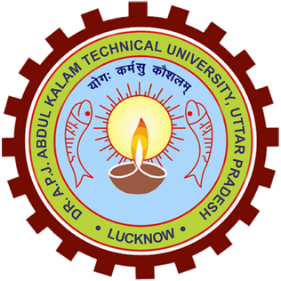
****

**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)

**Feb, 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:

Roll No.:

Date:

## CERTIFICATE

This is to certify that Project Report entitled “CrimeGuard” which is submitted by Student name 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.

Date:

Signature:

Name :

Roll No.:

**ABSTRACT**

CrimeGuard is an extensive system which is intended for predictive crime analysis as well as real-time violence detection through Machine Learning and Computer Vision. The initiative has the objective to optimize public safety through the use of past crime data in order to anticipate possible crime hotspots, and real-time video analysis to identify violent activities. The system incorporates deep learning models like VGG16 for feature extraction and custom-trained neural networks to detect violent activity from surveillance videos. Furthermore, predictive modeling is based on ARIMA to examine crime trends and predict future rates of crime. CrimeGuard possesses an interactive dashboard for visualizing crime data, allowing law enforcement agencies to make evidence-based decisions. Developed with Flask as the backend and HTML, CSS, JavaScript as the frontend, CrimeGuard is an efficient and scalable solution to support crime prevention and response.

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **Page No.** |
| DECLARATION……………………………………………………………………. | ii |
| CERTIFICATE……………………………………………………………………… | iii |
| ACKNOWLEDGEMENTS…………………………………………………………. | iv |
| ABSTRACT………………………………………………………………………..... | v |
| LIST OF FIGURES…………………………………………………………………. | viii |
| LIST OF TABLES…………………………………………………………………… | ix |
| LIST OF ABBREVIATIONS………………………………………………………. | x |
| CHAPTER 1 (INTRODUCTION)…………………………………………………. | 1 |
| 1.1. Overview……………………………………………………………………... | 1 |
| 1.2. Project Description…………………………………..…………………………  1.3.Problem Statement……………………………………………………………….  1.4.Existing System………………………………………………………………….  1.5.User Requirement Analysis………………………………………………………  1.6.Feasibility Study…………………………………………………………………. | 1  2  2  3  4 |
| CHAPTER 2 (LITERATURE REVIEW)…………………………………………. | 5 |
| 2.1.Literature review……………............................................................................. | 5 |
| 2.2. Research Gap .…................................................................................................ | 6 |
|  |  |
| CHAPTER 3 (PROPOSED METHODOLOGY) ….................................................. | 7 |
| 3.1. Methodology ……................................................................................................  3.2. Dataset…………………………………………………………………………..  3.3 Pre-Processing………………………………………………………..…………..  3.4 Dataset Split…………………………………………………………...………….  3.5 Model Architecture……………………………………………………...………..  3.6 Hyperparameter Tuning……………………………………………….………….  3.7 Model Training…………………………………………………………..……….  3.8 Model Prediction and Visualization………………………………………..…….  3.9 Tech Stack………………………………………………………………………...  3.10 Use Case Model…………………………………………………………………  3.11 Training and Testing…………………………………………………………….  3.12 Sequence Diagram………………………………………………………………  3.13 System Architecture……………………………………………………………. | 7  7  8  9  9  10  10  10  10  11  12  14  15 |
| CHAPTER 4 (RESULTS AND DISCUSSION) ........................................................ | 16 |
| 4.1 Results……………………………………………….…………………………… | 16 |
| 4.2 Observations……………………………………………………………………... | 17 |
| 4.3 Conclusion……………………………………………………………………….. | 17 |
| 4.4 Output……………………………………………….…………………………… | 20 |
|  |  |
| CHAPTER 5 (CONCLUSIONS AND FUTURE SCOPE)................................. | 22 |
| 5.1. Conclusion........................................................................................................... | 22 |
| 5.2. Future Scope........................................................................................................ | 22 |
|  |  |
| REFERENCES………………………………………………………………………. | 24 |
|  |  |
|  |  |
|  |  |

**LIST OF FIGURES**  
  
Figure No. Description No. Page  
Figure 1 Dataset Split 8  
Figure 2 Preprocessing Pipeline 9  
Figure 3 Use Case Diagram 11  
Figure 4 Training Workflow 12  
Figure 5 Testing Workflow 13  
Figure 6 Sequence Diagram 14  
Figure 7 System Architecture 15  
Figure 8 Model Accuracy 18  
Figure 9 Loss Curves of CNN+LSTM 18  
Figure 10 ROC Curve 19  
Figure 11 Home Page 20  
Figure 12 Our Services 21

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table. No.** | **Description** | **Page No.** |
| 1.1 | Performance metrics for violence detection models | 16 |

**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**

In today's rapidly evolving digital age, law enforcement and security agencies face a significant issue in preventing crimes and identifying threats in real time. Traditional surveillance methods and manual observation are typically ineffective and lead to delayed responses and missing critical incidents. Furthermore, analyzing large amounts of crime data to identify pattern and predict potential threats remains a difficult task.

