomaly types.

The next task is spatial feature extraction using a ResNet-18 model, modified to take as input single-channel grayscale images instead of RGB images.Transfer learning is employed here—ResNet-18 is pre-trained on the ImageNet dataset, with its final classification layer replaced by a custom 256-dimensional feature vector output. This adaptation allows the model to learn spatial features like textures, object edges, and motion cues efficiently. These spatial vectors are then reshaped to vectors of all of the spatial features, and sent to a two-layer LSTM network with a hidden size of 128. The LSTM, while processing these spatial features, will begin to learn about the temporal dynamics (such as sudden motions or transitions in a scene) highlighting abnormalities from normal activities.

Finally, the spatial and temporal outputs are concatenated into a 640-dimensional feature vector and forwarded to a fully connected classification layer that uses a cross entropy loss function during training to map the learned features to one of four defined classes of anomalous behaviour. The model learns to distinguish between normal versus suspicious behaviour. The end-to-end system learns the two steps and captures both frame-level spatial behaviours and time-dependent behaviours for robust, real-time anomaly detection in surveillance streams.

5.1.1 Video processing

Good pre-processing of videos is important to the functionality and efficiency of the CNN + LSTM-based anomaly detection system being proposed in this chapter. For the sake of standardisation and computational efficiency, each video in the dataset was converted to the grayscaled format to reduce the capacity of the video file while retaining the important visual features (including the edges, textures and motion). From each video, 100 frames were randomly drawn, spaced evenly in the length of the videos regardless of their original length. This helps to ensure time covering across the entire video without selecting frames redundantly or drawing unnecessary data. All frames were resized to the standard size of 64×64 pixels, which would reduce the dimensionality of the data fed to the model while still allowing the model to effectively learn key visual behaviours..

Each extracted frame is then labeled according to the video's corresponding anomaly category—Arson, Explosion, Fighting, or Stealing—forming the basis for supervised learning. A balanced subset of the UCF-Crime dataset is used to avoid class imbalance and ensure consistent training performance across categories. Rather than using full-length surveillance videos, which often contain long durations of irrelevant or uneventful content, only action-focused segments where the anomaly is clearly observable are selected. This targeted approach enhances model focus, reduces noise, and significantly improves training efficiency. Additionally, it helps preserve the most relevant temporal and spatial context, which is essential for the LSTM component to learn motion and sequence-based patterns effectively. Overall, this pre-processing strategy forms a solid and optimized foundation for training the anomaly detection system.

Fig. 5.1.1 Output of video preprocessing layer

5.1.2 ResNet-18 – Spatial Feature Extraction

ResNet-18 (Residual Network) is a deep Convolutional Neural Network used to extract spatial features from individual video frames. Unlike traditional CNNs, ResNet introduces skip connections or residual links that help in training deeper networks by preventing vanishing gradient problems. ResNet-18, specifically, consists of 18 layers including convolutional, batch normalization, and fully connected layers.

In this project, the ResNet-18 model is modified to accept grayscale input, since the dataset frames are single-channel (instead of RGB). The first convolutional layer is adapted accordingly. The model is pre-trained on the ImageNet dataset, which allows it to benefit from transfer learning — meaning it already understands low-level visual patterns like edges and textures. The final classification layer of ResNet-18 was modified to produce a 256-dimensional vector for each frame. Each output vector contains spatial information (e.g., object shapes, contextual background) and potentially additional cues with regard to motion of objects. The output vectors were that provided to the temporal model, LSTM, as input.

