A Project Report

on

CrimeGuard

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Computer Science and Engineering (Artificial Intelligence & Machine Learning)

By

Shreya Goswami (2100291530051)

Khushi Bansal (2100291530028)

Mahi Tyagi (2100291530032)

Under the supervision of

Mr Rajeev Kumar Singh

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature
Name: Shreya Goswami
Roll No.: 2100291530051
Date:

Signature
Name: Khushi Bansal
Roll No.: 2100291530028
Date:

Signature
Name: Mahi Tyagi
Roll No.: 2100291530032
Date:

CERTIFICATE

This is to certify that Project Report entitled "CrimeGuard" which is submitted by Shreya Goswami, Khushi Bansal and Mahi Tyagi in partial fulfillment of the requirement for the award of degree B. Tech. in Department of CSE(AIML) of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Mr Rajeev Kumar Singh	**Dr. Rekha Kashyap**
(Designation)	**(Head of Department)**

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Mr Rajeev Kumar Singh, Department of CSE(AIML), KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Rekha Kashyap, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature
Name: Shreya Goswami
Roll No.: 2100291530051
Date:

Signature
Name: Khushi Bansal
Roll No.: 2100291530028
Date:

Signature
Name: Mahi Tyagi
Roll No.: 2100291530032
Date:

ABSTRACT

CrimeGuard is a comprehensive system which is designed for predictive crime analysis and real-time violence detection using Machine Learning and Computer Vision. The project has the goal to maximize public safety by using historical crime data to predict potential crime hotspots, and real-time video analysis to detect violent activities. The framework involves deep learning frameworks such as VGG16 for feature extraction and custom-trained neural networks to identify violent activity from CCTV videos. Predictive modeling relies on ARIMA to analyze crime patterns and forecast future crime rates. CrimeGuard has an interactive dashboard for visualizing crime data to enable law enforcement agencies to make informed decisions based on evidence. Created with Flask as the backend and HTML, CSS, JavaScript as the frontend, CrimeGuard is an effective and scalable solution to enable crime prevention and response.

TABLE OF CONTENTS

Page No.

.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
CNN	Convolutional Neural Network
BiLSTM	Bidirectional Long Short-Term Memory
ReLU	Rectified Linear Unit SMTP Simple Mail Transfer Protocol
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
VGG	Visual Geometry Group
RNN	Recurrent Neural Network
RLVS	Real-Life Violence Situation

CHAPTER 1
INTRODUCTION

1.1 Overview

With the advent of this fast-changing digital era, the task of policing and security institutions becomes very challenging as far as stopping crimes and real-time detection of threats are concerned. Surveillance activities based on traditional techniques and direct observation often become useless and cause untimely responses along with overlooking the critical incidents. Besides, correlating huge data regarding crimes and detection of pattern for anticipating threats in the future still becomes challenging.

CrimeGuard prioritizes enhanced public safety using real-time automated detection of violence and predictive crime forecasting. It performs video streams via a well-trained CNN+BiLSTM model to recognize violent activities and trigger real-time alarms. Crime prediction models are also integrated in it to validate past crime trends and identify risk areas. Comparative study with other models like VGG16 and ResNet is sure to retain the effectiveness of the introduced architecture. The system reduces manual monitoring to a great extent, providing a faster, more accurate response to security threats and providing crime pattern insights in the form of an interactive visualization module.

1.2 Project Description

CrimeGuard is an AI platform for augmenting public safety through predictive crime analysis and real-time violence detection. It uses machine learning algorithms to examine past patterns of crime and forecast high-risk areas. At the same time, it uses deep learning-based computer vision to track live video feeds and detect violent activity such as assaults and brawls.After the detection of suspicious activity, the system triggers automatic alerts so that the law enforcement agency takes quick action.

The architecture is developed with OpenCV for image analysis and video analytics, TensorFlow to detect violence with deep learning-based detection, and Scikit-learn to conduct crime pattern analysis and predictive analytics. The front end is designed using React with an interactive dashboard through which law enforcement agencies visualize crime trends, analyze historical patterns, and are alerted in real-time. The smart strategy in CrimeGuard delivers proactive crime prevention, maximizing the deployment of resources and response levels.

Blending deep learning, machine learning, and live video monitoring, CrimeGuard is a high-impact, scalable, and efficient crime prevention and detection solution. Through its automated alerting function and AI-driven insights, CrimeGuard allows law enforcement to make data-driven, informed decisions, resulting in safer communities and more effective crime management planning.

1.3 Problem Statement

Crime detection and prevention remain principal concerns due to the limitations in manual surveillance and delayed response time. Current systems are not competent in real-time detection, multi-environment variability, and prognostic crime analysis. A need exists for an efficient solution that can

Process real-time video streams to identify violent crimes. Predict hotspots of high crime based on historical patterns. Alert authorities in real-time for quick response. Visualize crime insights interactively for better decision-making. By bridging these gaps, CrimeGuard offers quicker response times, improved situational awareness, and proactive crime prevention.

1.4 Existing System

Current crime prevention and detection means rely on surveillance or rule-based methods, the shortcomings of which are as follows:

Human Error – Fatigue of surveillance officers and therefore, incidents remain undetected.Delayed Response – Inability to automate leads to slow response times.

Limited Predictive Analysis – Historically, most systems have not used past data to predict crime patterns.

Lack of Real-Time Alerts – Conventional systems lack real-time notifications, lowering emergency response effectiveness.

Lack of Advanced Deep Learning Models – Most systems do not have sophisticated deep learning models for strong video analysis.

These shortcomings emphasize the necessity for an automated, AI-based system that can identify violence, forecast crime trends, and provide timely interventions.

1.5 User Requirement Analysis

CrimeGuard is designed to meet the following requirements:

❖ Functional Requirements:

Real-time video processing for violence detection.

Differentiation between violent and non-violent activities with high accuracy.

Module for crime prediction to study history and predict potential high-risk hotspots.

Automated alerts received by concerned authorities through email in the form of incident details, timestamp, and image evidence.

Visualization module rendering real-time trends of crime by interactive dashboards.

❖ Non-Functional Requirements:

Suitability with pre-existing CCTV and surveillance systems.

Scalability to support multiple live video streams at once.

Low latency to provide timely detection and alerts.

Secure handling of data in accordance with privacy laws.

❖ End-User Needs:

Smooth integration with existing surveillance infrastructure.

Easy-to-use interface with automated alerting.

Low false positives to maximize reliability.

1.6 Feasibility Study

Technical Feasibility:

CrimeGuard is built using CNN+BiLSTM for visual analysis and ML-based prediction models for crime forecasting. Flask framework ensures a lightweight and scalable deployment. SMTP-based email notifications provide real-time updates to security personnel.

Economic Feasibility:

Uses open-source tools (TensorFlow, OpenCV, Flask, Scikit-learn), minimizing costs. Leverages existing surveillance infrastructure, reducing deployment expenses.

Operational Feasibility:

User-friendly design, requiring minimal technical expertise for operation. Automated alerts and crime insights, enhancing law enforcement response efficiency.

Legal Feasibility:

Adheres to data privacy regulations, ensuring secure local processing of video feeds. Complies with ethical AI guidelines to prevent misuse of surveillance data.

Schedule Feasibility:

The project is feasible within the proposed timeline, as it uses readily available datasets and pre-trained models for accelerated development.

1.7 Objective of the Project

The primary aim of this project is to develop an intelligent and efficient system that can detect violent activities in real-time video feeds while also forecasting crime trends using historical data. The key objectives are:

- To design and implement a deep learning-based model capable of identifying violent actions from surveillance footage with high accuracy.

- To integrate a real-time video processing system that can generate immediate alerts upon detecting potential threats.

- To build a predictive model that analyzes past crime data to forecast the number of crimes likely to occur in the future using statistical techniques.

- To create a user-friendly interface that visualizes both real-time detections and crime predictions effectively.
- To ensure the system is scalable and adaptable for various public safety domains, including city surveillance, schools, and transport hubs.

1.8 Scope of the Project

This project focuses on addressing two major aspects of public safety: real-time detection of violence and prediction of crime patterns. The scope of this project includes:

- Processing live video input and detecting aggressive or violent behavior using a deep learning model.
- Utilizing historical crime datasets to train a predictive model capable of estimating future crime volumes based on location and time.
- Designing a web-based dashboard to present alerts and predictions in an intuitive manner for law enforcement or safety personnel.
- Ensuring minimal latency and high reliability in detection for potential real-world deployment.
- Supporting basic alert mechanisms (such as email notifications) upon detection.

However, the system does not currently:

- Identify specific individuals or recognize faces due to privacy and ethical concerns.
- Classify different types of crimes beyond the binary classification of violent vs. non-violent behavior.
- Function optimally in low-light or extremely crowded environments without additional enhancements.

1.9 Applications of the Project

The system developed in this project holds significant potential for enhancing safety and crime prevention efforts in various sectors. Key applications include:

- **City Surveillance Systems:** The project can be integrated into public CCTV networks to monitor streets, parks, and other public spaces, providing real-time alerts to police in case of violent incidents.

- **Transportation Hubs:** Implementation at bus terminals, railway stations, or airports can help in detecting fights or suspicious behavior, ensuring the safety of commuters.
- **Educational Institutions:** Schools and universities can use this system to monitor student behavior and promptly respond to potential threats within the campus.
- **Shopping Malls and Markets:** High-footfall areas prone to disputes can benefit from automated violence detection to assist security teams in faster response.
- **Smart Policing and Planning:** The predictive crime analysis component can assist law enforcement agencies in resource allocation and crime prevention strategies based on forecasted data.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review

Recent advancements in violence detection systems have demonstrated significant improvements in real-time monitoring, scalability, and computational efficiency. Convolutional Neural Networks (CNN) and convolutional Long Short-Term Memory (LSTM) models have been widely used to extract spatio-temporal features from surveillance videos, achieving high accuracy by analyzing frame-level differences to identify violent events [1][13]. Lightweight architectures such as MobileNetV2, ResNet, and MobileNet-TSM integrate spatial and temporal feature extraction, offering solutions that are computationally efficient and suitable for deployment in resource constrained environments [6][7][8][9]. Hybrid methodologies combining handcrafted features with deep transfer learning models like Xception and 2D CNNs have proven effective in classifying violent and non-violent behaviors, particularly in public datasets like HBD21 [4][11]. Advanced spatio-temporal frameworks, including 3D CNNs and Motion Saliency Maps (MSM) integrated with Temporal Squeeze-and-Excitation (T-SE) modules, outperform traditional models by providing state-of-the-art results on benchmark datasets such as Hockey Fight and Crowd Violence [3][5]. Deep learning models like ViolenceNet and multi-stream networks, including DenseNet, multi-head self-attention, and bidirectional LSTMs, enhance detection capability from interpersonal to person-to-person violence. Despite this, generalizing these models over diverse sets of datasets is still a challenge [10][12].

In addition, motion blob-based methods have been developed to emphasize computational

speed over accuracy, making them practical for real-time applications in high-risk settings like

prisons and psychiatric facilities [14].Augmenting it, CNN models equipped

of identifying items like knives and firearms have indicated promise in anticipating crime scenes with high precision, thereby providing reliable alerts and improving public safety interventions [15]. The results of these studies indicate the significance of the advancement in developing

intelligent surveillance systems capable of addressing real-world constraints while maintaining high accuracy and real-time responsiveness.

2.2 Research Gap

• Dataset Diversity: Existing records have particular information, and models cannot generalize to real-world violence due to it. Models perform fairly well on typical datasets but perform poorly in altered environments.

• Real-World Challenges: Models typically have trouble with diverse real-world scenarios like contrasting illumination, crowd densities, and motion patterns, which influence the accuracy and reliability of violence detection systems.

• False Positives: Excessive misclassification rates, particularly in intricate environments such as sports or dense areas, lead to large false positives, lowering the detection system's reliability.

* Small Dataset Size: Most models rely on small datasets, decreasing their generalizability and constraining their ability to function well in varied real-world scenarios.

* Safety System Integration: Few studies cover integration of violence detection models with real-time alert systems, which are crucial in preventing incidents and maintaining safety.

* Real-Time Constraints: Current models typically have difficulties with real-time frame processing, hence there is a need for low computational cost and high accuracy systems, needed in resource-limited environments. These limitations identify key dimensions that can be enhanced in developing practical and robust violence detection systems.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Methodology

To address the growing demand for crime detection in real time within surveillance systems, we adopted an organized approach that integrates deep learning methods with methodologies of application practicality. Our goal is to create an effective system that can proficiently detect violent activities from existing videos and live streams and provide timely intervention through produced notifications.The system also comprises visualization dashboard for the display of crime information in interactive form.

The suggested system combines Convolutional Neural Networks (CNNs) to extract spatial information from video frames and Long Short-Term Memory (LSTM) networks to capture time-based patterns between sequences of frames.This hybrid architecture allows the system to process video data , combining both spatial and temporal analysis. Furthermore, predictive crime analysis is incorporated using historical crime data and machine learning models to forecast crime-prone areas. The visualization dashboard presents real-time crime trends and model predictions using interactive dashboards. The following sections detail the key steps, including dataset preparation, model design, training, evaluation, real-time implementation, and visualization. This approach ensures that the system can be implemented in various real-world situations and can help assist in enhancing public safety.