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 artificial intelligence system for enhancing public safety by predictive crime analysis and detection of violence in real time. It applies machine learning algorithms to analyze historical crime patterns and predict high-risk zones. Simultaneously, it applies deep learning-based computer vision to monitor live video feeds and identify violent activity such as assaults and fights. 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 are still major issues because of the limitations of manual surveillance and slow response times. Existing systems are not capable of real-time detection, varied environmental conditions, and predictive crime analysis. There is a requirement for an effective solution that can:

Process live video streams to detect violent acts. Forecast hotspots of high crime using historical trends. Instantly alert authorities for rapid intervention. Display crime insights interactively for enhanced decision-making. Through filling these gaps, CrimeGuard provides faster response times, enhanced situational awareness, and active crime prevention.

**1.4 Existing System**

Existing crime detection and prevention mechanisms are based on manual surveillance or rule-based approaches, which have the following drawbacks:

Human Error – Fatigue for surveillance staff and as a result, incidents go unnoticed.

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.

**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 architectures such as ViolenceNet and multi-stream networks, which includes DenseNet, multi-head self-attention, and bidirectional LSTMs, increase ability of detection from person to person violence. However, challenges persist in generalizing these models across diverse datasets [10][12].

Furthermore, motion blob-based techniques have been introduced to prioritize computational

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

prisons and psychiatric centers [14].Complementing it, CNN models capable

of detecting objects such as knives and guns have shown potential in predicting crime scenes

with high accuracy, hence ensuring alerts that are reliable and enhancing public safety measures [15]. These findings highlight the importance of the progress 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: Current records contains specific data, limiting the ability of models to generalize to real-world violence detection. Models often perform well on standard datasets but struggle with changed environments.  
• Real-World Challenges: Models often struggle with varied real-world conditions such as different lighting, crowd sizes, and motion patterns, which affect the accuracy and robustness of violence detection systems.  
• False Positives: High rates of misclassification, especially in complex environments like sports or crowded spaces, result in significant false positives, reducing the reliability of the detection system.  
\* Limited Dataset Size: Many models depend on small datasets, which reduces their generalizability and limits their capability to perform well in diverse real-world situations.

\* Integration with Safety Systems: Few researches addresses integration of violence detection models with real-time alert systems, which are important for preventing incidents and ensuring safety.

\* Real-Time Constraints: Existing models generally face challenges with real-time frame processing, therefore, there is need for high accuracy and low computational cost systems, required in resource constrained environments. These shortcomings highlight critical dimensions which can be improved in developing robust and practical violence detection systems.

**CHAPTER 3**

**PROPOSED METHODOLOGY**

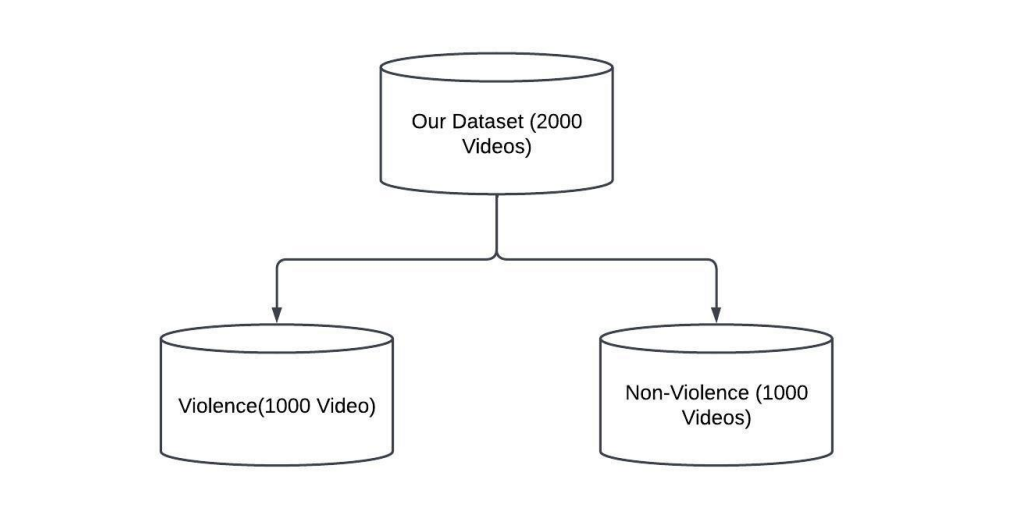
**3.1 Methodology**

In order to deal with the increasing need for real-time crime detection in surveillance systems, we implemented a structured approach that combines deep learning techniques with strategies of practical application. The aim is to develop an efficient system that is capable of accurately identifying violent activities from both recorded videos and live feeds while ensuring timely intervention through generated notifications.The system also includes visualization dashboard for presenting crime data in interactive manner.

The proposed system integrates Convolutional Neural Networks (CNNs) for extracting spatial features from frames of videos and Long Short-Term Memory (LSTM) networks for capturing time based patterns across 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.

**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 order to ensure that the model can generalise across diverse situations and environments, this directory aims to capture different types of violent behaviour in real-world. The diversity and balance in dataset help to improve the model's training, allowing it to differentiate between violent and non-violent behaviours with larger accuracy and fewer false positives. Additionally, a crime dataset containing historical crime records is used for predictive analysis, providing insights into crime trends based on time and location.

Figure 1: Dataset Split

**3.3 Pre-processing**

The pre-processing of the video dataset is a very crucial step in preparing the data for model training. The process starts with reading the input videos using OpenCV. Videos are then splited into individual frames, that allows capture of temporal patterns essential for analyzing activities. The extracted frames are then resized to a fixed dimension of (64 × 64) pixels for ensuring uniformity and reduce 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.