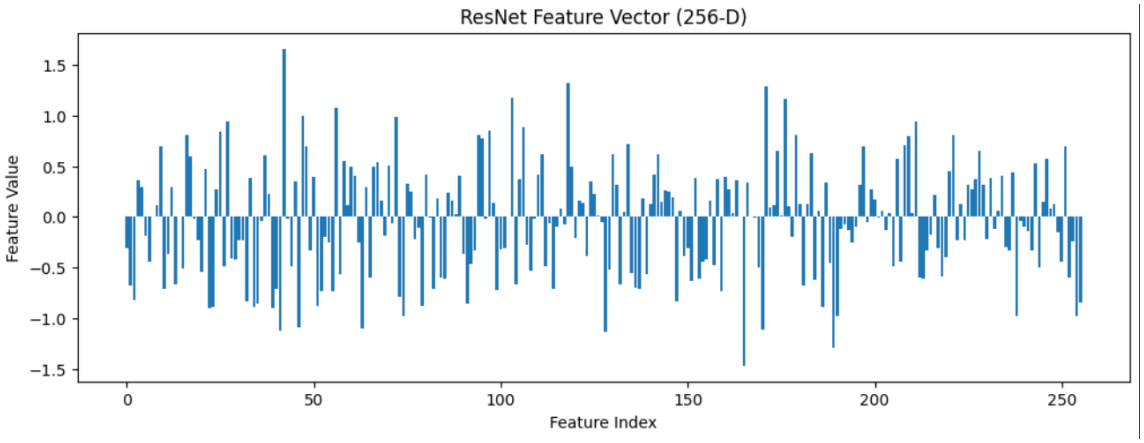

Fig 5.1.2: 256- dimensional vector for a sample image frame

5.1.3 LSTM – Temporal Feature Extraction

LSTMs, long short-term memory or LSTMs, are recurrent neural networks (RNNs) which can learn time dependencies in sequences and maintain them. LSTMs can overcome the vanishing gradient problem that RNNs have architecturally, and possess offset gates (input, forget, and output) that control the flow of information in time.

In our system, LSTM is used to process the sequence of 512-dimensional spatial vectors produced by ResNet-18. The feature vector for each frame is provided in sequential order to the LSTM, which is composed of 2 layers, each with a hidden size of 128. These layers learn how frame features change over time; for example, recognizing a sudden change in motion or behavior that could indicate an anomaly such as an explosion or fighting. The LSTM effectively captures each frame with respect to the frames before it: that is, it is essential to depict normal actions (walking) as different from abnormal actions (running from a fire). Recall that the output of LSTM is a 128 dimensional temporal feature vector summarizing the sequence behavior that was learned.

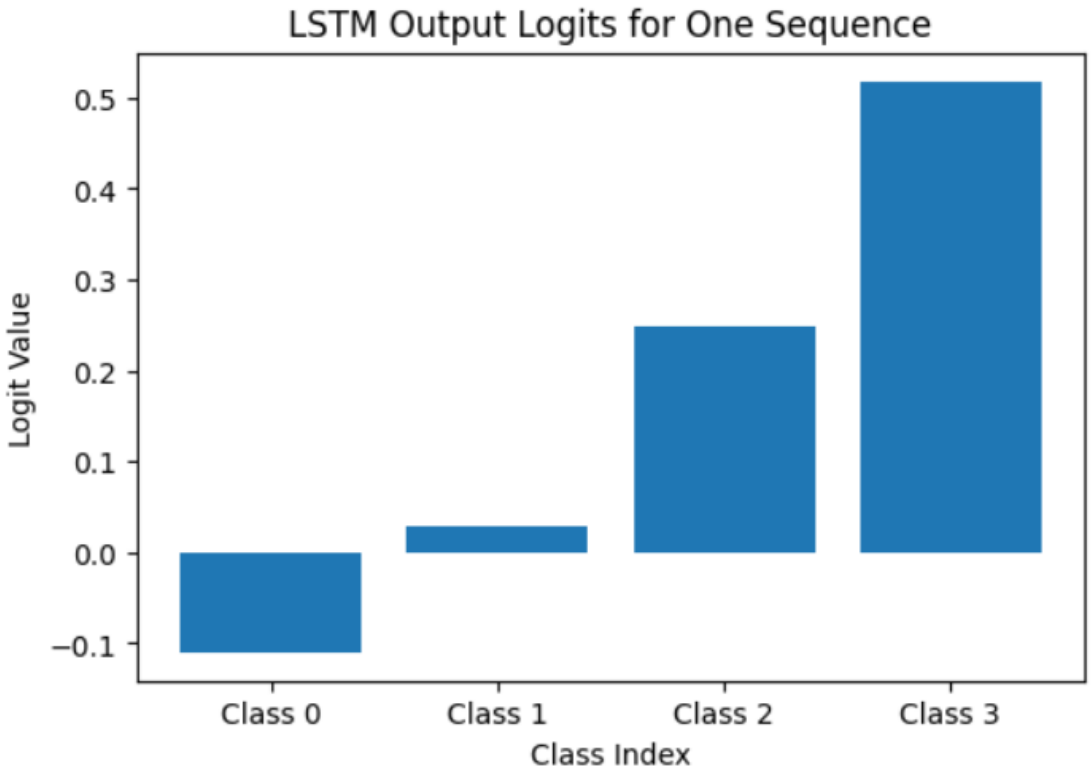

Fig 5.1.3: Probability of input sequence belonging to certain class

5.1.4 Classification of Anomalies

The classification layer represents the final decision-making component in the anomaly detection pipeline, combining the strengths of spatial and temporal feature extraction. After passing the video frames through the ResNet-18 and LSTM layers, each of the respective outputs are obtained: a 512-dimensional spatial feature vector from ResNet-18 and a 128-dimensional temporal feature vector from LSTM Network, where the two vectors represent the static visual cues and the progression of the sequential video frames. Combining these two vectors provides a unified 640-dimensional feature vector. This feature vector captures the complete rationalization of the video segment in question, including whether an object existed, behavioural motion and the progression of actions over time.

