Real-Time Crime Detection System Using Video Surveillance Technology

N. Umamaheswari[1], Dr. Anitha T N[2], Sudeep[3], Shwetha S V[4], Govardhana M[5], Gagan D M[6]
[1]*Assistant Professor,vDepartment of CSE Sir M. Visvesvaraya Institute of Technology VTU, Bangalore, India*
[2,3,4,5,6] *Department of CSE, Sir M. Visvesvaraya Institute of Technology VTU, Bangalore, India*

***Abstract*—This paper presents a real-time crime detection system leveraging video surveillance technology to identify and classify critical incidents such as weapon detection, accidents, and explosions. The system processes video input from a webcam using advanced machine learning techniques to detect suspicious activities accurately. Upon detection, the system captures a snapshot of the event and sends an alert through the Telegram messaging platform, ensuring immediate notification to relevant authorities. By automating crime detection and alerting, the proposed system minimizes human monitoring requirements and enhances public safety through rapid incident response. This research contributes to the growing need for intelligent surveillance systems capable of addressing real-world security challenges effectively.**

***Index Terms*—Real-time crime detection, video surveillance, machine learning, weapon detection, accident detection, explo- sion detection, Telegram alerts, intelligent surveillance.**

I INTRODUCTION

The rapid advancements in technology have significantly enhanced the capabilities of video surveillance systems, making them indispensable tools for maintaining security. However, traditional systems often rely on continuous human monitoring, which is prone to errors and inefficiencies.

This paper proposes a real-time crime detection system that automates the identification of critical incidents, such as weapons, accidents, and explosions, through live video feeds. By integrating machine learning algorithms with a webcam input, the system analyzes video frames to detect suspicious activities accurately. Upon detection, immediate alerts are sent to relevant authorities via Telegram, along with snapshots of the incident.

The proposed solution aims to reduce the dependence on manual monitoring, improve response times, and contribute to public safety. This research focuses on the development,

implementation, and evaluation of the system's accuracy and reliability under various conditions.

II LITERATURE REVIEW

Recent research has focused on real-time crime detection systems for surveillance using machine learning and deep learning techniques. These approaches typically analyze video data captured by surveillance cameras to identify suspicious activities.

Machine learning algorithms, such as decision trees, sup- port vector machines, and random forests, are widely em- ployed for crime prediction and detection. These models are effective in classifying various criminal activities based on patterns in video data. Deep learning techniques, particularly Convolutional Neural Networks (CNNs), have shown promise in detecting objects and actions in surveillance footage, en- abling more accurate and automated crime detection.

In the context of edge devices, many systems are being developed to process surveillance data locally, reducing la- tency and computational burden. This is particularly useful for real-time detection in areas with limited resources. Another challenge is detecting specific events, such as weapons or violence, within video frames. Deep learning models trained for such tasks are increasingly being used to improve system accuracy.

Overall, the integration of machine learning, deep learning, and edge computing has significantly enhanced the perfor- mance of crime detection systems. However, issues such as data privacy, real-time processing, and environmental adapt- ability remain as challenges to be addressed in future research.

III SYSTEM DESIGN AND ARCHITECTURE

The proposed system is designed to detect real-time crimes or critical events such as the presence of weapons, accidents,
and explosions using AI-based models. The architecture con- sists of three main components:

III-A Overview of System Components

- Input: Live video feed captured using a webcam or a connected surveillance camera.
- Processing Unit: An AI-based detection model analyzes the video feed in real-time. Models such as YOLO (You Only Look Once) and SlowFast are employed to identify specific objects and actions.
- Output: Alerts are sent via the Telegram API, including an image of the detected event annotated with bounding boxes or relevant information.

III-B Flowchart or Block Diagram

The system architecture consists of the following steps:

1) Video feed is continuously captured from the surveil- lance camera or webcam.
2) Captured frames undergo preprocessing, including re- sizing and normalization.
3) Detection models analyze the frames to identify events such as weapons, accidents, or explosions.
4) If an event is detected:
   - Annotate the frame with bounding boxes or labels.
   - Trigger the alert mechanism to send an annotated frame and a message via Telegram.
5) If no event is detected, the system continues monitoring.