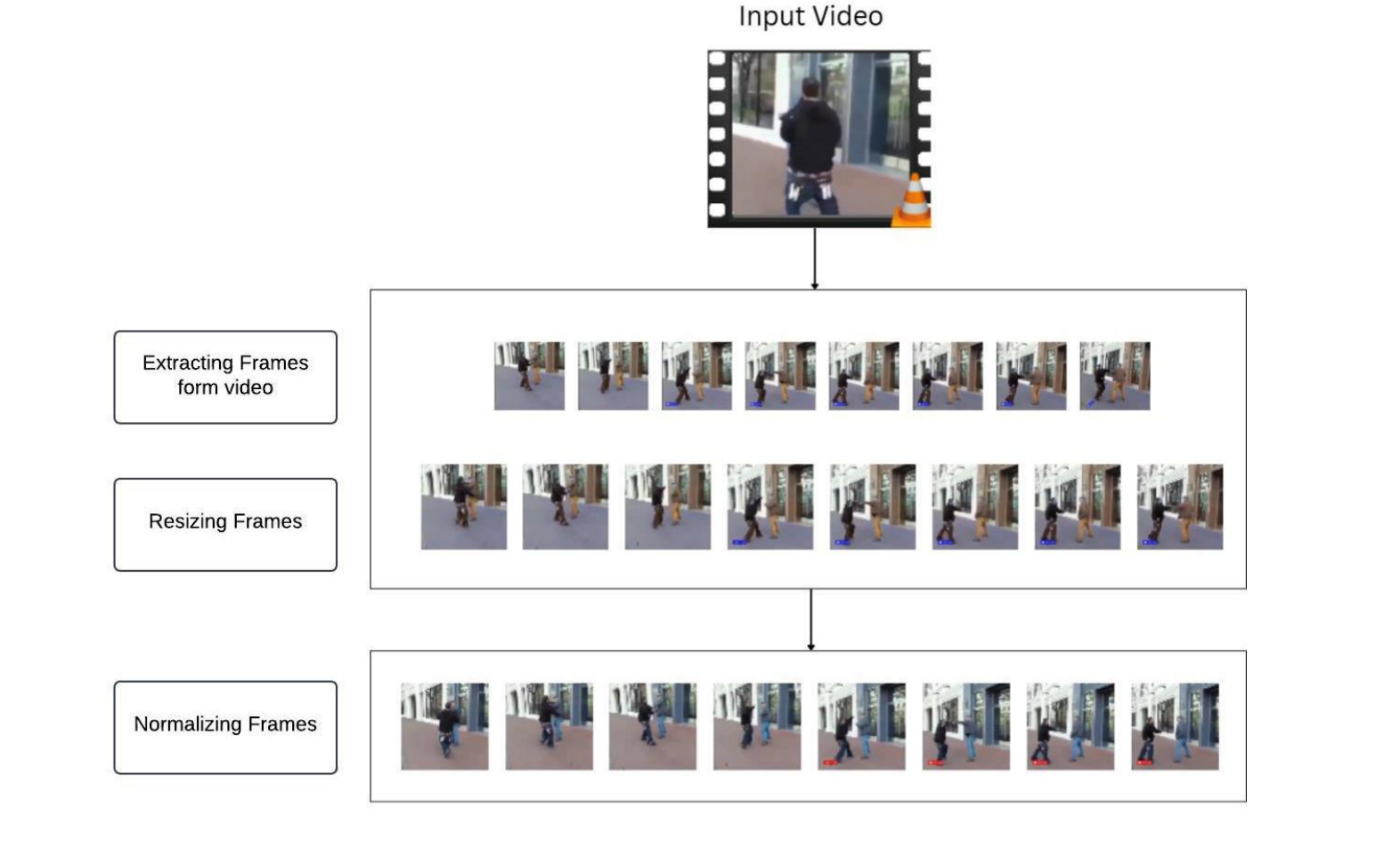


Figure 2: Preprocessing Pipeline

**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 model is trained using the fit() function with a batch size of 8 for 12 epochs. Early stopping is implemented to prevent overfitting. Long short term memory (LSTM) is applied for capturing sequential vedio frames. Also, ARIMA model is used for forecasting total number of crimes.

**3.8 Model Prediction and Visualization**

After training, the CNN-BiLSTM model uses the predict() function to classify real time video sequences. For crime detection, each frame is continuously processed, and alert is sent when violence is detected. Additionally, the predictive crime analysis model forecasts crime count based on historical data. The visualization segment provides real-time graphs, heatmaps, and insights of data analysis results. It also includes interactive tools that allow users to explore trends in crime patterns visually.