This 640-dimensional vector passes through a fully connected linear (dense) layer that acts as the classifier. The dense layer then makes the final predictions based on the probabilities for each of the four anomaly classes: Arson, Explosion, Fighting or Stealing. To facilitate learning during training, We use a cross-entropy loss function

that outputs the difference between the predicted class probabilities and the actual ground truth labels. This loss can be minimized over the continual update through back propagation, allowing the model to learn the correct mapping to the classes and to generalize well on unseen video input. The classification layer acts as a connector of spatial features (what is happening in the framed) and temporal features (how the action is being executed over time) so the model is able to detect anomalies reliably with contextual relevance in a surveillance environment.

5.2 DATASET DETAILS AND SPLITS

The project utilizes a curated subset of the UCF-Crime dataset, focusing on four key anomaly categories: Arson, Explosion, Fighting, and Stealing. From the full set of 14 classes, 50 representative videos per selected category were extracted for training and testing. To maintain uniformity and reduce computational load, grayscale frames were extracted and resized to 64×64 pixels. This preprocessing step helped remove redundant or noisy segments and ensured class balance.

The training dataset included approximately 19,000 frames (~4,700–4,800 per class), and the test dataset comprised 1,000 frames (~200–300 per class). Frames were sampled at 10-frame intervals to preserve temporal relevance while avoiding repetition, laying a reliable foundation for training an effective anomaly detection system.

```
# Dataset paths
DATA_SOURCE = {
    "Arson": "/kaggle/input/anomalydetectiondatasetucf/Anomaly-Videos-Part-1/Anomaly-
Videos-Part-1/Arson",
    "Explosion": "/kaggle/input/anomalydetectiondatasetucf/Anomaly-Videos-Part_2/Anom
aly-Videos-Part-2/Explosion",
    "Fighting": "/kaggle/input/anomalydetectiondatasetucf/Anomaly-Videos-Part_2/Anoma
ly-Videos-Part-2/Fighting",
    "Stealing": "/kaggle/input/anomalydetectiondatasetucf/Anomaly-Videos-Part-4/Anoma
ly-Videos-Part-4/Stealing",
}
```

Fig 5.2: Dataset path added for training

5.3 TRAINING SETUP

The model was trained using a hybrid CNN–LSTM architecture where a modified ResNet-18 extracted spatial features and a two-layer LSTM captured temporal dependencies. The training was performed over 15 epochs, with early stopping enabled at epoch 3 to avoid overfitting. A batch size of 64 and an initial learning rate of 0.001 were used with the Adam optimizer. The cross-entropy loss function was used to compare predicted and actual labels.

Transfer learning was employed using ImageNet pre-trained weights for the ResNet-18 backbone. GPU acceleration was used during training in Kaggle Notebook, which in turn enabled faster training and better feature extraction performance.