The figure 1 illustrates the sequential steps involved in the Real-Time Crime Detection System, showcasing its method- ology for efficient detection and reporting of crimes.

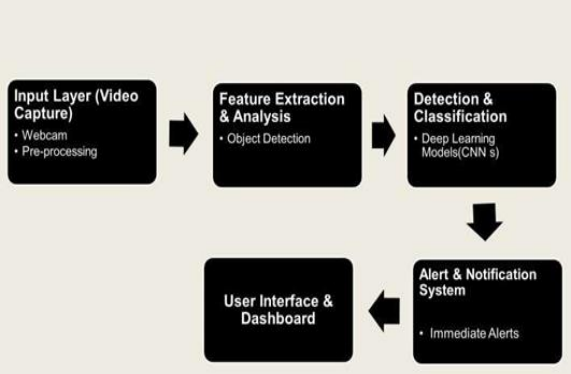

Fig. 1. Flowchart of the Real-Time Crime Detection System.

IV METHODOLOGY

The methodology involves several stages, from dataset preparation to training and implementing the detection mech- anism. These are elaborated as follows:

IV-A Dataset Preparation

- Datasets Used: The system leverages pre-trained mod- els on publicly available datasets such as COCO and Kinetics for object and action detection. Additionally, a custom dataset specific to weapons, accidents, and explosions was created by curating and labeling video samples.
- Labeling Process: Tools like LabelImg and custom scripts were used for labeling. The key classes include:
  - *Weapons:* Bounding boxes were annotated around objects such as firearms and knives.
  - *Accidents:* Motion patterns or collisions were la- beled frame-by-frame.
  - *Explosions:* Bright flashes and smoke patterns were identified and labeled.

IV-B Model Development

- Algorithms and Techniques:
  - *YOLO (You Only Look Once):* A fast and accurate object detection model used for real-time identifica- tion of weapons.
  - *SlowFast Network:* Used for action recognition by capturing fast and slow-motion patterns, ideal for detecting accidents and explosions.
  - *Inflated 3D CNN (I3D):* Employed for spatiotempo- ral feature extraction in video sequences, improving dynamic event detection.
- Model Selection Justification: YOLO is highly efficient for real-time detection, SlowFast excels at capturing temporal dependencies, and I3D ensures comprehensive video analysis for accurate action

recognition.

IV-C Training Process

- Preprocessing:
  - Frames were extracted from videos at 25 fps using OpenCV.
  - Images were resized to 224x224 resolution and normalized to a range of [0, 1].
  - Data augmentation techniques like flipping, rotation, and brightness variation were applied to enhance dataset diversity.
- Training and Validation: The dataset was split into 80% training and 20% validation. Cross-entropy loss was optimized during training, with early stopping employed to avoid overfitting.

IV-D Crime Detection Mechanism

- Weapon Detection: YOLO detects weapons such as guns and knives in the video feed, highlighting them with bounding boxes.
- Accident Detection: SlowFast analyzes collision pat- terns or abnormal activity, identifying accidents in real- time.
- Explosion Detection: I3D detects sudden brightness or fire patterns indicative of explosions through spatiotem- poral analysis.

IV-E Alert System

- Telegram API Integration: A Telegram bot is config- ured to send real-time alerts. Alerts include:
  - A message describing the detected event.
  - An annotated image of the detected frame.
- Image Capture and Annotation: Detected event frames are saved using OpenCV and annotated with bounding boxes or action labels before being sent via Telegram.

V IMPLEMENTATION

V-A Data Preprocessing and Input Prepara- tion

The implementation begins with preparing the video dataset. Videos are split into frames, and the Videoto3D class is used to extract a specified number of frames (depth) from each video. These frames are resized to a fixed width and height and normalized to scale pixel values between 0 and 1. This step ensures uniformity in the input data, a critical factor for neural network training. Each video is represented as a 3D array with a shape of (depth, height, width, channels).

For weapon detection, YOLOv5s processes individual video frames, annotating bounding boxes around detected weapons. This data is augmented and preprocessed to main- tain a consistent format, enhancing the performance of YOLOv5s during training.