**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

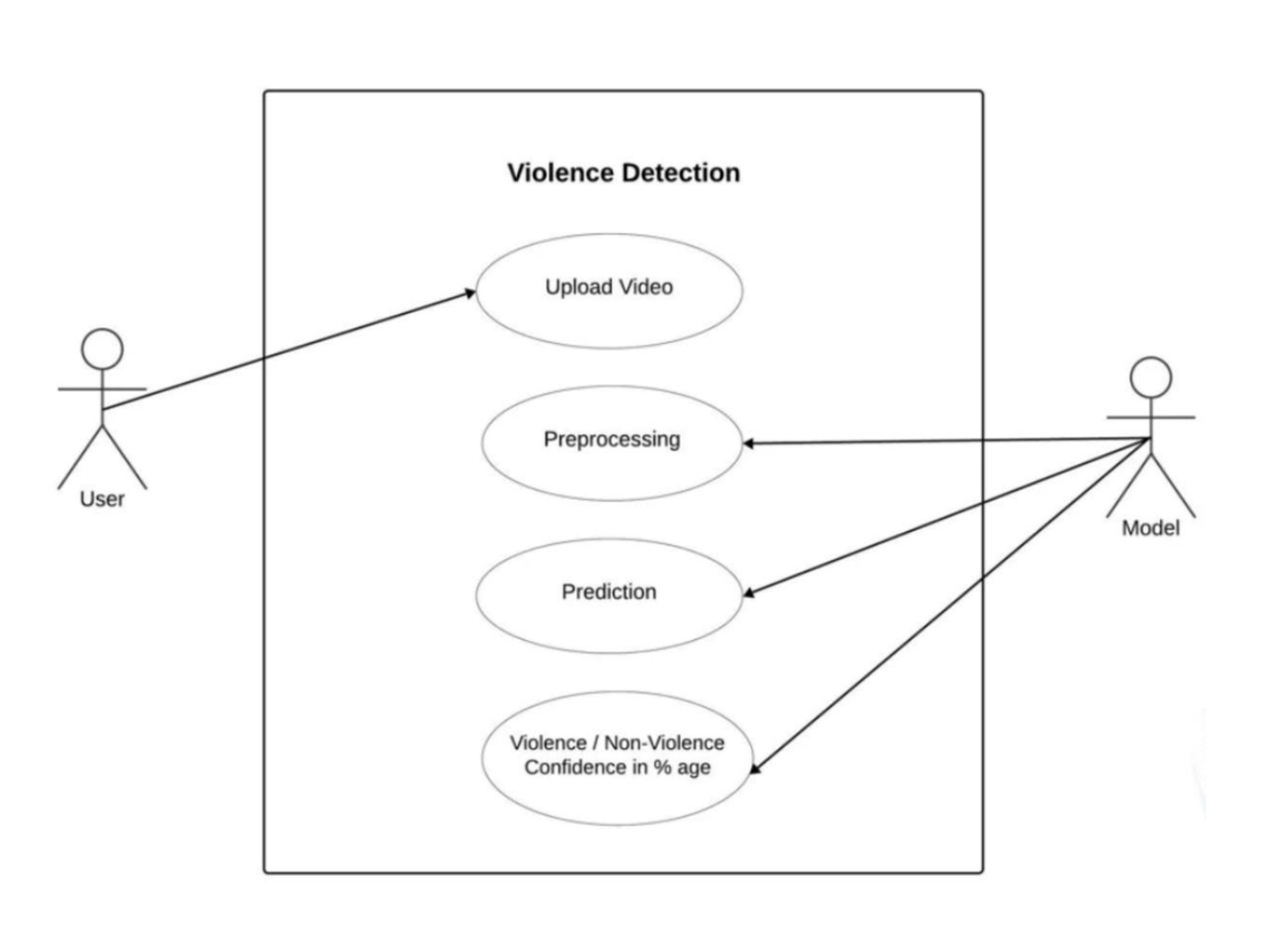
❖ Machine Learning Tools: TensorFlow/Keras, OpenCV, Scikit-learn