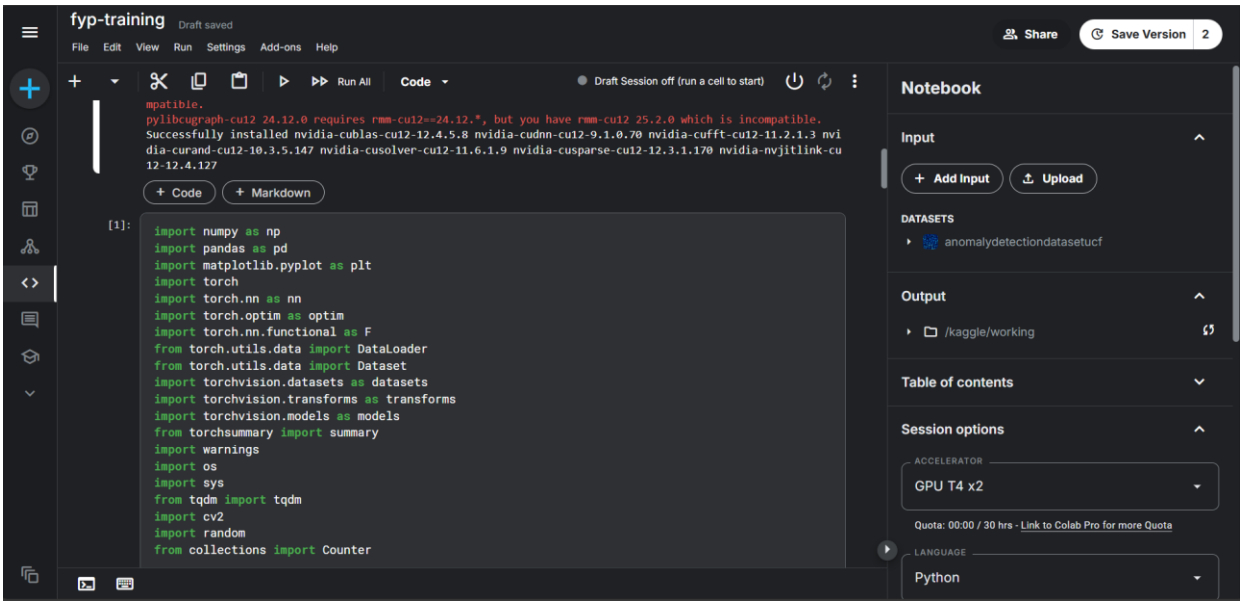

Fig 5.3.1: Training setup(Kaggle notebook)

```
Model will be trained with 4 classes
Training model...
Epoch 1/15...
Batch: [20/297], Loss: 0.4741
Batch: [40/297], Loss: 0.5115
Batch: [60/297], Loss: 0.2353
Batch: [80/297], Loss: 0.1749
Batch: [100/297], Loss: 0.1604
Batch: [120/297], Loss: 0.1114
Batch: [140/297], Loss: 0.1548
Batch: [160/297], Loss: 0.0421
Batch: [180/297], Loss: 0.1179
Batch: [200/297], Loss: 0.1908
Batch: [220/297], Loss: 0.1217
Batch: [240/297], Loss: 0.0793
Batch: [260/297], Loss: 0.0725
Batch: [280/297], Loss: 0.0754
Epoch 1 results:
  Train Loss: 0.2328, Train Accuracy: 92.08%
  Val Loss: 3.2008, Val Accuracy: 24.20%
  New best model saved! (Val Loss: 3.2008)
```

Fig 5.3.2: Training the model(first epoch) and saving the best model

```
Epoch 5/15...
Batch: [20/297], Loss: 0.0056
Batch: [40/297], Loss: 0.0216
Batch: [60/297], Loss: 0.0365
Batch: [80/297], Loss: 0.0124
Batch: [100/297], Loss: 0.0145
Batch: [120/297], Loss: 0.0543
Batch: [140/297], Loss: 0.0011
Batch: [160/297], Loss: 0.0344
Batch: [180/297], Loss: 0.0115
Batch: [200/297], Loss: 0.0019
Batch: [220/297], Loss: 0.0034
Batch: [240/297], Loss: 0.0040
Batch: [260/297], Loss: 0.0442
Batch: [280/297], Loss: 0.0008
Epoch 5 results:
  Train Loss: 0.0323, Train Accuracy: 99.03%
  Val Loss: 1.3869, Val Accuracy: 62.20%
  Patience: 3/3
Early stopping triggered after 5 epochs
```

Fig 5.3.3: Early stopping of the training process

Chapter 6

RESULTS AND DISCUSSIONS

The developed video anomaly detection system was a hybrid CNN-LSTM model which had sufficient performance and recognized suspicious behaviors in video. By utilizing both spatial and temporal deep learning methods the model provided a test accuracy of 62.20%, and had relatively high recall rates against visually different labels like Arson (95.5%) and Stealing (81.0%). The training accuracy was eventually reached, reporting at an accuracy level of 92.08% and early validation suggested we achieved an accuracy of 75.5% which suggested learning features well.However, performance was affected in cases involving visually similar anomalies like Explosion and Fighting, revealing limitations in class differentiation. Despite these challenges, the system's integration with a real-time interface and email alert mechanism significantly enhanced its practical utility. Overall, the results highlight the potential of deep learning in automating surveillance and anomaly detection, while also pointing to future opportunities for improving model generalization, precision, and scalability.

6.1 TRAINING AND VALIDATION CURVES

The model's performance was tracked across each epoch using accuracy metrics. During the final epoch, training accuracy reached 92.08%, while validation accuracy was slightly lower at 24.2%, indicating potential overfitting. However, early in training (by epoch 2), the model showed significant improvement, reaching 97.83% training accuracy and 75.5% validation accuracy. The loss curves revealed good convergence patterns, although fluctuations in validation loss indicated the model's sensitivity to generalization. These observations validated the model's ability to learn spatial-temporal patterns quickly from moderately sized training data.

