As per UGC guidelines an electronic bar code is provided to secure your paper

International Journal for Modern Trends in Science and Technology
Volume 11, Issue 07, pages 72-79
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol11issue07.html
DOI: https://doi.org/10.5281/zenodo.15779023

AI Enabled Real Time Crime Detection and Alert System using IoT Cameras

M M Junitha | Kambhampati Deepthi

Department of Electronics and Communication Engineering, NRI Institute of Technology (A), Guntur, Andhra Pradesh, India.

To Cite this Article
M M Junitha & Kambhampati Deepthi (2025). AI Enabled Real Time Crime Detection and Alert System using IoT Cameras. International Journal for Modern Trends in Science and Technology, 11(07), 72-79. https://doi.org/10.5281/zenodo.15779023

Article Info
Received: 07 June 2025; Accepted: 26 June 2025.; Published: 30 June 2025.

KEYWORDS

IoT, Facial Recognition, Smart Security, Criminal Detection, Machine Learning, Real-Time Surveillance.

ABSTRACT

This project presents a comprehensive AI-based security system that leverages advanced deep learning techniques for real-time object detection and crime scene analysis. The system is designed to enhance surveillance capabilities, ensure public safety, and enable immediate response to potential threats. It employs the YOLOv8 architecture, trained on custom datasets, for accurate detection of weapons such as knives and guns, as well as recognition of criminal and violent activities including chain snatching, fighting, kidnapping, and accidents. The system processes live video feeds using a laptop, which runs the YOLOv8 models in real time. Upon detection of any suspicious activity or weapon, the system triggers alerts through two channels:

It sends immediate Telegram notifications to law enforcement authorities or concerned personnel, along with the captured image evidence. It activates a local alert system at the CCTV site using an Arduino Uno, Wi-Fi module, LCD, and buzzer to warn nearby public and deter further escalation. This dual-alert mechanism significantly improves situational awareness, crime prevention, and rapid response in public areas and sensitive environments.

1. INTRODUCTION

Urban crime such as theft, violence, and illegal weapon possession is rising, while traditional CCTV systems remain manual and error-prone. To enhance safety, we propose a smart surveillance system using YOLOv8 for real-time detection of weapons and crimes. It sends instant alerts via Telegram and triggers on-site notifications using an Arduino-based module, offering quick response and increased public awareness through deep learning and IoT integration.

1.1 Problem Statement

Conventional surveillance systems depend on human monitoring, often failing to detect crimes in real time. This delay in response can lead to unprevented incidents. There's a clear need for an AI-based system that detects threats like weapons and violent actions automatically and provides immediate alerts to both law enforcement and nearby individuals.

1.2 Main Objectives

1. Use YOLOv8 for real-time detection of weapons and crimes.
2. Process live CCTV video using a laptop or edge device.
3. Send instant Telegram alerts with images.
4. Activate Arduino-based public alert system (LCD, buzzer).
5. Deploy a scalable, low-cost solution for various public areas.

1.3 Literature Survey

Past research highlights the evolution of object detection from traditional methods to CNN-based models like YOLO. While early systems lacked real-time performance or integration with alert mechanisms, recent works show the potential of combining deep learning with messaging platforms like Telegram. However, real-world deployment with local IoT alerts remains limited.

1.4 Existing Methods and Drawbacks

- Traditional CCTV: Manual, slow, error-prone.
- Rule-based systems: Inflexible and outdated.
- Classical ML (SVM, HOG): Poor generalization, not real-time.
- Basic CNNs: Heavy and inefficient for small object detection.
- YOLOv4 and earlier: Good, but lacks YOLOv8's speed and accuracy.
- Alert-only systems: No local feedback; delayed response.

II. Proposed System

2.1 Proposed System

The system integrates YOLOv8 and IoT to detect weapons and violent actions in public spaces. A laptop connected to CCTV runs YOLOv8 to identify threats like knives, guns, or fights. When a threat is detected:

- A Telegram alert with an annotated image is sent to authorities.
- An Arduino-based local module with Wi-Fi, buzzer, and LCD issues on-site alerts.

This dual-alert system improves responsiveness and is suitable for locations like banks, schools, and public areas.

2.2 Proposed System Modules

1. Video Input Module: Captures live feed from CCTV or webcam.
2. Detection Module (YOLOv8): Identifies weapons and violence in real time.
3. Telegram Alert Module: Sends detection snapshot and message instantly to law enforcement.
4. Arduino Alert Module: Activates buzzer and LCD at the site when a threat is found.
5. Data Logging (Optional): Saves incidents for review and training.