❖ 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**

Figure 3: Use case diagram

**3.11 Training and Testing**

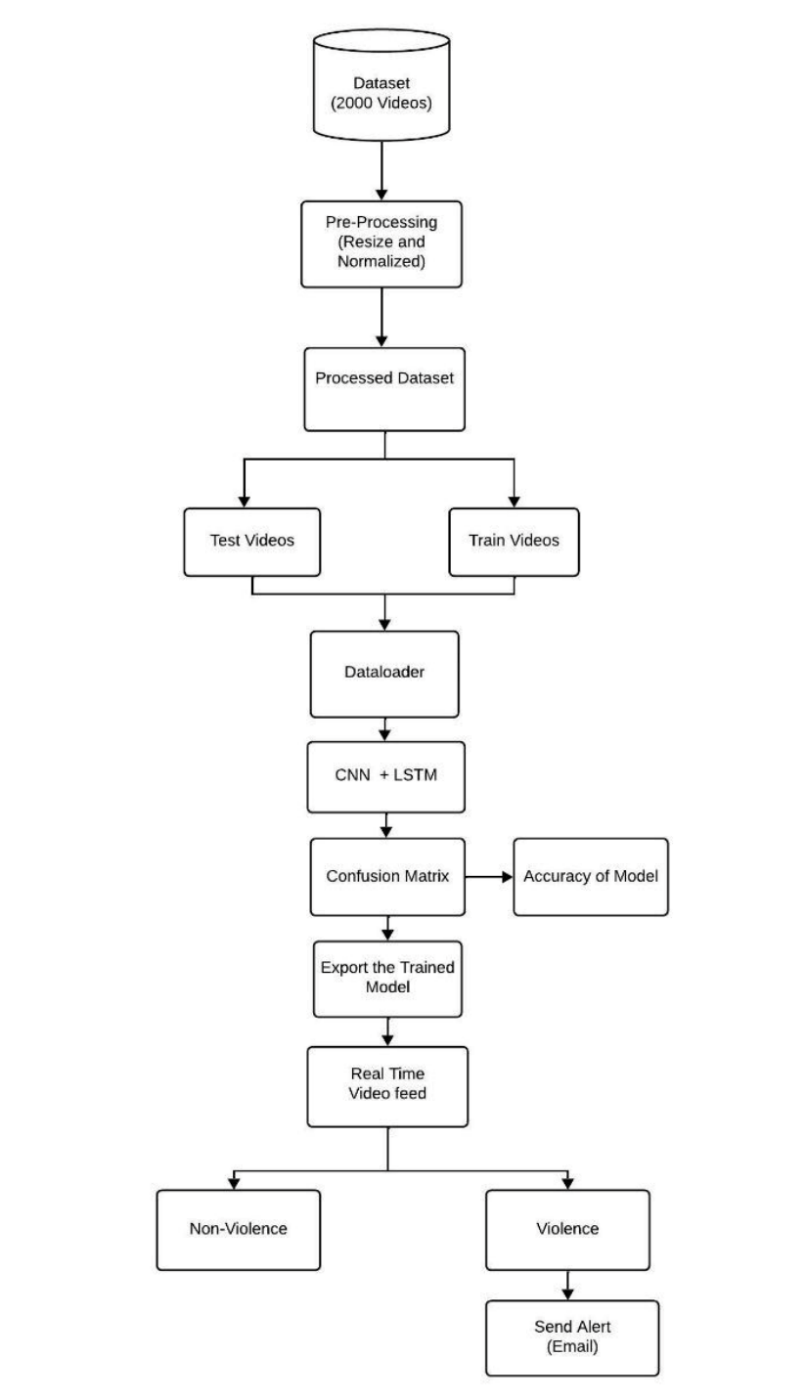


Figure 4: Training Workflow

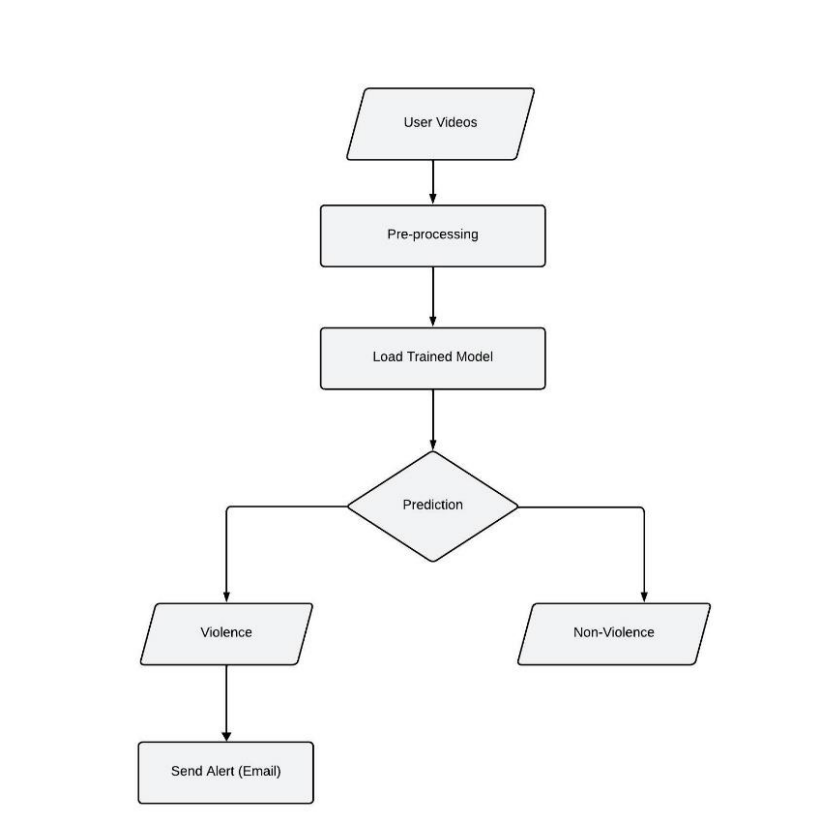


Figure 5: Testing Workflow

**3.12 Sequence Diagram**

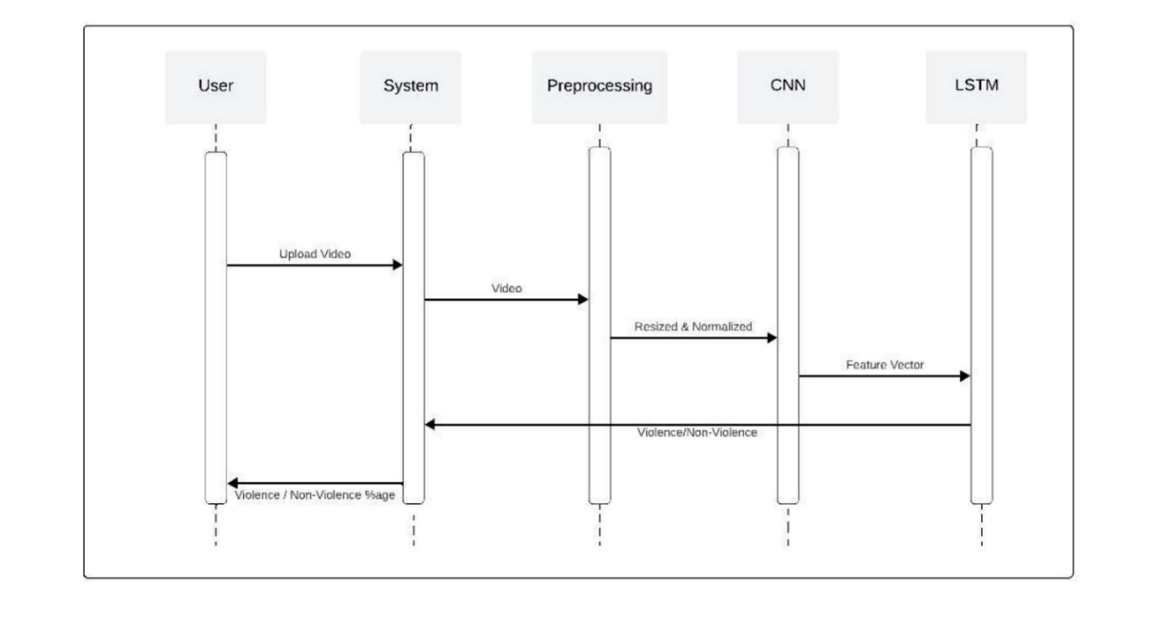


Figure 6: Sequence Diagram

**3.13 System Architecture**

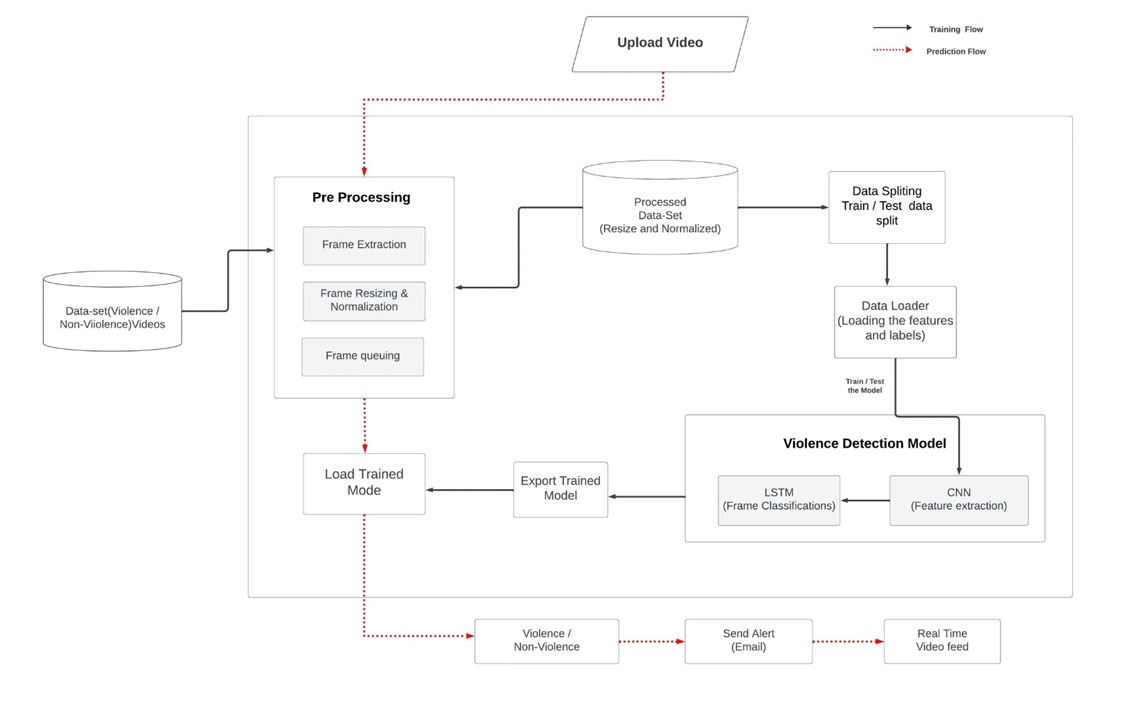


Figure 7: System Architecture

**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.

1. Metrics Definition

• Accuracy: Measures the percentage of correctly classified violent and non-violent incidents.

• Precision: Represents the proportion of detected violent incidents that were actually violent, minimizing false positives.

• Recall: Indicates how well the system identifies all violent events, reducing false negatives.

• F1-Score: The harmonic mean of precision and recall, providing a balanced evaluation of the model’s effectiveness.

• Computational Efficiency: Evaluates the training and inference time required for real-time performance.

1. 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**

• Classification Performance: CNN+LSTM scores the highest accuracy of 96% and F1-score of 0.96, showing its ability to detect violent acts accurately.

• Precision and Recall: MobileNet-V2 too performed well with 93% accuracy, but VGG16 and VGG19 showed moderate results. ResNet50 performed worst due to its inability to capture temporal dependencies well.