V-B Model Architecture

V- B1 3D CNN for Spatiotemporal Analysis

The 3D CNN model is tailored to analyze video sequences by capturing spatiotemporal patterns. The architecture com- prises:

1) Convolutional Layers: This extract spatial and tem- poral features using 3D convolution filters. Batch nor- malization stabilizes and accelerates training, while ReLU activation introduces non-linearity.
2) Pooling Layers: MaxPooling3D or AveragePooling3D layers downsample feature maps, reducing complexity and focusing on salient features.
3) Fully Connected Layers: The classification task is performed by fully connected layers. The output layer uses a softmax activation function to predict class prob- abilities (e.g., weapon, accident, explosion, or normal).
4)

V-B2 YOLOv5s for Real-Time Weapon Detection

YOLOv5s is integrated to complement the 3D CNN by detecting weapons in individual frames. Built on CSPNet (Cross-Stage Partial Network), YOLOv5s enables efficient feature extraction with minimal computational overhead. Its lightweight structure supports real-time inference, making it ideal for dynamic environments.

V-C Training Process

The dataset is divided into training and validation sets using train_test_split from sklearn. The 3D CNN is trained using categorical cross-entropy loss and optimized

with Adam or SGD optimizers. Dropout layers mitigate overfitting. Simultaneously, YOLOv5s is trained on anno- tated datasets containing weapon labels, leveraging its built- in mechanisms for handling bounding box regression and classification loss.

Both models are trained in parallel:

- The 3D CNN focuses on classifying the overall

scene (e.g., detecting accidents or explosions).
- YOLOv5s focuses on detecting and localizing weapons within video frames.

V-D Evaluation and Visualization
- 3D CNN Evaluation: Training history, including accu- racy and loss for training and validation data, is plotted using the plot_history function. A CSV logger records training metrics for further analysis.
- YOLOv5s Evaluation: Metrics such as precision, recall, and mean Average Precision (mAP) are analyzed. Detec- tion results are visualized with bounding boxes drawn around identified weapons in the frames.

Figures 2 and 3 show the accuracy and loss during 3D CNN training.

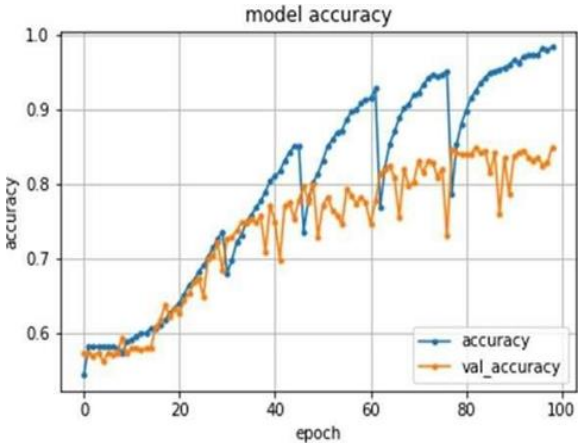

Fig. 2. Model Accuracy for 3D CNN during training and validation.

V-E Model Checkpoints

A ModelCheckpoint callback saves the best-performing 3D CNN model based on validation accuracy. YOLOv5s checkpoints are managed within its integrated training pipeline, saving weights at specified intervals.

V-F Deployment

The trained 3D CNN and YOLOv5s models are deployed for real-time detection:

1) Frame Processing: Video frames are extracted and passed to YOLOv5s for weapon detection. Detected weapons are highlighted with bounding boxes.
2) Sequence Analysis: The 3D CNN processes video sequences to classify scenes as accidents, explosions, or normal.

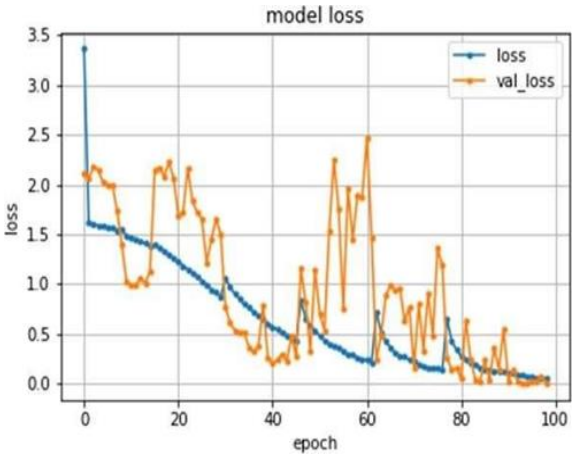

Fig. 3. Model Loss for 3D CNN during training and validation.