Pseudocode

```
# Crime Data Preprocessing
LOAD 'crime_data.csv'
CLEAN missing or inconsistent values
NORMALIZE relevant features
SPLIT dataset into training (80%) and testing (20%) sets

# Crime Trend Forecasting (ARIMA)
FOR each state in dataset:
   FIT ARIMA model on historical crime data
   PREDICT future crime trends (next 5 years)

# Video Preprocessing for Violence Detection
LOAD input video
EXTRACT frames from video
RESIZE and NORMALIZE each frame
```

```
# CNN-LSTM Violence Detection
LOAD pretrained CNN model (e.g., MobileNetV2, VGG16, etc.)
FOR each frame:
   EXTRACT spatial features using CNN
   FEED features to LSTM for temporal analysis
   IF violence is detected:
      TRIGGER alert via email

# Flask Backend API
DEFINE POST route '/uploadvideo':
   RECEIVE video from request
   PROCESS frames and apply detection model
   RETURN prediction result (violence or not)

DEFINE POST route '/sendalert':
   IF triggered:
      SEND email notification to configured recipients
```

3.2 Dataset

The dataset includes total of 2,000 videos are split equally into two categories—violence and non-violence. There are 1,000 videos in the Non-Violence category showing actual events like sporting events, singing, vlogging, meals, and movie sequences. To help the model in differentiate between violent and non-violent human behaviours, these videos are varied and feature a variety of human behaviours. There are 1,000 videos in the Violence category showing various scenes of fight from movies, sports and the streets. In an effort to make sure that the model can generalise to various violent behaviour situations and environments, this directory seeks to record various forms of violent behaviour in real life. Diversity and balance in dataset assist in enhancing the training of the model so that it can distinguish between violent and non-violent behaviours with greater accuracy and less false positives. Besides, a crime database comprising past crime records is utilized for predictive analysis, which yields information about crime trends in terms of time and geographical area.

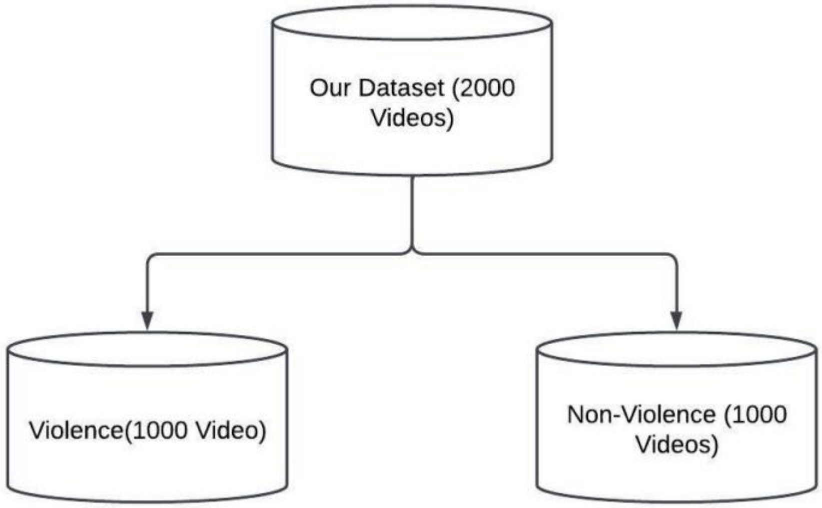


Figure 1: Dataset Split

3.3 Pre-processing

Pre-processing of the video dataset is a highly important activity in data preparation for training models. It begins with reading in input videos with OpenCV. Videos are splited into frames individually, which enables temporal patterns required to analyze activities. The frames are resized into a fixed size of (64 × 64) pixels for uniformity purpose and minimize computational complexity.Subsequently, the pixel values of each frame is normalized to maintain a consistent range, for faster and more stable model convergence. The frames are then grouped into eight consecutive batches of frames, flattened, and stored along with their corresponding one-hot encoded labels: [0, 1] for violence and [1, 0] for non-violence. For forecasting crime count using historic crime data, dataset is pre-processed by handling missing values, normalizing numerical features, feature extraction and encoding categorical variables. The visualization component processes the data to generate heatmaps and graphical representations of crime trends. This systematic pre-processing pipeline approaches ensures the dataset is organized and optimized for further training.

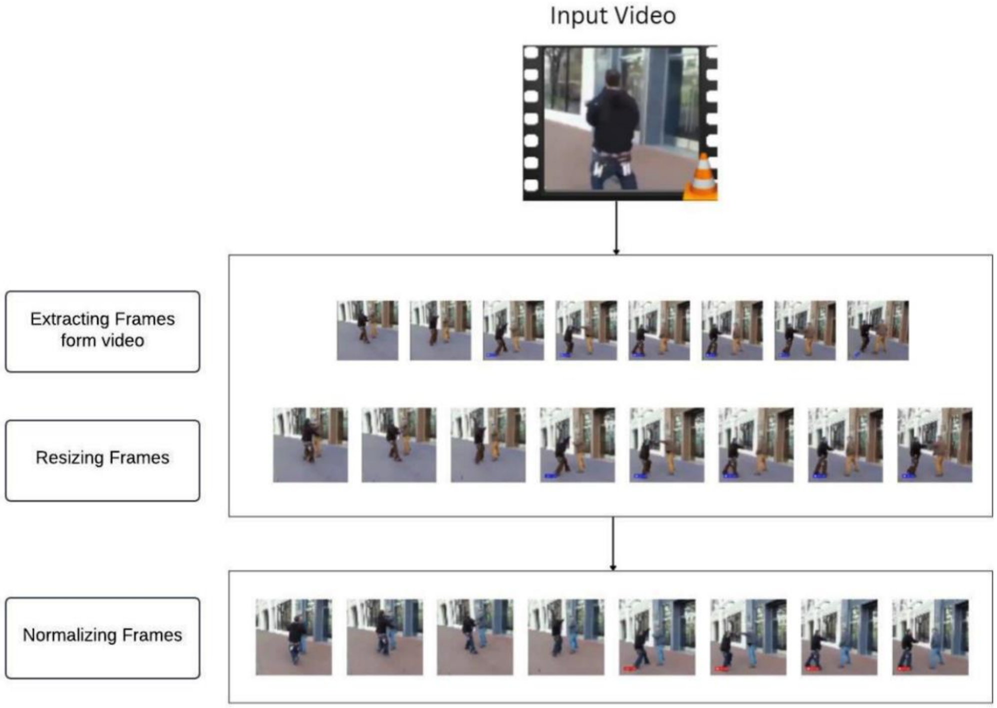


Figure 2: Preprocessing Pipeline

3.4 Dataset Split

The dataset is split into 80% training data and 20% validation data. The training set contains 25,862 frames and the validation set contains 6,466 frames. Each split contains an equal distribution of violent and non -violent videos. The crime dataset is also split similarly for training and evaluation of forecasting.

3.5 Model Architecture

The proposed crime detection system uses a hybrid architecture that combines Convolutional Neural Networks (CNN) for spatial feature extraction and Bidirectional Long Short-Term Memory (BiLSTM) networks for capturing sequential dependencies across each video frames. The input of the model consists of sequence of 8 frames, each resized to (64 × 64 × 3) pixels (height, width, and RGB channels). Therefore, the input shape is ((8, 64, 64, 3)). The CNN layers extracts spatial features by applying convolutions over these frames, making the model to learn important visual patterns such as movements, shapes, and textures. The output of these layers is made by a set of feature maps, which are then processed to capture temporal dependencies across the sequence of frames.

After the CNN layers, the feature maps are downsampled using MaxPooling3D layers to reduce both spatial and temporal dimensions, enabling the model to focus on the most important features and improve computational efficiency. The feature maps are reduced progressively through three pooling layers. These downsampled features are reshaped into a 2D tensor, preparing the data for sequential processing by the BiLSTM layer. The reshaped features are then passed into the BiLSTM layer, which captures the temporal dependencies between frames across the 8-frame sequence.

For crime prediction, a separate model is developed using machine learning techniques such as Random Forest and Gradient Boosting to analyze historical crime data and predict high-risk locations. The visualization module employs interactive graphs and dashboards to display real-time detections and predictive crime patterns.

3.6 Hyperparameter Tuning

ReLU activation is used in convolutional and dense layers, while the Sigmoid activation function is used for the output layer. The optimizer chosen is SGD, and categorical cross-entropy is used as the loss function. Grid search is applied to tune hyperparameters in the predictive model for better accuracy.

3.7 Model Training

The proposed system is trained using the fit() function provided by the TensorFlow Keras API. The training process utilizes a batch size of 8 and runs for a total of 12 epochs. To enhance generalization and avoid overfitting, an **early stopping** mechanism is incorporated. This technique monitors the validation loss during training and halts the process if the model's performance ceases to improve after a defined number of epochs.

The architecture combines a **Convolutional Neural Network (CNN)** with a **Bidirectional Long Short-Term Memory (BiLSTM)** network. The CNN component is responsible for extracting spatial features from individual frames, while the BiLSTM captures temporal dependencies across sequences of frames. This fusion enables the model to effectively recognize patterns of violence over time in video streams, making it well-suited for real-time surveillance scenarios.

In addition to the deep learning-based video classification module, a **time series forecasting model** is also employed for crime trend analysis. The **AutoRegressive Integrated Moving Average (ARIMA)** model is used to predict the future count of criminal activities based on past records. The ARIMA model is trained on historical crime data to learn temporal patterns and generate accurate forecasts, which can assist law enforcement in strategic planning and resource allocation.

3.8 Model Prediction and Visualization

After the training phase, the system is transitioned into inference mode using the predict() function. The trained CNN-BiLSTM model is capable of classifying sequences of video frames in real-time to detect instances of violence. Video input is continuously streamed to the system, where each frame undergoes preprocessing and is passed through the network. If the model detects signs of violence with a confidence above a predefined threshold, an **immediate alert is triggered**, notifying the concerned authorities or system administrators.

Parallelly, the ARIMA-based predictive model is used to estimate the number of crimes that are likely to occur in upcoming days, weeks, or months. This predictive insight is particularly valuable for crime prevention and proactive decision-making.

The system also integrates a **comprehensive visualization dashboard** that enhances interpretability and user engagement. This includes:

- **Real-time Line Graphs:** Displaying the temporal fluctuation in detected crime rates.
- **Heatmaps:** Illustrating spatial concentration of criminal activity over different geographic locations.
- **Trend Charts and Bar Plots:** Depicting the distribution of crime types over time or regions.
- **Interactive Tools:** Allowing users to filter data based on location, date range, or crime category, thereby making the interface more dynamic and user-centric.

These visualizations help stakeholders—such as police departments, analysts, and policy-makers—gain actionable insights from both real-time video feeds and historical datasets. The overall system thus functions not only as a detection tool but also as a predictive and analytical platform for enhancing public safety.

3.9 Tech Stack

❖ Programming Languages: Python 3, JavaScript, HTML, CSS

❖ Backend Frameworks: Flask

❖ Frontend Frameworks: React

❖ IDEs: Google Colab, Jupyter Notebook, Visual Studio Code

❖ Machine Learning Tools: TensorFlow/Keras, OpenCV, Scikit-learn

❖ Email Service: SMTP

❖ Visualization Tools: Matplotlib, Seaborn, Plotly, Dash

Used tech stack efficiently handles vedio processing and model training to real-time violence detection, crime forecasting, and interactive crime data visualization, ensuring user-friendly experience for crime monitoring and prevention.

3.10 Use Case Model

The Use Case Model provides a high-level view of the interaction between the user and the system. It identifies the primary actors and the key functionalities of the system. For our violence detection project, the system enables users to upload video files, processes them using deep learning models, and returns predictions indicating whether violence is detected.

Actors:

- User: The individual interacting with the system by uploading video files.
- Model: The backend framework responsible for processing videos and generating predictions.

Use Cases:

1. Upload Video

The user uploads a video through the web interface. This video can be either a recorded file or a real-time camera feed.

2. Preprocess Video

Once uploaded, the system extracts frames from the video, resizes them, and normalizes pixel values. This prepares the input for the deep learning model.

3. Predict Violence / Non-Violence

The preprocessed frames are passed through a trained CNN+LSTM or transfer learning model (e.g., MobileNetV2, VGG16) which classifies each video as either violent or non-violent.

4. Send Notification (if violence is detected)

If violence is detected, the system sends an alert via email to predefined recipients.