**4.3 Conslusion**

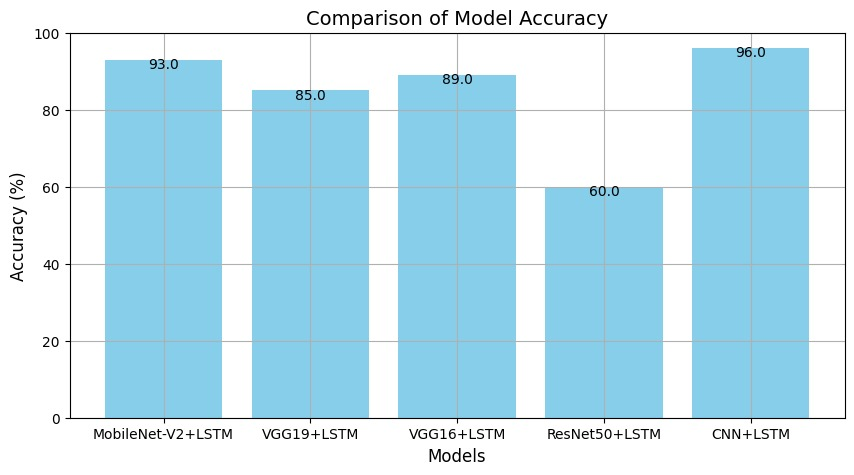
The evaluation results prove that CNN+LSTM is the most effective model for detecting real-time violence, providing the highest accuracy and balanced performance across all metrics. While MobileNet-V2 and VGG16 are close behind, ResNet50 lags due to its struggles to handle sequential frames. The computational complexity analysis ensures that CrimeGuard stays quick and effective for real use. Going forward, the focus will be on making these tech setups faster and even more accurate.