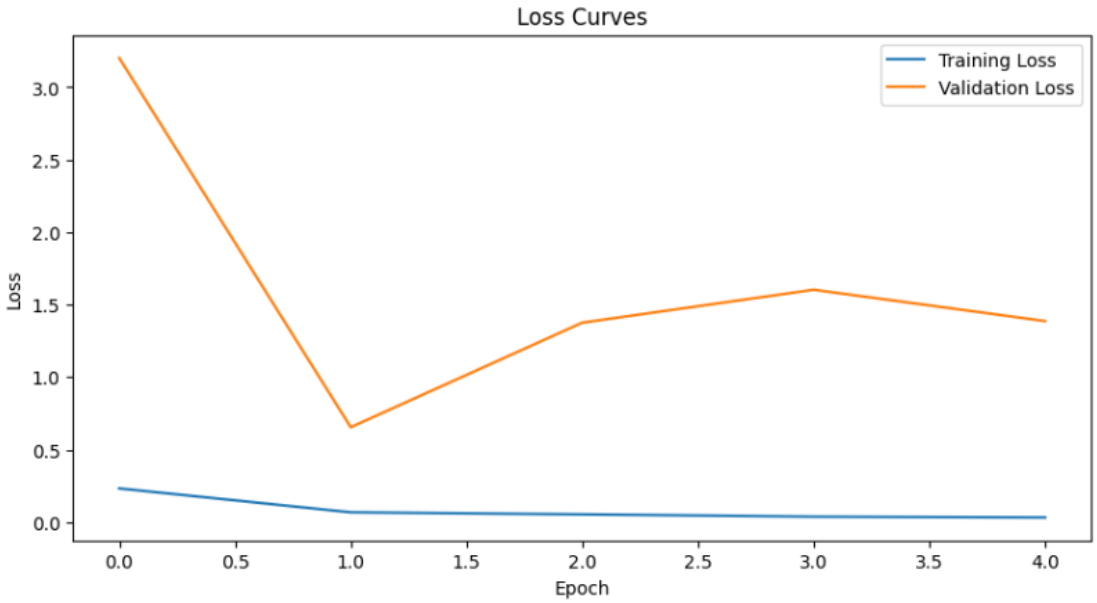

Fig. 6.1: Loss curve

6.2 TEST SET EVALUATION

Evaluation on the test dataset yielded a 62.2% overall accuracy, using the equation 6.2, confirming the model's effectiveness in classifying video anomalies. A confusion matrix was generated to assess per-class prediction performance. The model showed high recall for Arson (95.5%) and Stealing (81.0%), attributed to distinct visual and motion patterns.

However, Explosion and Fighting categories showed misclassifications due to visual similarity — e.g., Explosion frames misinterpreted as Fighting. These overlaps highlight the challenge of distinguishing visually similar classes and suggest further improvement areas.

$$\text{Accuracy} = \frac{\text{Total Number of Predictions}}{\text{Number of Correct Predictions}} = \frac{\sum_{i=1}^{N} 1(y_i = \hat{y}_i)}{N} \quad (6.2)$$

Where:

1) y_i is the ground truth label for sample i
2) $\hat{y}_i$ is the predicted label for sample i
3) $\mathbf{1}(\cdot)$ is the indicator function (1 if the condition is true, 0 otherwise)