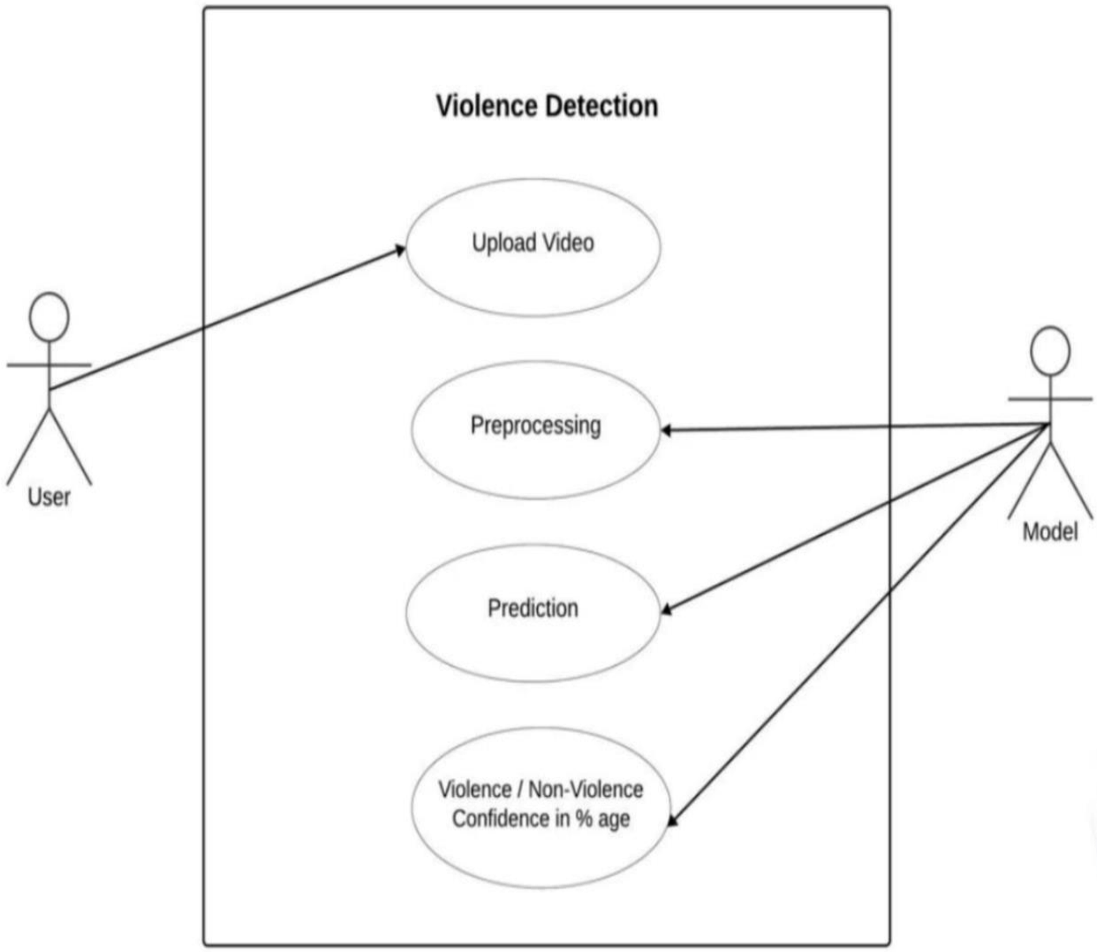


Figure 3: Use case diagram

3.11 Training and Testing

The Training Workflow represents the complete pipeline used to develop the violence detection model, from data collection and preprocessing to training, evaluation, and deployment. This structured approach ensures the model is optimized for both accuracy and real-time performance.

Workflow Steps:

1. Dataset Collection
   - A dataset consisting of 2000 labeled videos (1000 violence, 1000 non-violence) is used.

2. Preprocessing
   - Each video is processed frame-by-frame.
   - Frames are resized to a standard input size and normalized for pixel value scaling.
   - Frames are organized into sequences to retain temporal information.

3. Processed Dataset
   - The dataset is split into training and testing sets (e.g., 80% train, 20% test).
   - A DataLoader is created to feed batches of frame sequences into the model efficiently.

4. Model Training (CNN + LSTM)
   - A hybrid architecture combining Convolutional Neural Networks (CNN) for spatial feature extraction and LSTM for temporal sequence learning is trained.
   - Training continues for multiple epochs until convergence.

5. Model Evaluation
   - The model is evaluated using a confusion matrix, providing metrics such as accuracy, precision, recall, and F1-score.
   - This ensures robust classification of violent vs non-violent content.

6. Model Export
   - The trained model is saved for later use in inference on real-time data.

7. Deployment and Real-time Inference
   - The exported model is integrated with a live camera or video feed.
   - Incoming videos are processed similarly and passed to the trained model.

8. Based on the output:
   - If non-violence is detected, no action is taken.
   - If violence is detected, an alert email is automatically triggered to notify concerned authorities.

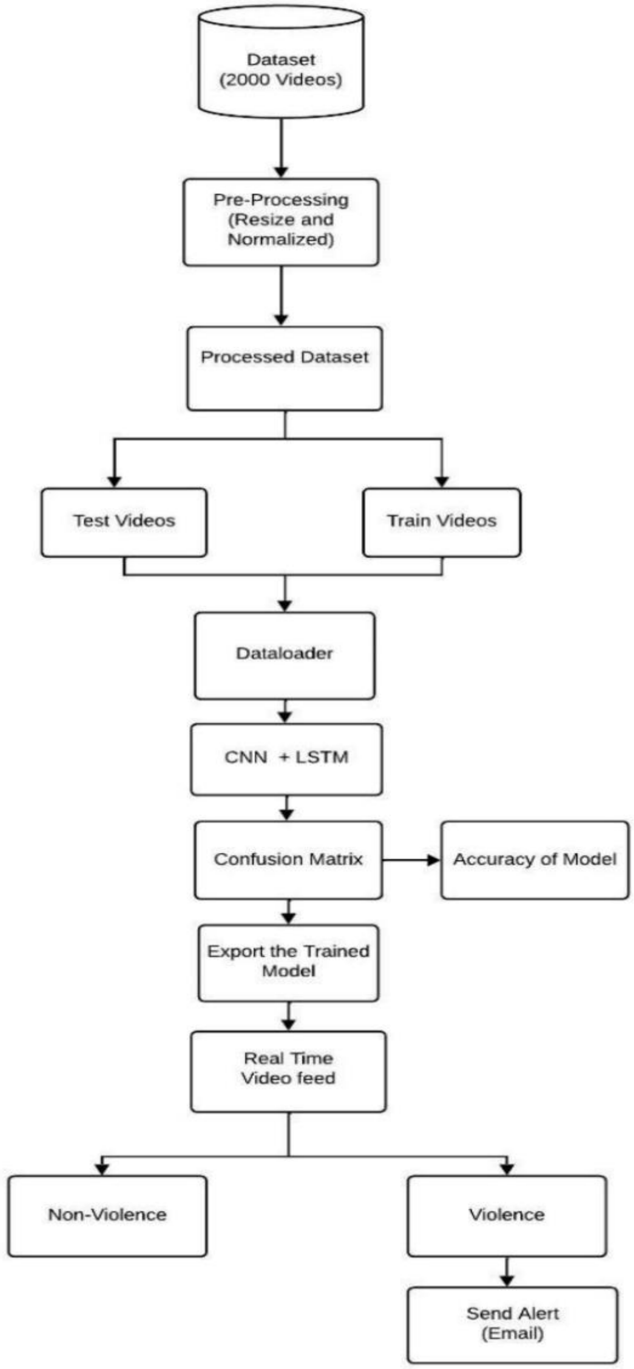


Figure 4: Training Workflow

The Testing Workflow outlines the step-by-step process of handling new or unseen videos during the testing or deployment phase. This workflow leverages the trained CNN+LSTM model to classify videos in real-time or batch mode as violent or non-violent.

Workflow Steps:

1. User Uploads Video
   - The user provides a new video via the web interface or live camera feed.

2. Preprocessing
   - The uploaded video is broken down into frames.
   - Frames are resized and normalized, ensuring they match the format used during training.

3. Load Trained Model
   - The previously exported CNN+LSTM model is loaded into memory.

4. Prediction
   - The preprocessed frames are fed into the model.
   - The model outputs a classification: Violence or Non-Violence.

5. Alert System (If Violence Detected)
   - If the output is "Violence", the system automatically triggers an email alert to notify authorities or administrators.
   - If "Non-Violence" is detected, the result is simply logged or displayed without sending any alerts.

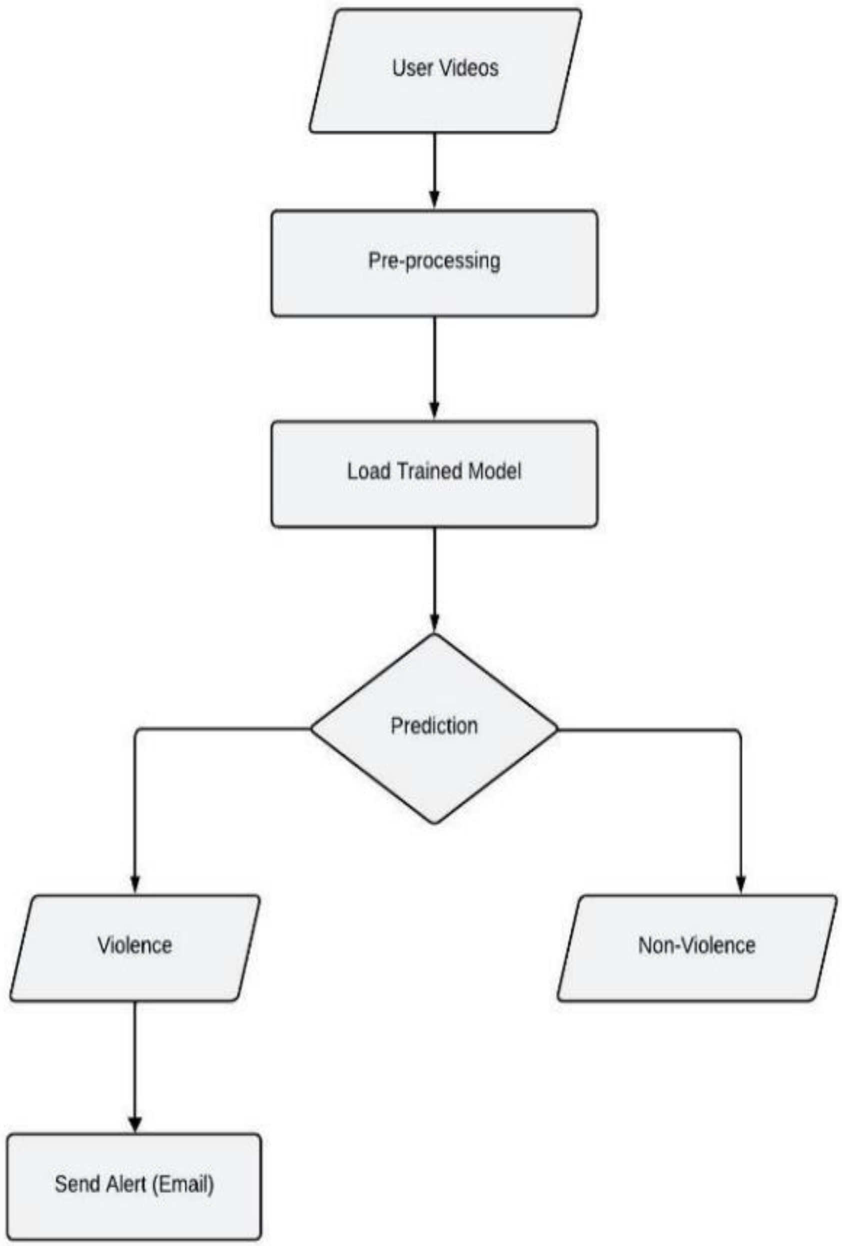


Figure 5: Testing Workflow

3.12 Sequence Diagram

Figure 6 illustrates the Sequence Diagram representing the step-by-step interaction between different components of the violence detection system. It outlines the flow of video data from the user to the final classification output generated by the deep learning model.

Actors and Components:

- User: Initiates the process by uploading a video through the interface.
- System: Acts as the main controller that handles communication between modules.
- Preprocessing: Responsible for extracting frames, resizing them, and normalizing pixel values to prepare data for the model.
- CNN (Convolutional Neural Network): Extracts spatial features from video frames.
- LSTM (Long Short-Term Memory Network): Analyzes the temporal patterns from sequences of feature vectors to classify the video.

Flow Description:

1. The User uploads a video to the System.
2. The System forwards the video to the Preprocessing module.
3. Preprocessing extracts individual frames, resizes and normalizes them, then sends them to the CNN.
4. The CNN processes the frames and generates feature vectors.
5. These feature vectors are passed to the LSTM model, which performs temporal analysis.
6. The LSTM outputs a classification result—Violence or Non-Violence.
7. The result is returned to the System, which then communicates the violence percentage or classification output back to the User.

This diagram captures the sequential collaboration of components required for real-time video analysis and showcases how deep learning modules are integrated into the processing pipeline.