Figure 8: Model Accuracy

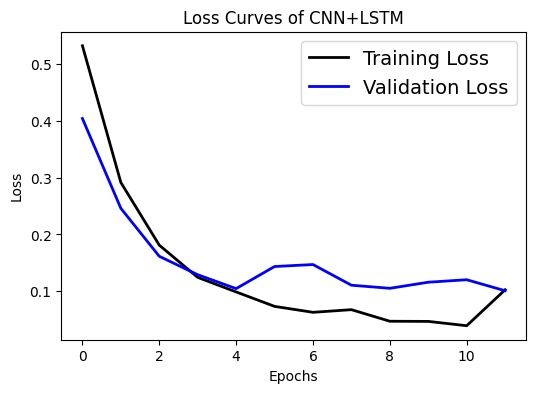


Figure 9: Loss Curves of CNN+LSTM

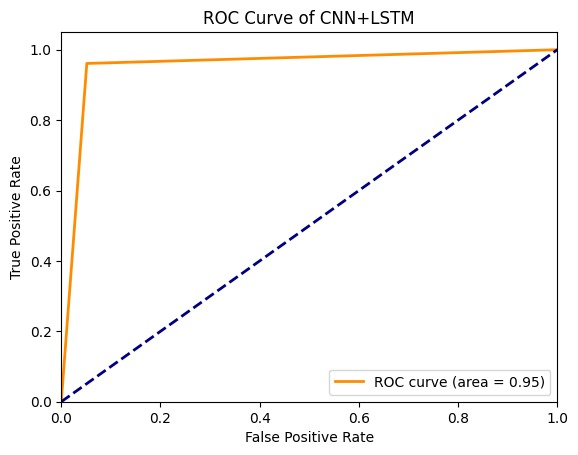
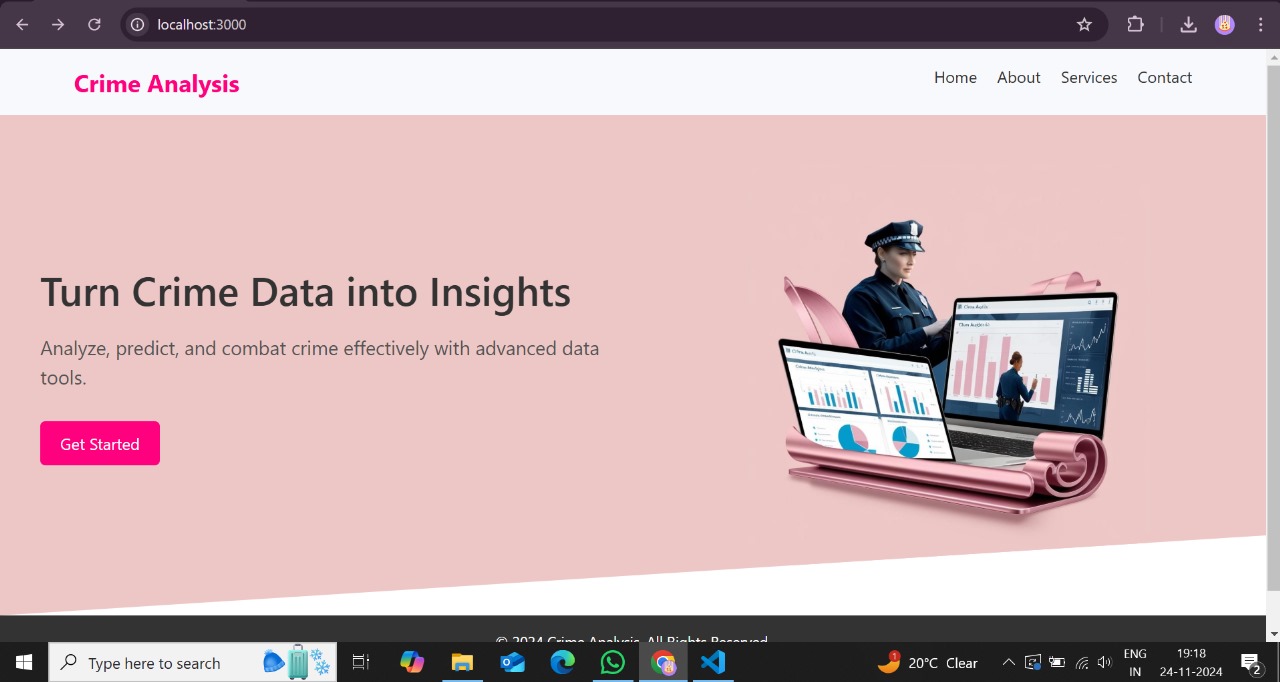
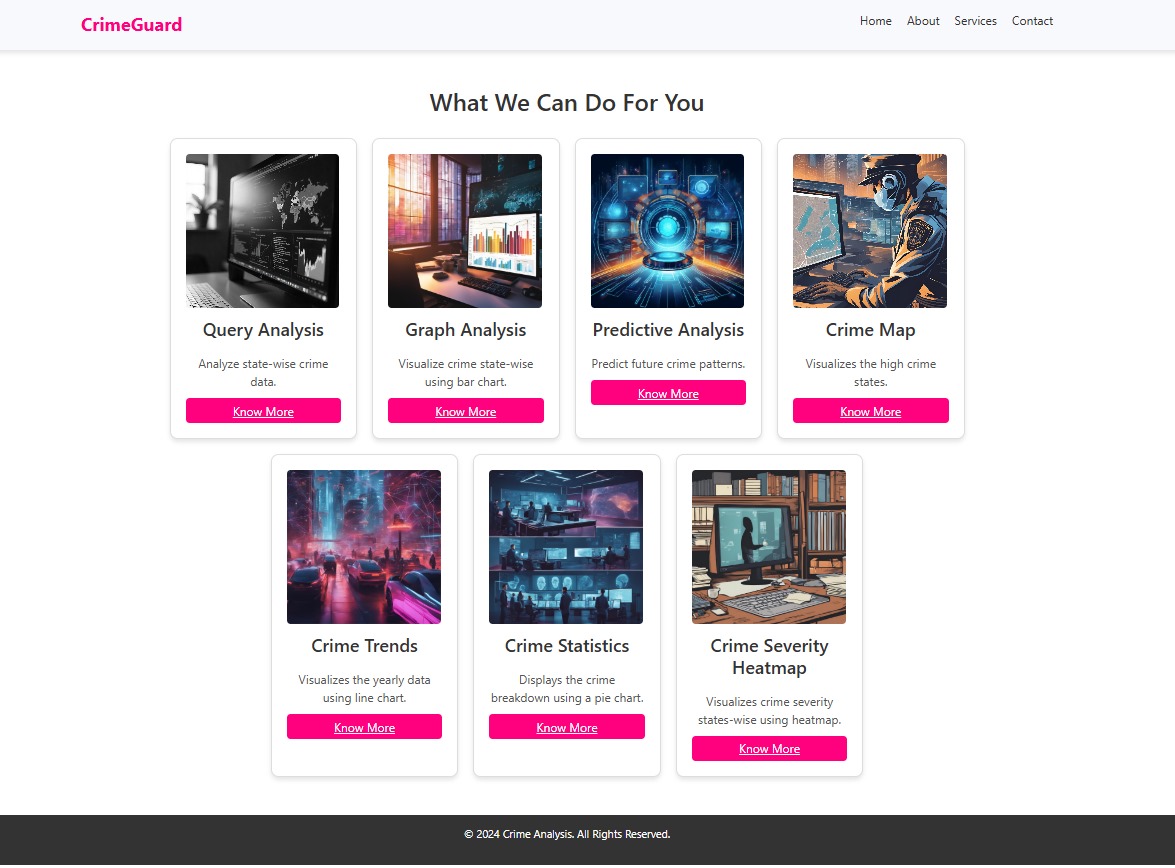


Figure 10 :ROC Curve

**4.4 Output**

Figure 11: Home Page

Figure 12: Our Services

**CHAPTER 5**

**CONCLUSION AND FUTURE SCOPE**

#### **5.1 Conclusion**

CrimeGuard is an innovative crime monitoring and analysis system that improves public safety through real-time violence detection and crime forecasting. It utilizes Convolutional Neural Networks (CNNs) for finding patterns and Long Short-Term Memory (LSTM) networks for continuous data analysis. It accurately detects violent activities and visualizes high risk crime areas based on historical patterns. Through alert system and proactive law enforcement strategies, CrimeGuard empowers authorities to manage threats effectively, guaranteeing public safety through intelligent decision making.

#### **5.2 Future Scope**

* Expanding of Data Sources
  + Use social media and other sites to get better data.
* Reinforcement Learning
  + Make the system better at keeping up with different crime trends.
* Socio-Economic Factors Incorporation
  + Improving crime forecasting accuracy by considering demographic and economic data.
* Algorithm Optimization & Cloud Integration
  + Ensure scalability and real-time performance using optimized algorithms and cloud infrastructure.
* Expansion of Dataset Diversity
  + Train the model on broader datasets to enhance detection accuracy for multiple types of violence.
* Edge Device Deployment
  + Enable real-time processing by deploying the model on edge devices.

These future advancements will **enhance CrimeGuard’s efficiency, scalability, and applicability for real world**.

**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.