4) **N** is the total number of samples

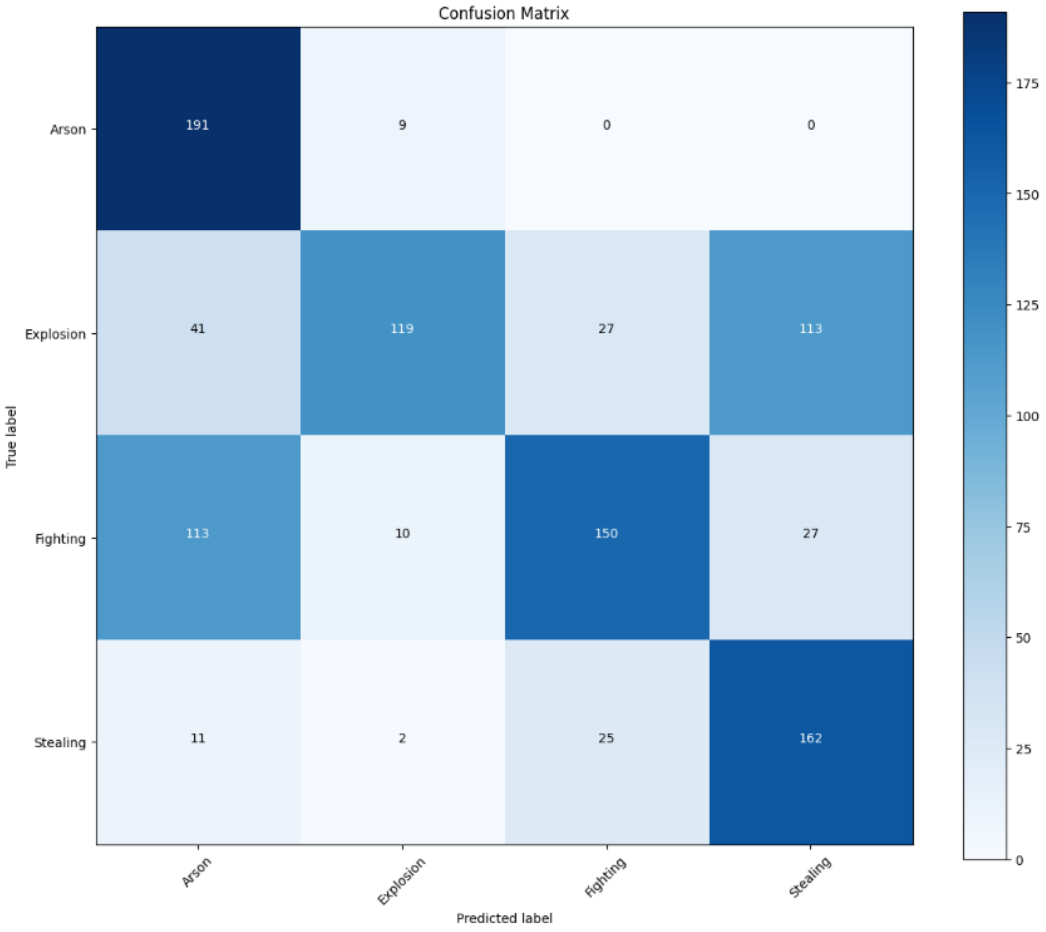

Fig. 6.2: Confusion matrix

6.3 PER-CLASS METRICS

In this project, per-class evaluation metrics-Precision, Recall, and F1 Score-were computed for each anomaly class: Arson, Explosion, Fighting, and Stealing. These metrics help assess how well the model performs for each class individually, rather than just evaluating its overall accuracy.

1) Precision (equation 6.3.1) indicates how many of the instances predicted as a certain class were actually correct. A high precision means fewer false positives.
2) Recall (equation 6.3.2) measures how many actual instances of a class were correctly identified. A high recall means fewer false negatives.

3) F1 Score (equation 6.3.3) is the harmonic mean of precision and recall, providing a single metric that balances both concerns, especially when they trade off against each other.

For example, in the results, the Arson class has high recall (0.9550) but lower precision (0.5365), which means the model detects most arson cases but also incorrectly labels some non-arson videos as arson. Explosion shows the opposite—high precision (0.8500) but low recall (0.3967), suggesting it predicts explosion cases accurately when it does, but misses many real explosion cases. Fighting and Stealing have more balanced or moderate performance, which helps identify areas where the model may need improvement or rebalancing. These class-wise insights are crucial for refining the anomaly detection system.

$$\text{Precision}_\text{i} = \frac{TP_i}{TP_i + FP_i} \tag{6.3.1}$$

$$\text{Recall}_\text{i} = \frac{TP_i}{TP_i + FN_i} \tag{6.3.2}$$

$$\text{F1 score}_\text{i} = \frac{2 \times \text{Precision}_\text{i} \times \text{Recall}_\text{i}}{\text{Precision}_\text{i} + \text{Recall}_\text{i}} \tag{6.3.3}$$

Where:

1) TP_i: True Positives for class i
2) FP_i: False Positives for class i
3) FN_i: False Negatives for class i
4) If $\text{Precision}_\text{i} + \text{Recall}_\text{i} = 0$, then $\text{F1 score}_\text{i} = 0$

Class	Precision	Recall	F1 Score
Arson	0.5365	0.9550	0.6871
Explosion	0.8500	0.3967	0.5409
Fighting	0.7426	0.5000	0.5976
Stealing	0.5364	0.8100	0.6454

Table 6.3: Precision, Recall, F1 score values for each class

Fig. 6.3: Prediction results for random frames

(Actual class → Predicted class)

6.4 ANOMALY DETECTION INTERFACE

The developed interface supports frame-by-frame analysis of uploaded video clips using the trained CNN–LSTM model. It displays detection status, anomaly type, and confidence scores. A supporting activity log panel tracks system actions like model loading, frame extraction, anomaly classification, and completion status. This interfaces visualizes the system's decision-making and identifies significant anomalous events.