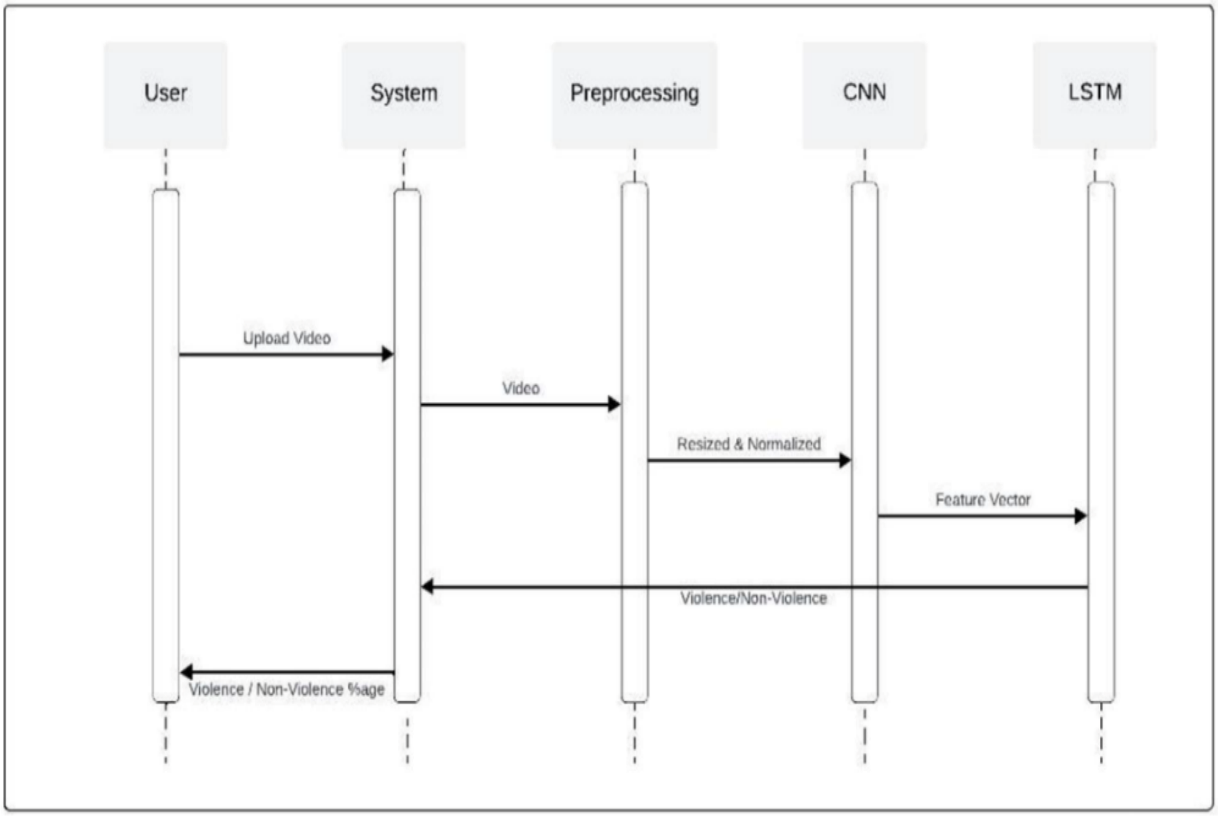


Figure 6: Sequence Diagram

3.13 System Architecture

System Architecture illustrates the complete system architecture of the real-time violence detection framework developed for processing and analyzing video data using deep learning techniques. The architecture integrates multiple components that collectively manage the end-to-end pipeline—right from accepting raw video input to generating a prediction and triggering alerts when violence is detected. The system is designed with scalability and modularity in mind, ensuring that it can support both offline video uploads and real-time camera feeds.

The architecture is divided into two major flows:

- Training Flow (indicated by solid black arrows)
- Prediction Flow (indicated by red dashed arrows)

1. Dataset Acquisition and Preprocessing

At the foundation of the system lies a labeled video dataset, which includes 2,000 video samples—1,000 labeled as Violence and 1,000 labeled as Non-Violence. This dataset is crucial for training the deep learning model.

The raw videos are first passed through the Preprocessing module, which comprises the following key steps:

- Frame Extraction: Each video is decomposed into individual frames. This transforms the video from a temporal sequence into a series of images suitable for frame-level processing.
- Frame Resizing and Normalization: The extracted frames are resized to a standard dimension (e.g., 224x224 pixels) and normalized to scale the pixel values between 0 and 1. This ensures uniformity in the input data and helps improve the learning efficiency of the model.
- Frame Queuing: Frames are organized in a structured queue that enables smooth loading into the training pipeline and ensures temporal continuity in video sequences.

The output of the preprocessing phase is a processed dataset containing resized and normalized frames ready for use in both training and prediction flows.

2. Model Training Pipeline

After preprocessing, the processed dataset undergoes a Train/Test split, separating the data into training and evaluation sets. Typically, an 80:20 split is used to ensure that the model is evaluated on unseen data.

The Data Loader module is responsible for efficiently loading these datasets into the model, batching the data, and ensuring that both frames and corresponding labels (violence or non-violence) are properly aligned.

The core of the system is the Violence Detection Model, which is a hybrid deep learning architecture composed of:

- CNN (Convolutional Neural Network): Performs spatial feature extraction by analyzing each frame independently. It detects patterns such as movement, posture, and background context.

- LSTM (Long Short-Term Memory Network): Takes the feature vectors output by the CNN and performs temporal analysis. This is critical for understanding how actions evolve over time across a sequence of frames, enabling the system to distinguish between violent and non-violent events.

Once the model achieves satisfactory performance based on training metrics like accuracy, precision, recall, and the confusion matrix, it is saved for later use. This Export Trained Model step finalizes the training phase and produces a serialized version of the trained CNN+LSTM model.

3. Real-Time Prediction Pipeline

In the real-time or prediction phase, a user interacts with the system by uploading a video, which can be a pre-recorded clip or a live feed from a camera. This input video is immediately processed by the same Preprocessing module, following identical steps as those used in the training pipeline to ensure consistency.

The system then proceeds to the Load Trained Model step, where the previously exported CNN+LSTM model is loaded into memory. The preprocessed frames are fed into the model for inference.

The model predicts whether the given video contains violent activity or not:

- If the output is classified as Non-Violence, the result is returned to the user interface.

- If the output is classified as Violence, the system proceeds to the next stage: Alert Generation.

4. Alert System and Live Integration

Upon detecting violence in a video, the system is designed to automatically trigger an email alert to a designated recipient (e.g., security personnel, law enforcement, or a predefined contact group). This feature enhances the practical utility of the system in real-world surveillance applications.

Additionally, the system supports real-time video feed processing, making it suitable for deployment in environments such as public places, schools, hospitals, and transport hubs. The system continuously monitors incoming streams and applies the trained model in real time, classifying the content and issuing alerts where necessary.

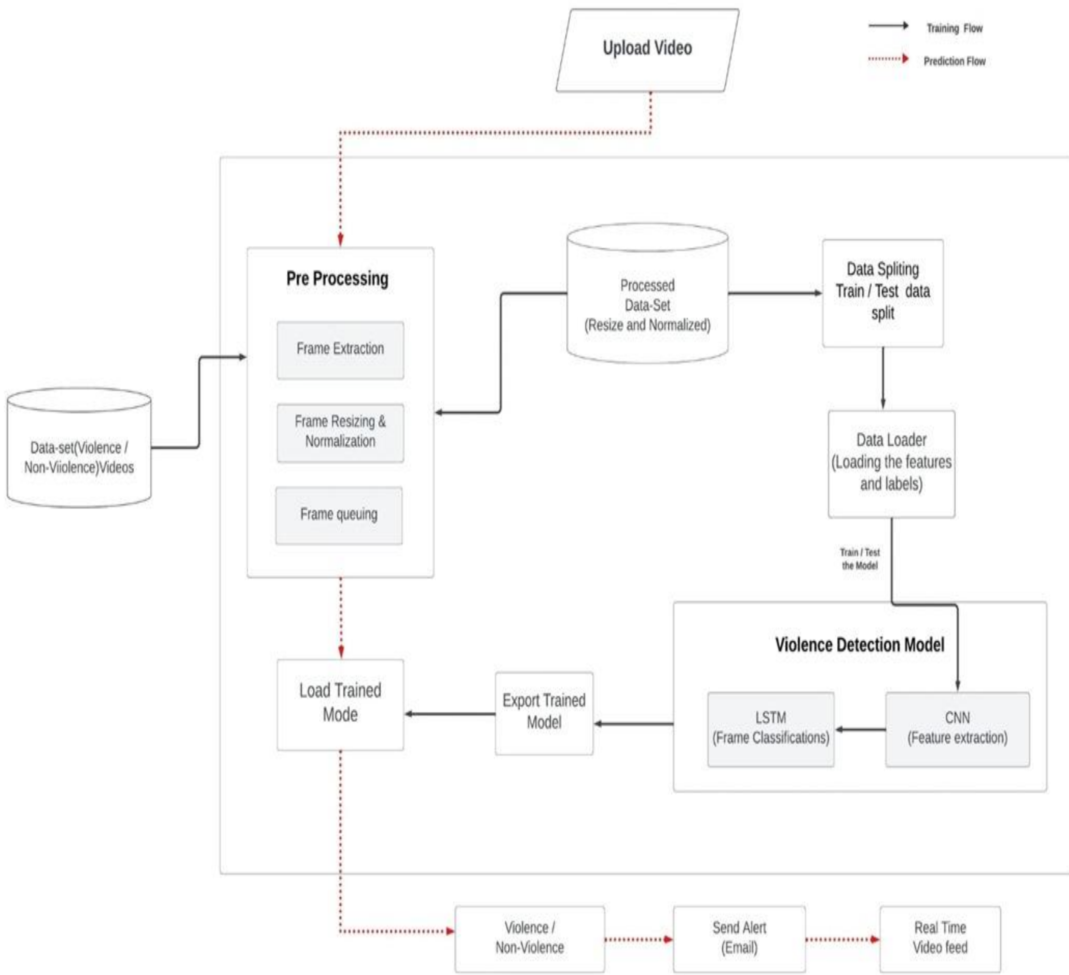


Figure 7: System Architecture

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Result

The CrimeGuard system was evaluated on a dataset containing a group of clips of violent activities and historical crime records. The model's performance was evaluated using 5 key factors that are accuracy, precision, recall, F1-score, and computational efficiency. The dataset included various video samples covering various violent and non-violent scenarios to ensure testing was strong. All the computations were performed in Python using libraries like TensorFlow, OpenCV, and Scikitlearn.

A. Metrics Definition

To assess the effectiveness of the models, the following evaluation metrics were used:

- Accuracy: Measures the overall correctness of the model
- Precision: Represents the proportion of correctly predicted positive instances among all predicted positives:
- Recall: Measures the proportion of actual positive instances correctly identified.
- F1-score: Provides a harmonic mean of precision and Recall.

Here, T P , T N , F P , and F N denote the true positives, true negatives, false positives, and false negatives, respectively.

B. Performance Comparison

The result reports of different deep learning models used in for detecting violence are summarized in Table 1.1.

Table 1.1 : Performance metrics for violence detection models

Model	Accuracy(%)	Precision	Recall	F1-Score
VGG16+LSTM	89.0	0.89	0.89	0.89
VGG19+LSTM	85.0	0.85	0.85	0.85
ResNet50+LSTM	60.0	0.61	0.60	0.60
MobileNetV2+LSTM	93.0	0.93	0.93	0.93
CNN+LSTM	96.0	0.95	0.96	0.96

4.2 Observations

- The CNN+LSTM model demonstrated the highest accuracy at 96.0%, surpassing all other models in terms of precision, recall, and F1-score. It emerged as the most effective model for the task of violence detection.

- The MobileNetV2+LSTM model also showed strong performance with an accuracy of 93.0%. Its lightweight architecture makes it particularly suitable for real-time deployment scenarios where efficiency is crucial.

- Other hybrid CNN-LSTM models, such as VGG16+LSTM and VGG19+LSTM, delivered decent results with accuracies of 89.0% and 85.0%, respectively. These models strike a balance between depth and performance but are comparatively less effective than CNN+LSTM.

- The ResNet50+LSTM model yielded a lower accuracy of 60.0%, suggesting possible issues such as overfitting or suboptimal feature representation in the context of violence detection.

- The training loss consistently decreased over time, indicating that the models were optimized effectively during training.

- CNN+LSTM, while having the highest training duration (2520 seconds), offered the fastest inference time (0.014 seconds), making it highly suitable for real-time violence detection systems.

- MobileNetV2+LSTM provided a balanced approach by maintaining a good accuracy level (93.0%) and achieving a relatively low training time of 1025 seconds, making it an efficient choice for time-sensitive applications.

- ResNet50+LSTM exhibited a significantly high inference time of 0.143 seconds, which limits its practicality for real-time processing.

4.3 Conclusion

The test results confirm that CNN+LSTM is the best model for real-time violence detection, with the highest accuracy and balanced performance on all metrics. Although MobileNet-V2 and VGG16 are not far behind, ResNet50 falls behind because of its inability to process sequential frames. The analysis of computational complexity guarantees that CrimeGuard remains fast and efficient for practical use. In the future, efforts will be directed towards making these tech configurations faster and even more accurate.