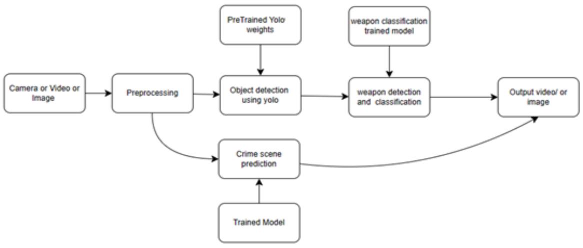


Fig-2.1 Block diagram of crime scene detection module

2.3 Implementation Methodology

1. Data Collection: Custom dataset of weapons and crime actions is labeled and used to train YOLOv8.
2. Detection: Real-time video processed via OpenCV and passed to YOLOv8.
3. Telegram Bot: Python script sends alerts using bot APIs.
4. IoT Alerts: Arduino module receives detection signal and triggers buzzer and display.
5. Deployment: Tested under real-world conditions for responsiveness.

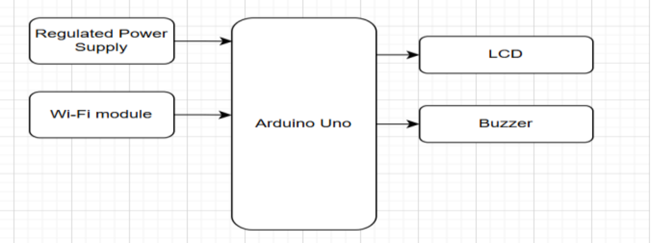


Fig 2.2 block diagram of alert system

2.4 Tools Used

- **Hardware:**
- CCTV/Webcam
- Laptop (YOLOv8 host)

- Arduino Uno + ESP8266
- LCD (16x2), Buzzer
- Power Supply, Wires
- **Software:**
- YOLOv8 (Ultralytics)
- Python, OpenCV
- Telegram Bot API
- Arduino IDE
- Roboflow (Annotation)
- Jupyter Notebook

2.5 Deep Learning Algorithms

Custom datasets were collected, annotated, and preprocessed for YOLOv8. Data is split into train/validation sets. YOLOv8 extracts features using its backbone, applies single-stage detection with multi-scale fusion, and is fine-tuned for this task.

2.6 Why YOLOv8?

- **High Accuracy & Speed**: State-of-the-art detection even on low-power devices.
- **Efficient & Real-time**: Optimized for fast inference.
- **Single-Stage Detection**: Reduces delay in detection.
- **Flexible & Open-source**: Easily customizable and community supported.

2.7 Model Training

Model is trained using annotated datasets, optimized with Adam/SGD, and evaluated using mAP. Trained weights are saved for inference. Prediction module processes input frames to show bounding boxes for weapons or threats.

2.8 Analysis & Prediction

- **Evaluation**: Uses precision, recall, F1-score, and confusion matrix.
- **Prediction**: Real-time inference from camera input.
- **Test Accuracy**: Final metrics assessed to ensure model reliability.
- **Model Saving**: YOLOv8 model stored for deployment.

2.9 Advantages and Applications

Advantages:

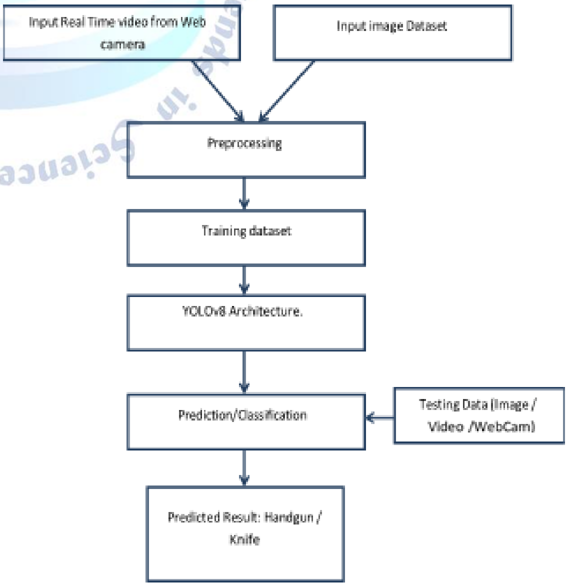
- Real-time detection with YOLOv8
- Dual alert system (Telegram + On-site)
- Low cost, scalable design
- High accuracy with custom dataset
- Automated 24/7 surveillance
- Expandable for new threats
- **Applications:**
- Public streets, ATM centers, schools
- Transport hubs, hospitals, isolated areas
- Gated communities, emergency zones