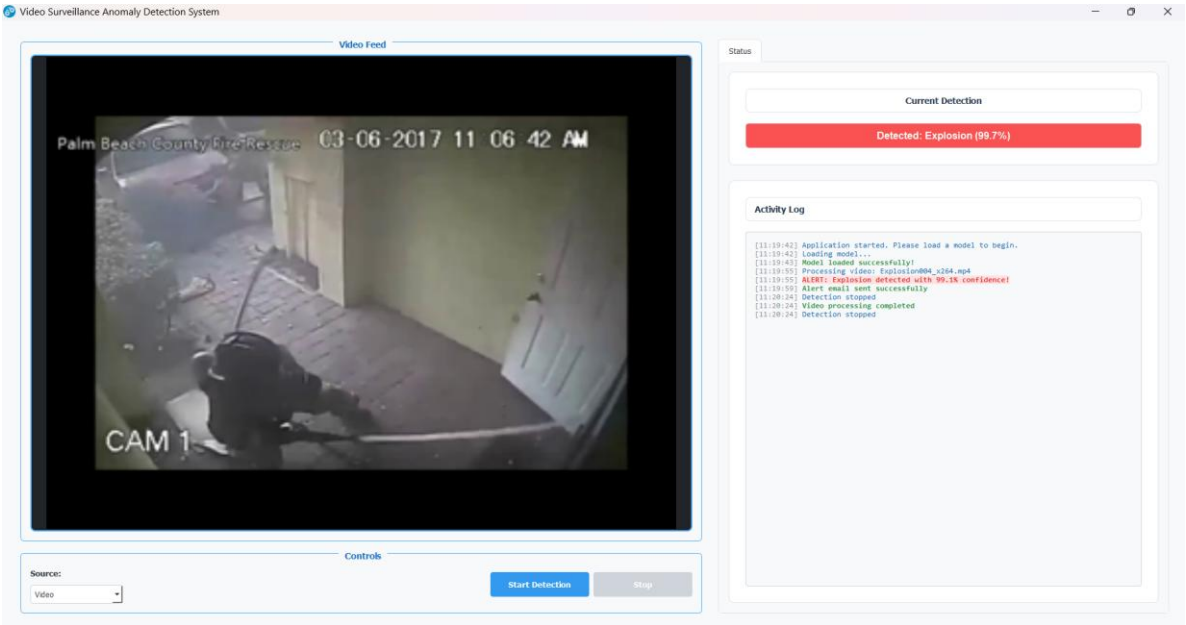

Fig. 6.4: User Interface

6.5 EMAIL ALERT SYSTEM

With an integrated email notification platform, the framework took on a dimension of real-time responsiveness. Multiple incident types with high confidence could prompt the system to send out an email, to notify the analyst, that denotes a type of anomaly, the time of detection, a confidence score, as well as a snapshot of a frame. This enables immediate communication to the analyst and provides a timely review of the incident by the analyst. As an example, the detection of "Explosion" at 99.1% confidence triggered a timely alert that served to simulate a real-time response that can be critical in perimeter surveillance applications.