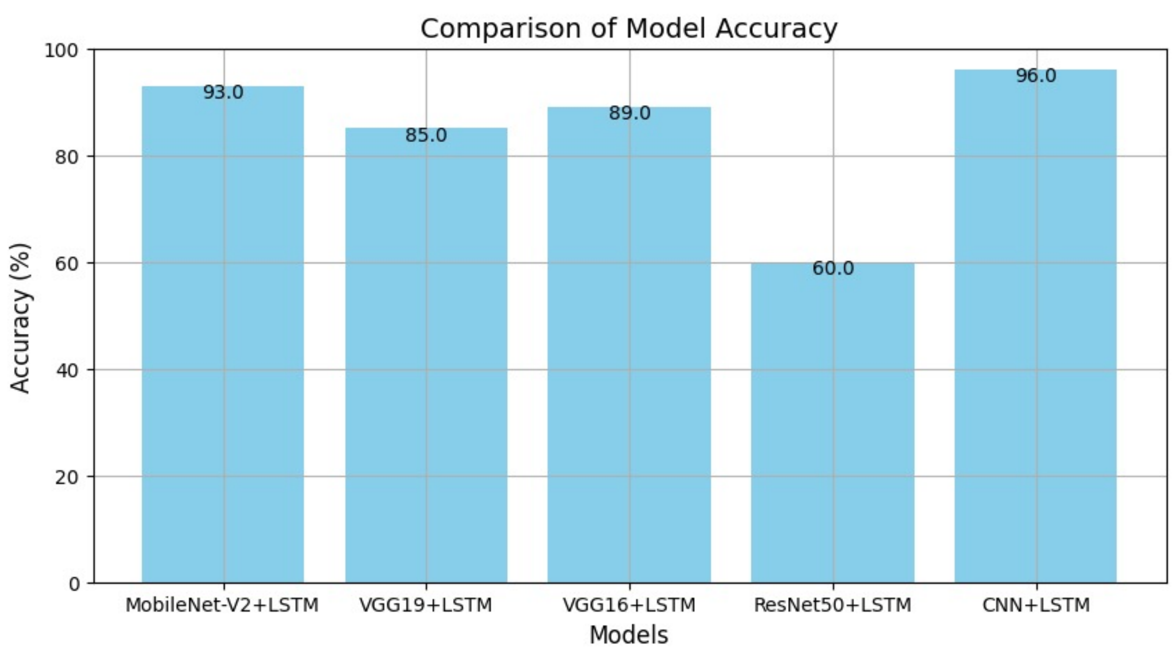


Figure 8: Model Accuracy

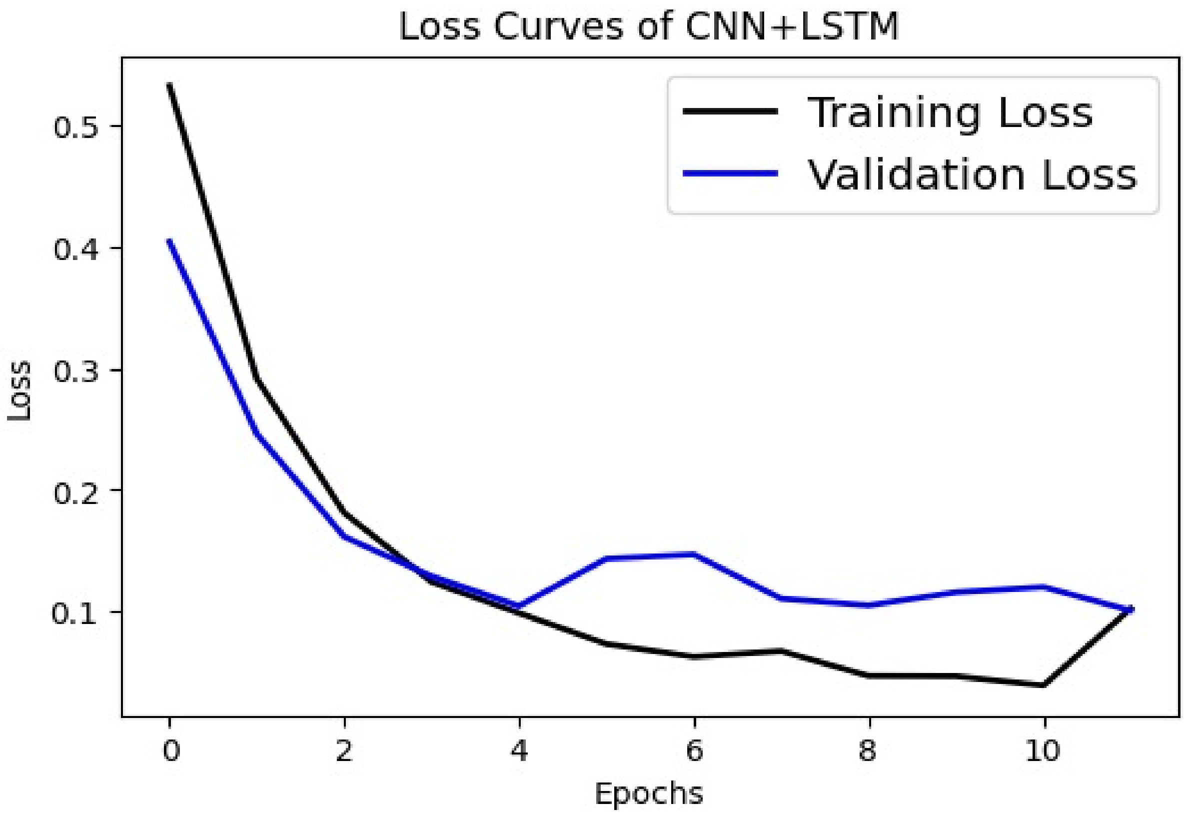


Figure 9: Loss Curves of CNN+LSTM

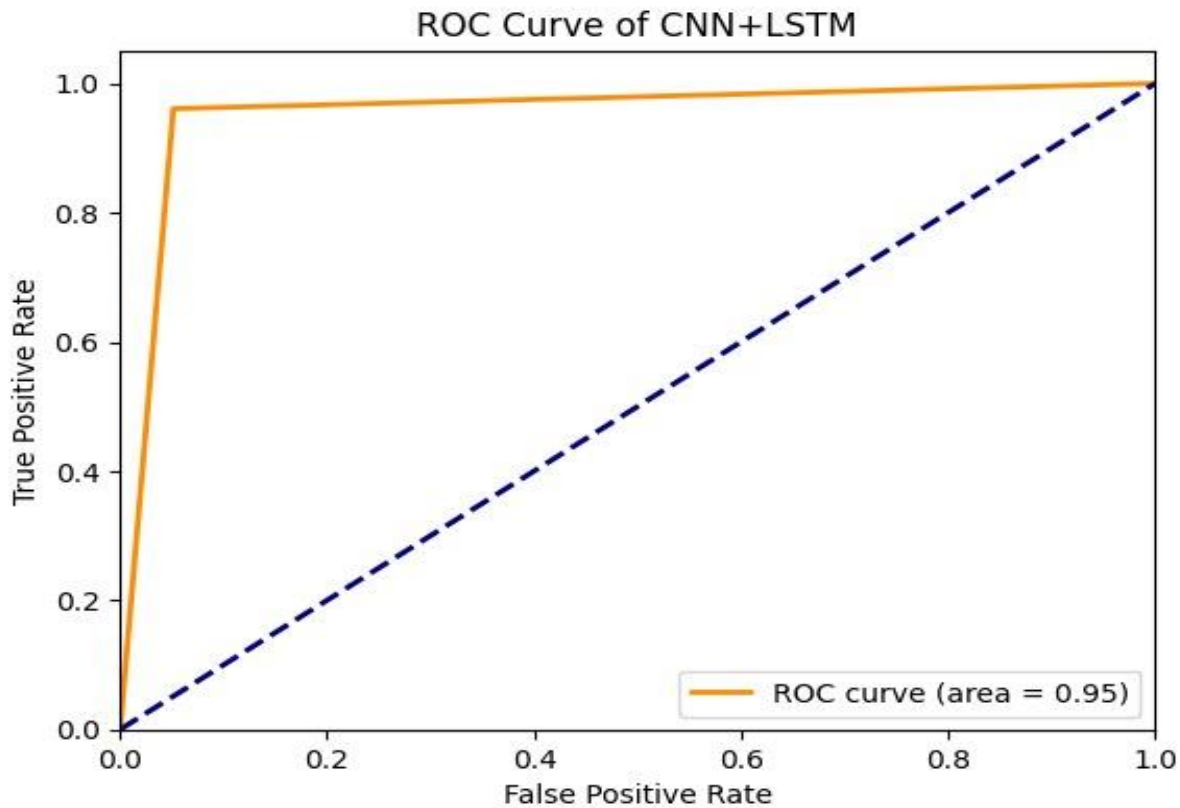


Figure 10 :ROC Curve

4.4 Output

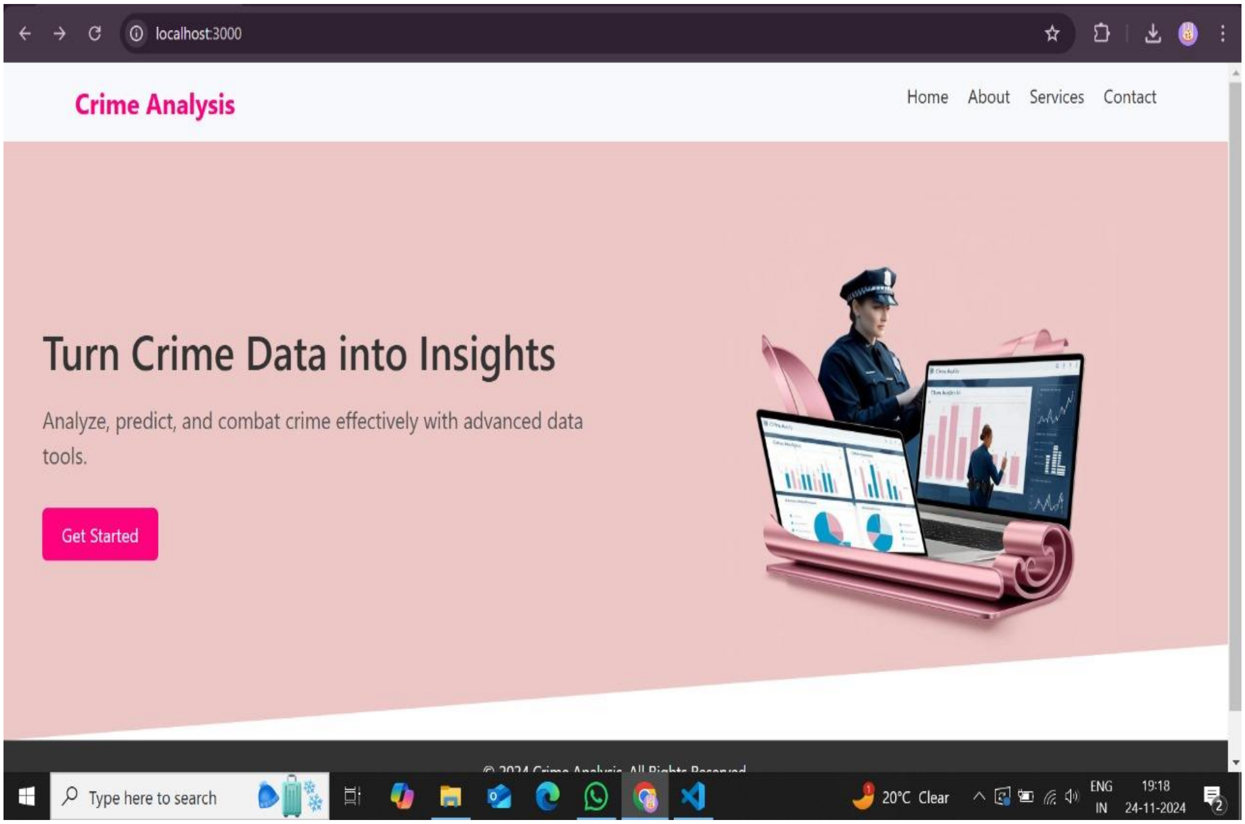


Figure 11: Home Page

Figure 12: Home Page-2

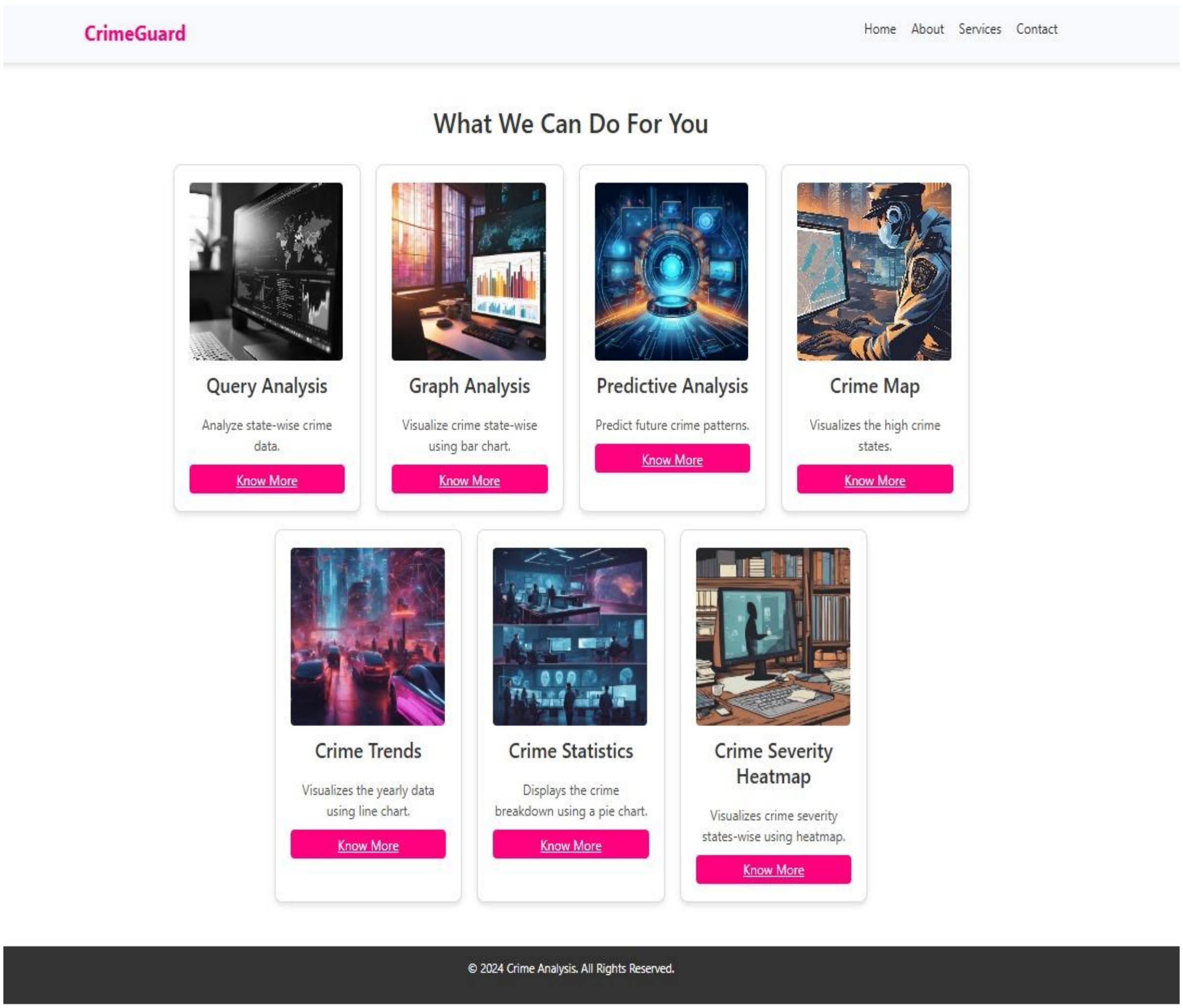


Figure 13: Our Services

CHAPTER 5

CONCLUSIONS AND FUTURE SCOPE

5.1 Conclusion

CrimeGuard is a groundbreaking crime tracking and analysis system that enhances public safety by detecting violence in real-time and predicting crimes. It applies Convolutional Neural Networks (CNNs) for pattern discovery and Long Short-Term Memory (LSTM) networks for continuous data analysis. It identifies violent activities effectively and maps high risk crime locations based on historical trends. With the alert system and proactive law enforcement measures, CrimeGuard enables the authorities to deal with threats efficiently, ensuring public safety through wise decision-making.