III. *METHODOLOGY*

This chapter outlines the systematic approach adopted for implementing the proposed system using Arduino Uno and NodeMCU ESP8266. The system is designed to provide real-time interaction and communication between sensors, output peripherals, and wireless communication modules to achieve the desired functionality. The steps in the methodology are structured to cover the integration, programming, and communication aspects of each hardware component involved in the system.

3.1 System Overview

The primary goal of this system is to collect real-time input from sensors or trigger mechanisms, process the data using microcontrollers (Arduino Uno or NodeMCU), and deliver output via display (LCD) or audible indicators (buzzer). Additionally, wireless transmission of critical data is facilitated through the NodeMCU ESP8266 for IoT-based monitoring.


Input Real Time video from Web camera
Input image Dataset
Preprocessing
Training dataset
YOLOv8 Architecture.
Prediction/Classification
Testing Data (Image / Video /WebCam)
Predicted Result: Handgun / Knife

3.2 Arduino Uno Integration

- The Arduino Uno acts as the central processing unit for sensor data acquisition and controlling actuators.

- It is programmed using the Arduino IDE with C/C++ based instructions to handle digital/analog I/O operations.
- It processes input signals (e.g., motion detection or button press) and triggers outputs such as the buzzer or LCD.
- Pin mapping and current specifications are strictly followed to avoid overloading the board.

3.3 NodeMCU ESP8266 Communication

- NodeMCU is used for transmitting data to the cloud or a remote monitoring dashboard over Wi-Fi.
- It is programmed via the Arduino IDE, selecting the ESP8266 board for compatibility.
- It handles HTTP/HTTPS or MQTT protocols to send messages or alerts to a server or smartphone app.
- GPIO pins are utilized for simple control or feedback signals, and analog pin A0 is used if required for voltage sensing.

3.4 16×2 LCD Display Usage

- The LCD 16×2 module is used to display real-time system status or alerts (e.g., “INTRUDER ALERT”, “SYSTEM ARMED”).
- It is interfaced in 4-bit mode to reduce the number of I/O pins used on the microcontroller.
- Commands and data are sent through digital pins using the LiquidCrystal library.
- Contrast is adjusted using a 10kΩ potentiometer connected to the V0 pin.

3.5 Buzzer Integration

- An active buzzer is used for audio alerts in the system.
- It is connected to one of the digital output pins of Arduino.
- The buzzer is activated using digitalWrite() logic when specific conditions are met (e.g., detection event or error).
- The buzzer alerts users with a continuous beep to indicate critical events.

3.6 Power Supply Management

- Arduino is powered via USB or a 9V DC adapter through the power jack.
- NodeMCU is powered via a micro-USB or 3.3V regulated input.
- LCD and buzzer are powered via the Arduino 5V and GND lines, considering the maximum current draw to ensure safe operation.

3.7 Programming and Control Logic

- The logic is implemented using conditional programming structures in Arduino IDE.
- Interrupts (e.g., external interrupts on pins 2 and 3) are used to capture real-time triggering events like motion detection or button press.
- Serial communication is used for debugging and status updates.
- Wi-Fi credentials and server endpoints are securely stored in the ESP8266 codebase.

3.8 Communication Protocol

- The system supports serial (UART) communication between Arduino and NodeMCU for basic handshaking or control.
- Wi-Fi communication is used for remote notification through HTTP requests or MQTT messages.
- LED and buzzer feedback is provided locally, while LCD shows status and message updates.

3.9 Final Integration and Testing

- All components are mounted on a breadboard or PCB for prototype testing.
- Power supply, wiring, and code logic are verified.
- Various test cases (e.g., alert triggering, message transmission, LCD updates) are executed to validate system functionality.
- The complete system is enclosed in a casing for safety and usability.

IV. RESULTS AND DISCUSSION

The proposed YOLOv8-based Real-Time Surveillance System was successfully implemented and evaluated on both static images and live video streams. The system is designed for the accurate detection of criminal and hazardous activities, including weapon presence (e.g., guns, knives), chain snatching, physical fights, kidnapping, and road accidents. It also integrates AI-based recognition with IoT-based local alert mechanisms.