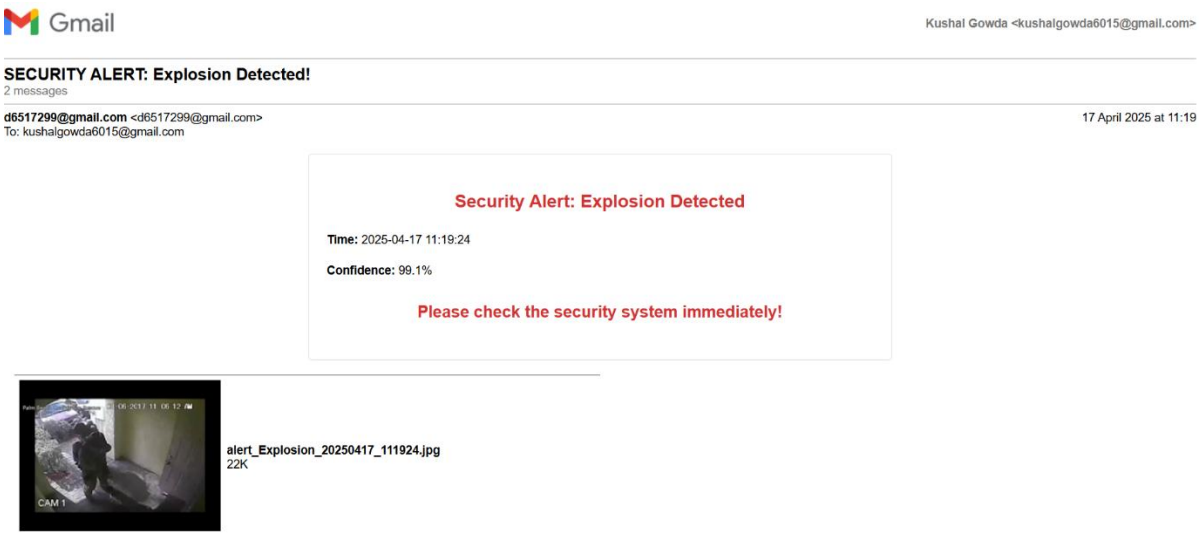

Fig. 6.5: Email notification

Chapter 7

CONCLUSION AND FUTURE SCOPE

In conclusion, the project applies deep learning based video anomaly detection system utilising a hybrid CNN–LSTM architecture. In terms of anomaly detection, it demonstrated an ability to learn spatial and temporal patterns in surveillance footage and important anomaly events including arson, stealing, fighting and explosions. It also demonstrated a real time live monitoring system with the combination of a real time interface and an email alerting system. Whilst good balanced accuracies and recall had been achieved for some incidents, some trouble in classifying visually similar incidents demonstrated a need for improved precision and model robustness in future work.

There will be many improvements that can be made in the future that will enhance performance and operation of the system. First, implementing processes to enable the real-time processing of video streams will allow the system to ingest live video streams from surveillance video cameras to allow for detection of anomalous events in real-time and subsequently allow the system a faster response time. In the event we trained on live events that were associated with anomalous events, the additional category of anomalous events could create additional diversity and flexibility in the system. Furthermore, implementing other procedures to improve model performance such as hyperparameter tuning, data augmentation, and attention mechanisms are all key to improving the model performance. In particular, attention layers would allow the model to focus on frames of video that contained the contextual information pertinent to lowering detection performance in highly complex frames. These extensions present interesting avenues for further development of reliable intelligent surveillance systems being deployed in high-dynamic contexts and high-threat environments.

REFERENCES

[1] M. Asim and E. Verdu, "Video anomaly detection system using deep convolutional and recurrent models," *Results Eng.*, vol. 17, 2023.

[2] J. L. S. Gonzalez, J. A. Alvarez-Garcia, F. J. Rendon-Segador, and F. Carrara, "Conditioned cooperative training for semi-supervised weapon detection," *Neural Netw.*, vol. 164, pp. 294–305, 2023.

[3] K. V. Hakare, D. P. Dogra, H. Choi, H. Kim, and I. J. Kim, "RareAnom: A benchmark video dataset for rare type anomalies," *Pattern Recognit.*, vol. 137, 2023.

[4] A. Mahmood, "Abnormal behavior detection in uncrowded videos with two-stream 3D convolutional neural networks," *Appl. Sci.*, vol. 11, no. 9, 2021.

[5] A. Man and G. M. Khan, "Real-world malicious event recognition in CCTV recording using Quasi-3D network," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, pp. 1245–1259, 2022.

[6] S. Mishra and S. Mishra, "Anomaly recognition from surveillance videos using 3D convolution neural network," *Proceedings of the 6th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1102–1107, 2021. doi: 10.1109/ICCES51350.2021.9488972

[7] S. Ozimi, D. D. Sio, F. Carlucci, and L. Sterpone, "Fighting for a future free from violence: A framework for real-time detection of 'Signal for Help'," *Intell. Syst. Appl.*, vol. 15, 2023.

[8] G. Tang, H. Ding, M. Duan, Y. Pu, Z. Yang, and H. Li, "Fighting against terrorism: A real-time CCTV autonomous weapons detection based on improved YOLO v4," *Digit. Signal Process.*, vol. 134, 2023.

[9] W. Ullah, T. Hussain, and S. W. Baik, "Vision transformer attention with multireservoir echo state network for anomaly recognition," *Inf. Process. Manag.*, vol. 60, no. 3, 2023.

[10] W. Ullah, F. U. M. Ullah, Z. A. Khan, and S. W. Baik, "Sequential attention mechanism for weakly supervised video anomaly detection," *Expert Syst. Appl.*, vol. 213, 2023.

[11] W. Ullah, T. Hussain, Z. A. Khan, U. Haroon, and S. W. Baik, "Intelligent dual stream CNN and echo state network for anomaly detection," *Knowl.-Based Syst.*, vol. 249, 2022.

[12] S. A. Umon, R. Goni, N. B. Hashem, M. T. Shahria, and R. M. Rahman, "Violence detection by pretrained modules with different deep learning approaches," *Vietnam J. Comput. Sci.*, 2019.

[13] Y. Yao, A. Dong, Z. Liu, Y. Jiang, Z. Guo, J. Cheng, Q. Guan, and P. Luo, "Extracting the pickpocketing information implied in the built environment by treating it as the anomalies," *Cities*, vol. 139, 2023.