5.2 Future Scope

The CrimeGuard system, while effective in its current form, holds significant potential for further development and real-world scalability. Future enhancements can not only improve detection and prediction accuracy but also broaden the system's applicability across diverse environments and use cases. The following key areas outline the planned directions for future research and implementation:

• Expansion of Data Sources

To enhance the depth and breadth of crime pattern analysis, future iterations of the system aim to incorporate a wider range of data sources. This includes integrating information from social media platforms, local news feeds, emergency response databases, and community reports. By leveraging these diverse and real-time data streams, the system can develop a more comprehensive understanding of crime events and public sentiment, enabling faster and more context-aware predictions.

• Integration of Reinforcement Learning

To adapt dynamically to changing crime trends and environmental conditions, reinforcement learning (RL) can be incorporated. Unlike static models, RL allows the system to learn from continuous feedback in real time, improving its decision-making ability over time. This advancement can significantly enhance the adaptability of the system in detecting evolving violence patterns and responding to novel situations.

• Incorporation of Socio-Economic Factors

Crime patterns are often influenced by various socio-economic factors such as population density, unemployment rates, income disparity, and education levels. Future versions of the predictive model will integrate such demographic and economic variables to provide more context-aware forecasting. This holistic approach can improve the accuracy of crime rate predictions by correlating trends with root causes and societal dynamics.

• Algorithm Optimization and Cloud Integration

To ensure that the system remains scalable and capable of handling large-scale data streams, there is a need for optimizing the underlying algorithms. Techniques like model pruning, quantization, and lightweight architectures will be explored. Additionally, cloud-based integration will be implemented to support distributed processing, real-time updates, and seamless access from multiple locations. This transition to cloud infrastructure will also facilitate collaboration between law enforcement agencies and public safety systems.

• Expansion of Dataset Diversity

To improve the model's generalizability and robustness, it is essential to train it on more diverse datasets. Future work will focus on curating and including video samples from different countries, cultures, lighting conditions, and types of violent incidents. This diversity will enhance the model’s ability to accurately detect and classify a wide variety of violent behaviors in heterogeneous real-world settings.

• Deployment on Edge Devices

For real-time crime detection, especially in remote or high-risk areas with limited connectivity, deploying the system on edge devices such as CCTV units, surveillance drones, or smart helmets is crucial. Edge deployment ensures low-latency processing and immediate response without the need to send video streams to the cloud, thereby improving system reliability and speed while maintaining privacy and reducing bandwidth usage.

5.3 Challenges Faced

During the development and implementation of the proposed CrimeGuard system, several technical and operational challenges were encountered. These challenges spanned across data collection, model design, training, and system deployment. Below is a summary of the key issues faced:

1. Data Collection and Labeling

One of the primary challenges was acquiring high-quality video datasets that accurately represent real-world violent and non-violent activities. Publicly available datasets were often limited in size or lacked diversity in terms of lighting, background noise, camera angles, and types of violence. Additionally, labeling video sequences with frame-level accuracy was a time-consuming and labor-intensive task, requiring human annotation to ensure consistency and correctness.

2. Model Complexity and Training Time

Combining CNN with BiLSTM introduced significant computational complexity. Training such a hybrid model required substantial processing power and memory, especially when working with large volumes of frame sequences. Ensuring the model learned both spatial and temporal patterns without overfitting required careful tuning of hyperparameters, regularization techniques, and the use of early stopping.

3. Handling Real-Time Video Input

Processing live video streams in real time posed additional difficulties. Maintaining a balance between inference speed and accuracy was crucial, as any delay in detecting violent activities could render the system ineffective. Optimizing the model for real-time performance, especially on hardware with limited GPU resources, involved streamlining preprocessing steps and batch management.

4. Noise and False Positives

In real-world surveillance videos, varying environmental conditions—such as low light, occlusion, and crowd density—often introduced noise that affected the model's performance. This occasionally led to false positives or missed detections. Reducing such errors required refining the training dataset and enhancing the robustness of the model through data augmentation techniques.

5. ARIMA Forecasting Limitations

The ARIMA model, though effective for univariate time series forecasting, exhibited limitations when crime patterns were influenced by external factors such as public events, festivals, or law enforcement interventions. Incorporating such contextual information into the forecasting model remains a complex task and suggests the need for future integration of multivariate models.

6. Visualization and Integration

Integrating real-time model outputs with an interactive visualization dashboard required seamless backend and frontend coordination. Displaying large volumes of dynamic data while ensuring usability, responsiveness, and scalability presented additional design and implementation challenges.

REFERENCES

[1] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, “Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes,” Computer Vision and Image Understanding, vol. 172, pp. 88-97, 2018.

[2] B. Sathyadevan and S. Gangadharan, “Crime analysis and prediction using data mining techniques,” 2014 First International Conference on Networks & Soft Computing (ICNSC), 2014, pp. 406-412.

[3] M. Ullah, M. A. Muhammad, G. H. Lee, and S. W. Baik, “Violence detection using spatiotemporal features with 3D convolutional neural networks,” Sensors, vol. 19, no. 11, p. 2472, 2019.

[4] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (XAI),” IEEE Access, vol. 6, pp. 52138-52160, 2018.

[5] X. Xu, L. Yang, H. Wang, and F. Zhao, “Edge computing: A new computing model for Internet of Things,” IEEE Internet of Things Journal, vol. 7, no. 3, pp. 1778-1786, 2020.

[6] N. K. Sharma, M. Pandey, and R. K. Gupta, “IoT-based smart surveillance systems for crime detection and prevention,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 8473-8490, 2021.

[7] G. Gorr and R. Harries, “Introduction to crime forecasting,” International Journal of Forecasting, vol. 19, no. 4, pp. 551-555, 2003.

APPENDIX

Appendix A: Performance Metrics Overview

The following table summarizes the performance metrics for all deep learning models evaluated during the development of CrimeGuard.

Table A1: Performance Metrics for Violence Detection Models

Model	Accuracy(%)	Precision	Recall	F1-Score
VGG16+LSTM	89.0	0.89	0.89	0.89
VGG19+LSTM	85.0	0.85	0.85	0.85
ResNet50+LSTM	60.0	0.61	0.60	0.60
MobileNetV2+LSTM	93.0	0.93	0.93	0.93
CNN+LSTM	96.0	0.95	0.96	0.96

Appendix B: Visualizations

Model Accuracy Comparison
A bar graph comparing the accuracy of different violence detection models.

CNN+LSTM Loss Curves
Training and validation loss plotted across epochs for the CNN+LSTM model.

ROC Curve for CNN+LSTM
Receiver Operating Characteristic curve showing the true positive vs. false positive rates.

Appendix C: User Interface Outputs

CrimeGuard Home Page
A screenshot of the web interface's home page, allowing video uploads and model selection.

Appendix D: System Environment and Configuration

Component	Details
Programming Language	Python 3.11
Libraries & Frameworks	TensorFlow, OpenCV, Scikit-learn, NumPy, Flask, ReactJS
IDE / Tools Used	VS Code, Jupyter Notebook
Operating System	macOS / Linux
Hardware Specification	8-core CPU, 16GB RAM
Dataset Size	2000 videos (1000 violence, 1000 non-violence)
Training Time (CNN+LSTM)	~2 hours (on CPU); faster with GPU

Appendix E: Evaluation Metric Definitions

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

Where:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

Appendix F: Observations Summary

CNN+LSTM achieved the highest accuracy (96%) and F1-score (0.96), confirming its robustness in detecting violent actions.

MobileNetV2+LSTM also performed well with 93% accuracy.

VGG16 and VGG19 achieved moderate performance with accuracies of 89% and 85% respectively.

ResNet50+LSTM underperformed with only 60% accuracy, likely due to its limited capability to model temporal dependencies in video data.

Appendix G: Project Repository and Structure

Folder Structure:
CrimeGuard/

```
│
├── frontend/          # React.js Frontend
│   └── src/
│       └── components/
│
├── backend/           # Flask Backend
│   ├── app.py
│   └── model/
│       └── violence_detection_model.h5
│
├── dataset/
│   ├── violence/
│   └── non-violence/
│
├── evaluation/        # Metrics, graphs, and reports
│   ├── model_accuracy.png
│   └── loss_curves.png
│
```

CrimeGuard: Real-time Violence Detection and Predictive Analysis System

1st Shreya Goswami
Computer Science Engineering (AIML)
KIET Group of Institutions, Delhi-NCR
Ghaziabad, UP, India
shreya.2125csme1036@kiet.edu

2nd Khushi Bansal
Computer Science Engineering (AIML)
KIET Group of Institutions, Delhi-NCR
Ghaziabad, UP, India
khushi.2125csme1013@kiet.edu

3rd Mahi Tyagi
Computer Science Engineering (AIML)
KIET Group of Institutions, Delhi-NCR
Ghaziabad, UP, India
mahityagi1222@gmail.com

4th Rajeev Kumar Singh
Associate Professor
Computer Science Engineering (AIML)
KIET Group of Institutions, Delhi-NCR
Ghaziabad, UP, India
rajeev.kumar.csai@kiet.edu

5th Dr Puneet Garg
Associate Professor
Computer Science Engineering (AI)
KIET Group of Institutions, Delhi-NCR
Ghaziabad, UP, India
puneetgarg.er@gmail.com

***Abstract*—This research presents CrimeGuard, a real-time violence detection system designed to enhance public safety by automating video surveillance analysis. The system processes live and recorded video feeds to classify activities as either violent or non-violent using deep learning architectures, including CNN+LSTM, MobileNetV2+LSTM, VGG16+LSTM, VGG19+LSTM, and ResNet50+LSTM. Among these models, CNN+LSTM achieved the highest accuracy of 96%. The backend, built using Flask, facilitates video processing and automated email alerts, while a React-based frontend ensures seamless user interaction. Additionally, the system integrates predictive modeling and visualization techniques to assist law enforcement in data-driven decision-making. The results demonstrate CrimeGuard's effectiveness in real-time crime monitoring, providing proactive alerts to aid law enforcement and security agencies.**

***Index Terms*—Violence Detection, Crime Analysis, Deep Learning, Predictive Modeling, Video Surveillance, Automated Alerts, Crime Prevention**

I. Introduction

Crime and violence remain persistent challenges, demanding real-time monitoring and intelligent analysis to enhance public safety. Traditional surveillance systems rely heavily on manual intervention, which is inefficient in preventing violent incidents. Similarly, crime data is often underutilized, limiting law enforcement agencies' ability to detect trends and prevent future crimes.

To address these issues, **CrimeGuard** integrates real-time violence detection from video feeds and crime pattern analysis using machine learning and deep learning techniques. The system processes live and recorded videos to identify violent activities using **CNN**, **MobileNetV2**, **VGG16**, **VGG19**, and **ResNet50** along with **LSTM** models. Upon detecting violence, it sends automated email alerts to authorities for quick intervention. Additionally, it analyzes historical crime data (**9,840** records) to identify trends, enabling better law enforcement strategies.

The violence detection model processes frames using feature extraction and sequence modeling, distinguishing between violent and non-violent activities with high accuracy. Meanwhile, crime analysis leverages statistical techniques and forecasting models to uncover regional crime patterns. The system is designed for scalability, ensuring adaptability to diverse surveillance environments and crime datasets.

By combining **computer vision**, **deep learning**, and **statistical analysis**, CrimeGuard offers an advanced framework for proactive crime detection and prevention, assisting law enforcement in maintaining public safety.

II. Related Work

Crime detection and analysis have been the focus of numerous studies, leveraging machine learning, deep learning, and statistical models to improve accuracy and efficiency in identifying criminal activities. Traditional methods relied on manual data collection and statistical techniques, which often lacked scalability and real-time applicability [1]–[3].

Several studies [4]–[7] have utilized deep learning-based models for violence detection, employing **Convolutional Neural Networks (CNNs)** and **Recurrent Neural Networks (RNNs)** for video classification. **CNN+LSTM** architectures have demonstrated improved performance by capturing both spatial and temporal features in violent activity recognition. **MobileNetV2** and **ResNet50** have also been explored for their lightweight architectures and efficient feature extraction capabilities [8], [9]. Despite promising results, many models face challenges such as high computational costs and difficulty in generalizing across different environments.