4.1 Model Training and Performance

The YOLOv8 object detection model was trained on a custom-labeled dataset covering four core classes relevant to urban surveillance. After 50 epochs of training:

mAP\@IoU=0.5: 94.6%

mAP@\[0.5:0.95]: 76.3%

Training environment: RTX-enabled GPU laptop

Real-time detection speed: \~23 FPS

These results indicate robust object localization and classification performance, even across varying object sizes, occlusions, and lighting conditions.

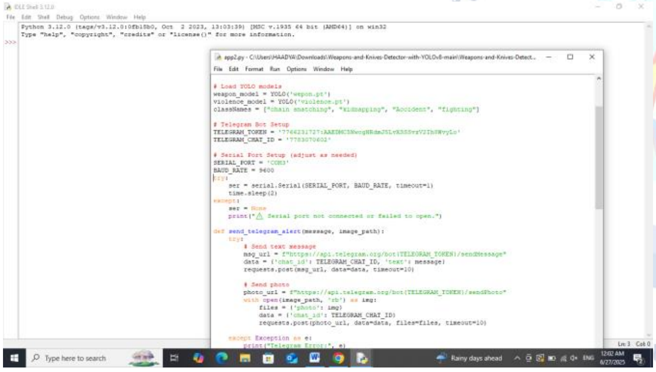
4.2 Real-Time Object Detection

The model was deployed for real-time surveillance using a webcam input. It reliably detected criminal actions with high accuracy and minimal latency. Visual outputs included:

Bounding boxes

Class labels (e.g., “Gun”, “Knife”, “Fight”)

Confidence scores (e.g., 97.2%)



```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

app2.py - C:\Users\HAADYA\Downloads\Weapons-and-Knives-Detector-with-YOLOv8-main\Weapons-and-Knives-Detect...

```
# Load YOLO models
weapon_model = YOLO('wepon.pt')
violence_model = YOLO('violence.pt')
classNames = ["chain snatching", "kidnapping", "Accident", "fighting"]

# Telegram Bot Setup
TELEGRAM_TOKEN = '7766231727:AAEDMC5NwogNRdmJSLvXSSSvzV2IhSWvyLo'
TELEGRAM_CHAT_ID = '7783070602'

# Serial Port Setup (adjust as needed)
SERIAL_PORT = 'COM3'
BAUD_RATE = 9600
try:
    ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
    time.sleep(2)
except:
    ser = None
    print("⚠ Serial port not connected or failed to open.")

def send_telegram_alert(message, image_path):
    try:
        # Send text message
        msg_url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendMessage"
        data = {'chat_id': TELEGRAM_CHAT_ID, 'text': message}
        requests.post(msg_url, data=data, timeout=10)

        # Send photo
        photo_url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendPhoto"
        with open(image_path, 'rb') as img:
            files = {'photo': img}
            data = {'chat_id': TELEGRAM_CHAT_ID}
            requests.post(photo_url, data=data, files=files, timeout=10)

    except Exception as e:
        print("Telegram Error:", e)
```

IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help
File Edit Format Run Options Window Help
Type here to search
Rainy days ahead
ENG
12:02 AM
6/27/2025
Ln: 3

Figure 1: Detection of a person holding a gun with accurate bounding box and label.

Figure 2: Successful detection of a chain-snatching attempt from test surveillance footage.

4.3 Telegram-Based Remote Alerts

Upon identifying a potential threat, the system automatically:

Captured a snapshot of the frame

Generated an alert containing:

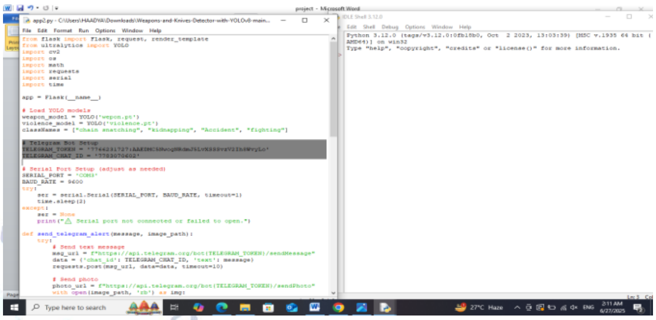
Detected object or activity

Confidence score

Timestamp

Annotated image

Sent the alert to a Telegram bot, notifying predefined recipients



app2.py - C:\Users\HAADYA\Downloads\Weapons-and-Knives-Detector-with-YOLOv8-main...