3) Integration: Outputs from both models are combined to provide comprehensive results. For instance, if YOLOv5s detects a weapon and the 3D CNN identifies an accident, the system can raise a high-priority alert.

This dual-model approach ensures robust and efficient detection, suitable for applications like surveillance, security, and forensic analysis.

VI RESULTS AND ANALYSIS

The proposed system effectively detects weapons, acci- dents, and explosions in videos by combining a 3D CNN for spatiotemporal analysis and YOLOv5s for precise weapon detection. Over 100 epochs of training, the 3D CNN demon- strated strong learning capabilities, with the training loss steadily decreasing and accuracy approaching near-perfect levels. Validation accuracy stabilized around 80%, indicating good generalization to unseen data, despite minor fluctuations in validation loss caused by dataset complexity.

YOLOv5s, leveraging transfer learning, achieved high pre- cision and recall, ensuring efficient weapon detection with minimal false positives. Its robust feature extraction capabil- ities, combined with lightweight architecture, made it highly effective in real-time detection scenarios.

The integrated framework excels in recognizing events by capturing motion patterns and spatial changes, supported by YOLOv5s' ability to detect

localized objects. Some over- fitting was observed, highlighting the need for potential improvements such as data augmentation or regularization techniques.

Figure 4 illustrates weapon detection using YOLOv5s in a real-time scenario, while Figure 5 demonstrates explosion detection using a 3D CNN with spatiotemporal analysis.

VII CONCLUSION

The proposed dual-model framework effectively addresses the challenges of analyzing video sequences for weapon de- tection and event classification. By integrating a 3D CNN for

Fig. 4. Weapon detection using YOLOv5s in a real-time scenario.

Fig. 5. Explosion detection using 3D CNN with spatiotemporal analysis.

spatiotemporal analysis and YOLOv5s for real-time weapon detection, the system demonstrates strong performance in identifying critical scenarios such as accidents, explosions, and weapon presence.

The experimental results show that the 3D CNN achieves high accuracy in scene classification, while YOLOv5s de- livers precise object detection with minimal false positives. The combination of these models ensures robust detection capabilities, making the framework suitable for applications in surveillance, security, and forensic analysis.Despite its strengths, the system encounters limitations such as overfitting and sensitivity to dataset complexity. These challenges can be mitigated in future work through enhanced data augmentation, hyperparameter tuning, and the incorpo- ration of attention mechanisms to refine temporal analysis. Expanding the dataset and exploring advanced architectures, such as transformer-based models, may further improve the system's performance and scalability.

In summary, this research presents a practical and efficient solution for video-based event detection, with significant potential for real-world deployment in dynamic and high-risk environments.

REFERENCES

[1] Sai Vishwanath Venkatesh, Adithya Prem Anand, Gokul Sahar S., Akshay Ramakrishnan, Vineeth Vija- yaraghavan," Real-time Surveillance-based Crime De- tection for Edge Devices," Conference Paper, 2024.

[2] Md. Mukto, M. Hasan, M.M. Al Mahmud, I. Haque, M.A. Ahmed, T. Jabid, M.S. Ali, M.R. Ahmmad Rashid, M. Islam, M. Islam," Design of a Real-Time Crime Monitoring System Using Deep Learning Tech- niques," Intelligent Systems with Applications, 2024.

[3] R. Ganesan, Dr. Suban Ravichandran," Performance Analysis for Crime Prediction and Detection Using Machine Learning Algorithms," IJISAE, 2024.

[4] Tufail Sajjad Shah Hashmi, Nazeef Ul Haq, Muhammad Moazam Fraz, Muhammad Shahzad," Application of Deep Learning for Weapons Detection in Surveillance Videos," 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), IEEE, 2021.

[5] Varun Mandalapu, Lavanya Elluri, Piyush Vyas, Nir- malya Roy," Crime Prediction Using Machine Learning and Deep Learning: A Systematic Review and Future Directions," IEEE Access, 2023.