Other works [10], [11] focus on crime data analysis using predictive models. Some studies employ time-series forecasting models like **ARIMA** and **LSTMs** to predict future crime occurrences based on historical data [12], [17]. However, challenges such as data imbalance and regional biases remain prevalent in crime prediction research.

Real-time alerting systems have also been explored, integrating cloud-based solutions with machine learning models to provide instant notifications upon detecting criminal activities [13], [14]. Implementing automated reporting systems and email notifications in specific frameworks has improved response efficiency in law enforcement applications.

Various limitations such as limited data availability, misinterpretation of video, and other real-world challenges are still present. It is necessary to address these challenges by improving current models and developing solutions to enhance their efficiency [15], [16].

III. Proposed Model

The model integrates crime analysis, prediction and real-time violence detection. It utilizes data analytics, statistical methods and deep learning technologies. This helps in better decision-making and in implementing preventive measures towards improving public safety against crimes.

A. Data Collection and Pre-processing

The proposed system involves collecting data related to crime including historical documents. The data is pre-processed and normalized to improve accuracy. Duplicate values are eliminated and missing values are handled to ensure consistency.

- **Feature Selection**: The purpose of feature selection is to improve the accuracy of prediction. The major features for this study are relevant crime indicators such as type of crime, state, district, and year.
- **Data Cleaning and Transformation**: This step is very critical in data preprocessing. In this step, we use procedures for data files. Normalization of numerical properties, handling of missing values, and encoding of categorical functions directly impact the result. We need data in a suitable format, so data transformation is required.
- **Data Visualization**: To simplify complex data, we use heatmaps, bar charts, and line graphs. Generally, these tools provide graphic representations in this study and are used to examine crime trends over time and in various places.
- **Time-Series Forecasting**: To predict future values, historical patterns are required. ARIMA model is used to forecast future crime incidents for this study. By using ARIMA, we analyze the crime data on the basis of past history. This is helpful in strategic law enforcement, like identifying crime trends and hotspots for the upcoming years.

B. Feature Extraction

Deep learning models automatically extract the features from data. Pre-trained models like CNN, VGG16, VGG19, ResNet50, and MobileNetV2 in association with LSTM are used to extract significant features from video frames. These models can be fine-tuned for real-time monitoring. LSTM extracts temporal dependency from data and concentrates on important trends associated with violent acts.

1) Key Feature Extraction Techniques:

- **Frame-Wise Feature Extraction:** To identify aggressive behavior or unusual activity, CNN-based models like VGG16, VGG19, ResNet50, and MobileNetV2 detect edges, textures, and objects in each video frame. These models can identify important features across multiple frames.
- **Temporal Relationship Modeling by LSTM:** It is used to capture relationship between events over time. The LSTM layer tracks temporal changes across frames and handles sequential data by preserving long-term dependencies. It captures patterns that help indicate violence and allow to understand change of motion over time.
- **Spatio-Temporal Fusion:** This integration combines spatial features and temporal dependencies in data. Characteristics such as shapes, textures, and objects are extracted from frames while also tracking changes in motion over time. Using CNN and LSTM together allows for better differentiation between violent and non-violent activities, improving detection accuracy.
- **Key Indicators of Violence:** The model helps to identify patterns that have distinct movement patterns(aggressive behaviours) like rapid fluctuations or posture changes that are different from normal behaviour.

C. Model Training and Inference

Real-time violence detection and crime trend forecasting are the two main functions that the system is trained to accomplish.

1) Violence Detection (Deep Learning): Deep learning-based models, like CNNs, and temporal dependency models like LSTM can analyze spatial and temporal patterns in video sequences. To improve accuracy for the detection of violence or crime prediction, CNN extracts visual details like textures, shapes, and object structures from each frame. The current model will help to identify key spatial features as well as detect motion patterns. LSTM processes the sequence of frames to detect movement changes, dependencies, and motion flow over time. By working together, these models provide a more accurate understanding of actions and behaviors in videos.

a) Mathematical Formulation for Violence Detection: Given an input video sequence represented as frames F_t, where $t = 1, 2, \ldots, T$, the spatial feature extraction using a CNN is defined as:

$$S_t = \mathrm{CNN}(F_t), \tag{1}$$

where S_t represents the extracted spatial features at time t.

To capture temporal dependencies, the LSTM processes sequential feature vectors:

$$h_t = \sigma(W_h \cdot S_t + U_h \cdot h_{t-1} + b_h), \tag{2}$$

where h_t is the hidden state at time t, W_h, U_h, and b_h are trainable parameters, and σ is the activation function.

The final classification probability is obtained using a softmax function:

$$P(y \mid F) = \text{Softmax}(W_o \cdot h_T + b_o), \tag{3}$$

where W_o and b_o are output layer parameters.

2) Crime Trend Forecasting (Time-Series Analysis): Crime trends over time are modeled using statistical forecasting techniques. Given a time series of crime incidents C_t, the future crime count is estimated:

$$C_t = \phi_1 C_{t-1} + \phi_2 C_{t-2} + \cdots + \phi_p C_{t-p} + \epsilon_t, \tag{4}$$

where ϕ_i are autoregressive coefficients and ϵ_t represents random noise.

To model seasonality and trend components, an **ARIMA** model is employed:

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d C_t = (1 + \sum_{j=1}^{q} \theta_j L^j)\epsilon_t, \tag{5}$$

where L is the lag operator, d is the differencing order, and θ_j are moving average coefficients.

D. Data visualization

Crime data, including historical records and predictive forecasts, are visualized through various graphical representations:

- **Heatmaps**: Show geospatial crime distribution.
- **Bar and Pie Charts**: Provide statistical breakdowns of crime types and occurrences.

E. Interactive Dashboard

An interactive dashboard is developed to allow users, including law enforcement and policymakers, to dynamically analyze crime trends. The dashboard provides:

- **Customizable Filters**: Users can filter data based on state, district, year, and crime category.
- **Real-Time Violence Detection**: Integrates a real-time surveillance system that supports both **live camera feeds** and **file uploads** for detecting violent activities.
- **Graphical Insights**: Interactive charts and visualizations enable an intuitive exploration of crime patterns.

F. Algorithm Flow

- **Crime Data Preprocessing**: Load and clean crime data, normalize features, and split data for training and testing.
- **Crime Trend Prediction (ARIMA)**: Use historical crime data for forecasting trends using the ARIMA model.
- **Video Preprocessing**: Extract frames from videos, preprocess frames, and resize for CNN processing.
- **Violence Detection (CNN-LSTM)**: Use CNN to extract spatial features and LSTM to analyze temporal dependencies for violence classification.
- **Alert System**: Trigger alerts if violence is detected via email notification.

The following pseudo code provides a high-level overview:

```
# Data Preprocessing (Crime Analysis)
LOAD crime_data.csv
HANDLE missing values
SPLIT dataset (80% train, 20% test)

# Model Training (Crime Trend Forecasting)
FOR each state:
    FIT ARIMA on historical crime data
    FORECAST crime trends for next 5 years

# Video Preprocessing (Violence Detection)
LOAD video
EXTRACT frames, preprocess (resize,
normalize)

# Model Inference (Violence Detection)
LOAD CNN model (MobileNetV2/VGG16/VGG19/
CNN/ResNet50)
FOR each frame:
    EXTRACT features with CNN
    INPUT features to LSTM for violence
    prediction
    IF violence detected, trigger alert

# Backend API (Flask)
DEFINE '/upload_video' METHOD POST:
    RECEIVE video, preprocess frames,
    apply violence detection, return status

DEFINE '/send_alert' METHOD POST:
    SEND email notification if violence detected
```

IV. Evaluation and Results

This section presents the performance evaluation of the proposed violence detection models. Various deep learning architectures were tested on the dataset, and their accuracy, precision, recall, and F1-score were recorded.

A. Evaluation Metrics

To assess the effectiveness of the models, the following evaluation metrics were used:

- **Accuracy:** Measures the overall correctness of the model and is given by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

- **Precision:** Represents the proportion of correctly predicted positive instances among all predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

- **Recall:** Measures the proportion of actual positive instances correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

- **F1-score:** Provides a harmonic mean of precision and recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

Here, TP, TN, FP, and FN denote the true positives, true negatives, false positives, and false negatives, respectively.

B. Model Performance

To compare different deep learning architectures, multiple models were trained and tested. The table below presents the results:

TABLE I
PERFORMANCE METRICS OF DIFFERENT MODELS

Model	Accuracy (%)	Precision	Recall	F1-score
CNN+LSTM	**96.0**	0.95	0.96	0.96
MobileNetV2+LSTM	93.0	0.93	0.93	0.93
ResNet50+LSTM	60.0	0.61	0.60	0.60
VGG16+LSTM	89.0	0.89	0.89	0.89
VGG19+LSTM	85.0	0.85	0.85	0.85

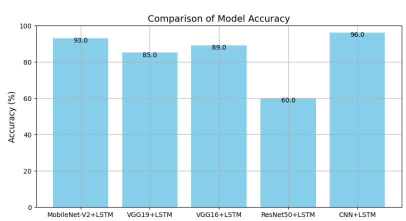


Fig. 1. Comparison of Model Accuracy for Violence Detection

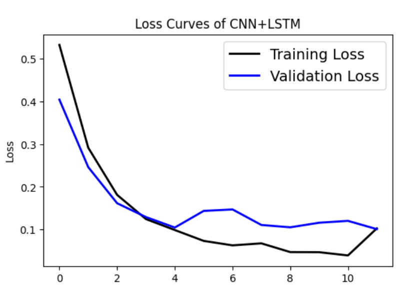


Fig. 2. Loss Curve of CNN+LSTM model

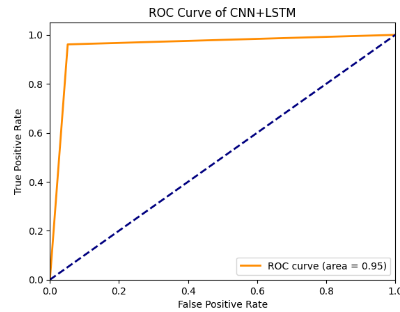


Fig. 3. Receiver Operating Characteristic (ROC) Curve of CNN+LSTM model

C. Computational Complexity

The computational efficiency of each model was assessed in terms of training time and inference time. The results are summarized in Table II.

TABLE II
COMPUTATIONAL COMPLEXITY (IN SECONDS)

Model	Training Time (s)	Inference Time (s)
CNN+LSTM	2520	0.014084
MobileNetV2+LSTM	1025	0.028255
ResNet50+LSTM	604	0.143210
VGG16+LSTM	383	0.035911
VGG19+LSTM	2058	0.043785

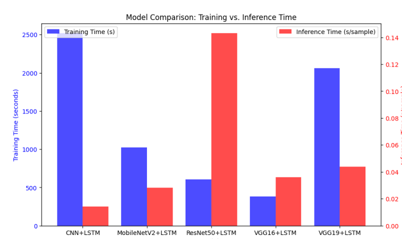


Fig. 4. Computational Complexity (Training and Inference Time)

D. Observations

From the results, the following observations can be made:

- The **CNN+LSTM** model achieved the highest accuracy (96.0%), outperforming all other models in precision, recall, and F1-score. Figure 1 highlights CNN+LSTM as the best-performing model.
- The **MobileNetV2+LSTM** model performed exceptionally well with 93.0% accuracy, making it a strong lightweight alternative for real-time applications.
- Hybrid CNN-LSTM architectures like **VGG16+LSTM** and **VGG19+LSTM** performed reasonably well, achieving 89.0% and 85.0% accuracy, respectively.
- **ResNet50+LSTM** underperformed with an accuracy of 60.0%, indicating potential overfitting or inefficiencies in feature extraction for violence detection.
- The loss curve (Figure 2) depicts smooth reduction, validating the optimization process.

- The ROC curve (Figure 3) indicates a high area under the curve (AUC) for CNN+LSTM, further proving its superior predictive performance.
- **Computational Complexity:** As shown in Table II and Figure 4, CNN+LSTM has the highest training time (2520s) but the lowest inference time (0.014s), making it ideal for real-time violence detection.
- MobileNetV2+LSTM provides a balance between computational efficiency and accuracy, achieving 93.0% accuracy while maintaining a relatively low training time of 1025s.
- ResNet50+LSTM had a very high inference time (0.143s), making it less suitable for real-time applications.
- Future optimizations may focus on reducing training complexity while maintaining high detection accuracy.

E. Result

The evaluation results demonstrate that deep learning models, particularly CNN+LSTM, significantly enhance violence detection accuracy. These findings support the use of hybrid architectures in real-world surveillance systems, improving public safety through automated violence detection.

V. Conclusion and Future Scope

The proposed system enhances public safety by combining predictive crime analysis with real-time violence detection using spatial feature extraction with motion analysis. Violent incidents can be detected in real-time generating alerts so that faster response can be taken to prevent them. This allows for better crime analysis and in taking preventive measures in advance. Real-time violent detection, future crime prediction and data visualizations allow concerned authorities to make policies and strategies in order to reduce crime rates and improve public security. Analyzing trends in data can help in law enforcement and enhancing decision-making.

The results obtained show that deep learning-based violence detection increases accuracy of the model. The CNN+LSTM approach performed the best among other models, highlighting the effectiveness of hybrid models for real-world surveillance. Additionally, time-series forecasting using ARIMA model for predictive crime analysis provides useful analysis into crime trends, helping authorities optimize resource allocation and improve their strategic planning.

A. Limitations

Several limitations are present in spite of promising results obtained through the proposed system:

- **Dataset Constraints:** The efficiency of the model depends on data which has limited availability. Efficiency of the model depends on quality and diversity of data. Privacy and various restrictions are also the concern. The system may not be able to perform well in practical scenario if dataset size is not sufficient. Restrictions and privacy are also the challenges in crime-related data availability.
- **Environmental Factors:** Environmental actors such as changes in camera angles, objects blocking the view, and poor lighting conditions can affect the accuracy of video-based violence detection. Crowded places also create challenges in identifying violent activities accurately.
- **False Positives and Negatives:** The model can sometimes misinterpret body gestures, making it possible for misclassification of events despite showing high accuracy. There is a need for continuous improvements and refinements in training data to avoid challenges like bias.

B. Future Scope

To improve the effectiveness of the system, future work can explore the following domains:

- **Multimodal Analysis:** Inclusion of audio data for analysis can enhance the accuracy of violence detection. For understanding of various complex environments like crowded place where visuals are not alone enough to take better decision, audio data can be beneficial.
- **Edge Computing:** It is highly useful in real time detection. Deployment of models on edge devices causes faster processing in real-time, reduced latency, lower bandwidth consumption, and improved privacy.
- **Enhanced Predictive Modeling:** Using advanced machine learning and deep learning, like deep reinforcement learning, geo spatial crime analysis, vision transformers, can help make crime predictions more accurate.
- **Partnerships and Collaborations:** Support to government, private sector, civil organisations, smart city developers, and surveillance technology providers which expand its real-world adoption for public safety.

This system creates a framework for AI-driven crime prevention. Improving its accuracy and using advanced analysis, it can help make communities safer and assist law enforcement in providing responding more effectively.

References

[1] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88-97, 2018.

[2] B. Sathyadevan and S. Gangadharan, "Crime analysis and prediction using data mining techniques," *2014 First International Conference on Networks & Soft Computing (ICNSC)*, 2014, pp. 406-412.

[3] M. Ullah, M. A. Muhammad, G. H. Lee, and S. W. Baik, "Violence detection using spatiotemporal features with 3D convolutional neural networks," *Sensors*, vol. 19, no. 11, p. 2472, 2019.

[4] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138-52160, 2018.

[5] X. Xu, L. Yang, H. Wang, and F. Zhao, "Edge computing: A new computing model for Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1778-1786, 2020.

[6] N. K. Sharma, M. Pandey, and R. K. Gupta, "IoT-based smart surveillance systems for crime detection and prevention," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 8473-8490, 2021.

[7] G. Gorr and R. Harries, “Introduction to crime forecasting,” *International Journal of Forecasting*, vol. 19, no. 4, pp. 551-555, 2003.

[8] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, “ActivityNet: A large-scale video benchmark for human activity understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 961-970.

[9] T. Hassner, M. Itcher, and R. Kliper-Gross, “Violent flows: Real-time detection of violent crowd behavior,” in *Proceedings of IEEE Workshop on Socially Intelligent Surveillance and Monitoring (SISM)*, 2012, pp. 1-6.

[10] Y. Zhang, Z. He, and S. Zhang, “A survey on violence detection from videos,” *Multimedia Tools and Applications*, vol. 79, no. 39, pp. 29321-29350, 2020.

[11] P. Sudhakaran and O. Lanz, “Learning to detect violent videos using convolutional long short-term memory,” in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1-6.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations (ICLR)*, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.

[14] A. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[16] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1993, pp. 207-216.

[17] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control*, 5th ed., Wiley, 2015.

Report (1).docx

Yeshwantrao Chavan College of Engineering, Nagpur, India

Document Details

Submission ID

trn:oid:::27005:97047463

Submission Date

May 21, 2025, 7:53 PM GMT+5:30

Download Date

May 21, 2025, 7:54 PM GMT+5:30

File Name

Report (1).docx

File Size

2.1 MB

51 Pages

7,714 Words

47,980 Characters

17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

89 Not Cited or Quoted 17%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

13% Internet sources
6% Publications
14% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **89** Not Cited or Quoted 17%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

13%	Internet sources
6%	Publications
14%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	**KIET Group of Institutions, Ghaziabad on 2024-05-24**	3%
2	Submitted works	**KIET Group of Institutions, Ghaziabad on 2024-05-24**	1%
3	Internet	**www.coursehero.com**	<1%
4	Internet	**arxiv.org**	<1%
5	Submitted works	**Liverpool John Moores University on 2025-02-25**	<1%
6	Internet	**www.mdpi.com**	<1%
7	Internet	**seyboldpublications.com**	<1%
8	Publication	**Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ...**	<1%
9	Internet	**pdfcoffee.com**	<1%
10	Submitted works	**Arab Open University on 2025-05-05**	<1%

11 Internet

www.americaspg.com <1%

12 Submitted works

Queen Mary and Westfield College on 2023-08-25 <1%

13 Internet

www.bright-journal.org <1%

14 Publication

T. Mariprasath, Kumar Reddy Cheepati, Marco Rivera. "Practical Guide to Machin... <1%

15 Submitted works

University of Lincoln on 2023-09-07 <1%

16 Internet

www.frontiersin.org <1%

17 Submitted works

ABES Engineering College on 2019-04-25 <1%

18 Internet

www.ir.juit.ac.in:8080 <1%

19 Submitted works

Hankuk University of Foreign Studies on 2024-05-05 <1%

20 Submitted works

University of Staffordshire on 2025-05-19 <1%

21 Internet

www.freepatentsonline.com <1%

22 Internet

ip.ios.semcs.net <1%

23 Submitted works

Nottingham Trent University on 2025-05-09 <1%

24 Internet

ir.juit.ac.in:8080 <1%

25 Submitted works

Technological Institute of the Philippines on 2025-05-16 <1%

26 Submitted works

Universiti Malaysia Sarawak on 2018-12-17 <1%

27 Submitted works

VIT University on 2025-04-23 <1%

28 Internet

www.cse.griet.ac.in <1%

29 Submitted works

University of Suffolk on 2025-02-04 <1%

30 Submitted works

University of Wales Institute, Cardiff on 2025-03-13 <1%

31 Internet

www.ijnrd.org <1%

32 Internet

www2.mdpi.com <1%

33 Publication

Ohri, Ankit. "Real-Time American Sign Language Recognition Using Webcam, Co... <1%

34 Internet

dokumen.pub <1%

35 Internet

e-research.siam.edu <1%

36 Internet

peerj.com <1%

37 Publication

Armin Choupani, Sadoon Azizi, Mohammad Sadegh Aslanpour. "Joint resource au... <1%

38 Submitted works

Australian National University on 2023-05-28 <1%

39 Publication

Chutchai Kaewta, Rapeepan Pitakaso, Surajet Khonjun, Thanatkij Srichok et al. "A... <1%

40 Submitted works

University of Hull on 2023-05-05 <1%

41 Internet

technodocbox.com <1%

42 Internet

tel.archives-ouvertes.fr <1%

43 Internet

tudr.thapar.edu:8080 <1%

44 Publication

Al-Mahmud Al Mamun, Md Rasel Hossain, Mst Mahfuza Sharmin. "Detection and ... <1%

45 Submitted works

Coventry University on 2025-04-10 <1%

46 Submitted works

Meerut Institute of Engineering & Technology on 2020-05-14 <1%

47 Submitted works

Middlesex University on 2024-10-05 <1%

48 Submitted works

Technological Institute of the Philippines on 2025-05-16 <1%

49 Internet

assets-eu.researchsquare.com <1%

50 Internet

fastercapital.com <1%

51 Internet

harbinengineeringjournal.com <1%

52 Internet

hdl.handle.net <1%

53 Internet

whatis.techtarget.com <1%

54 Internet

www-emerald-com-443.webvpn.sxu.edu.cn <1%

55 Submitted works

APJ Abdul Kalam Technological University, Thiruvananthapuram on 2025-05-15 <1%

56 Submitted works

AUT University on 2024-06-10 <1%

57 Submitted works

Concordia University on 2025-01-11 <1%

58 Submitted works

Middlesex University on 2025-04-10 <1%

59 Submitted works

UCL on 2025-04-25 <1%

60 Submitted works

University of Cape Town on 2022-02-10 <1%

61 Submitted works

University of Hertfordshire on 2024-08-12 <1%

62 Submitted works

University of Hertfordshire on 2025-04-27 <1%

63 Submitted works

University of Westminster on 2021-05-03 <1%

64 Submitted works

Université Virtuelle de Tunis on 2025-05-08 <1%

65 Publication

Vibhav Sachan, Shahid Malik, Ruchita Gautam, Kumar Parvin. "Advances in AI for ... <1%

66 Publication

Vinay Dubey, Rahul Katarya. "FloodCNN-BiLSTM: Predicting flood events in urban ... <1%

67 Internet

f6publishing.blob.core.windows.net <1%

68 Internet

like.edu.np <1%

69 Internet

www.turcomat.org <1%

Project Report 2.docx

Yeshwantrao Chavan College of Engineering, Nagpur, India

Document Details

Submission ID

trn:oid:::27005:96008527

Submission Date

May 15, 2025, 12:56 PM GMT+5:30

Download Date

May 21, 2025, 9:37 AM GMT+5:30

File Name

Project Report 2.docx

File Size

2.1 MB

45 Pages

6,194 Words

38,295 Characters

*% detected as AI

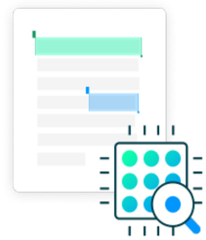
AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.



False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

Publication Partner :

Taru Publications

IJTE-ISTE
Peer Review

INTERNATIONAL CONFERENCE
ON
RECENT ADVANCES IN COMPUTATIONAL
AND MATHEMATICAL TECHNIQUES (RACMT- 2025)

Organized by : Department of Computer Science and Engineering, JIET Jodhpur
In Collaboration with : Department of Mathematics, PIET, Jaipur

Scopus

JIET UNIVERSE

CERTIFICATE OF PARTICIPATION

This is to certify that paper entitled

CrimeGuard: Real-time Violence Detection and Predictive Analysis System

of KIET Group of Institutions was presented / contributed by

Shreya Goswami in the International Conference

on "Recent Advances in Computational and Mathematical Techniques" (RACMT-2025) held during 25th – 26th April, 2025 at

Jodhpur Institute of Engineering & Technology, Jodhpur (Raj.)

Prof. (Dr.) Avnish Bora Director - JIET	**Prof. Mamta Garg** Dean, Faculty of Computer Engineering – JIET	**Prof. (Dr.) Rajendra Purohit** Professor - CSE, JIET & Conference Convener	Priya Mathur **Prof. (Dr.) Priya Mathur** Professor - Mathematics, PIET, Jaipur & Conference Convener

Publication Partner :

Taru Publications

IJTE-ISTE
Peer Review

INTERNATIONAL CONFERENCE
ON
RECENT ADVANCES IN COMPUTATIONAL
AND MATHEMATICAL TECHNIQUES (RACMT- 2025)

Organized by : Department of Computer Science and Engineering, JIET Jodhpur
In Collaboration with : Department of Mathematics, PIET, Jaipur

Scopus

JIET UNIVERSE

CERTIFICATE OF PARTICIPATION

This is to certify that paper entitled

CrimeGuard: Real-time Violence Detection and Predictive Analysis System

of KIET Group of Institutions was presented / contributed by

Khushi Bansal in the International Conference

on "Recent Advances in Computational and Mathematical Techniques" (RACMT-2025) held during 25th – 26th April, 2025 at

Jodhpur Institute of Engineering & Technology, Jodhpur (Raj.)

Mamta

Priya Mathur

Prof. (Dr.) Avnish Bora	**Prof. Mamta Garg**	**Prof. (Dr.) Rajendra Purohit**	**Prof. (Dr.) Priya Mathur**
Director - JIET	Dean, Faculty of Computer Engineering – JIET	Professor - CSE, JIET & Conference Convener	Professor - Mathematics, PIET, Jaipur & Conference Convener

Publication Partner :

Taru Publications

IJTE-ISTE
Peer Review

INTERNATIONAL CONFERENCE
ON
RECENT ADVANCES IN COMPUTATIONAL
AND MATHEMATICAL TECHNIQUES (RACMT- 2025)

Organized by : Department of Computer Science and Engineering, JIET Jodhpur
In Collaboration with : Department of Mathematics, PIET, Jaipur

Scopus

JIET UNIVERSE

CERTIFICATE OF PARTICIPATION

This is to certify that paper entitled

CrimeGuard: Real-time Violence Detection and Predictive Analysis System

of KIET Group of Institutions was presented / contributed by

Mahi Tyagi in the International Conference

on "Recent Advances in Computational and Mathematical Techniques" (RACMT-2025) held during 25th – 26th April, 2025 at

Jodhpur Institute of Engineering & Technology, Jodhpur (Raj.)

Prof. (Dr.) Avnish Bora Director - JIET	**Prof. Mamta Garg** Dean, Faculty of Computer Engineering – JIET	**Prof. (Dr.) Rajendra Purohit** Professor - CSE, JIET & Conference Convener	**Prof. (Dr.) Priya Mathur** Professor - Mathematics, PIET, Jaipur & Conference Convener