```
from flask import Flask, request, render_template
from ultralytics import YOLO
import cv2
import os
import math
import requests
import serial
import time

app = Flask(__name__)

# Load YOLO models
weapon_model = YOLO('wepon.pt')
violence_model = YOLO('violence.pt')
classNames = ["chain snatching", "kidnapping", "Accident", "fighting"]

# Telegram Bot Setup
TELEGRAM_TOKEN = '7766231727:AAEDMC5NwogNRdmJSLvXSSSvzV2IhSWvyLo'
TELEGRAM_CHAT_ID = '7783070602'

# Serial Port Setup (adjust as needed)
SERIAL_PORT = 'COM3'
BAUD_RATE = 9600
try:
    ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
    time.sleep(2)
except:
    ser = None
    print("⚠ Serial port not connected or failed to open.")

def send_telegram_alert(message, image_path):
    try:
        # Send text message
        msg_url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendMessage"
        data = {'chat_id': TELEGRAM_CHAT_ID, 'text': message}
        requests.post(msg_url, data=data, timeout=10)

        # Send photo
        photo_url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendPhoto"
        with open(image_path, 'rb') as img:
```

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
```

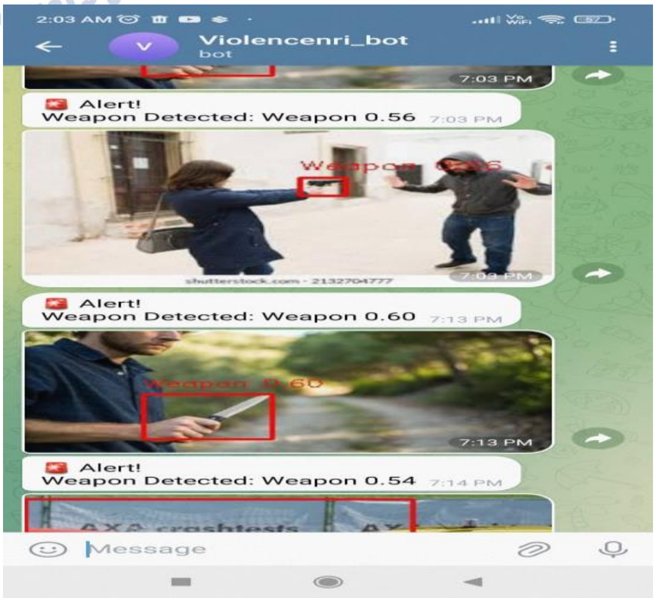
Figure 3: Telegram alert image showing “Gun Detected” label.

4.4 Local IoT-Based Warning System

A parallel IoT alert system was deployed to raise on-site awareness:

Arduino Uno served as the microcontroller

ESP8266 Wi-Fi module received detection signals from the main system



Violencenri_bot
bot

7:03 PM

Alert!
Weapon Detected: Weapon 0.56 7:03 PM

7:03 PM

Alert!
Weapon Detected: Weapon 0.60 7:13 PM

7:13 PM

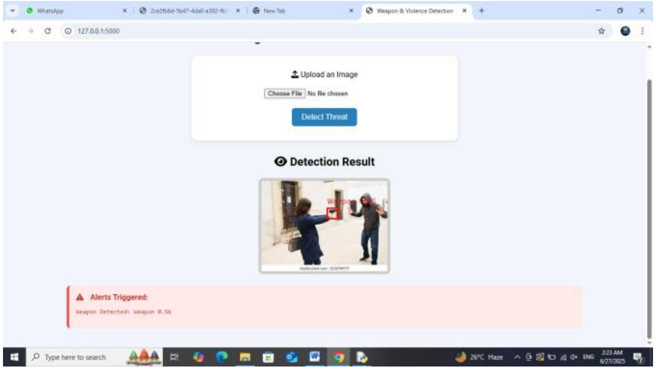
Alert!
Weapon Detected: Weapon 0.54 7:14 PM

Figure 4: Telegram notification for "Fighting Detected" with the attached real-time snapshot.

16×2 LCD display showed real-time event messages

Buzzer provided audible alert to nearby individuals

This setup allowed immediate crowd notification even without internet or smartphone access.



Alerts Triggered:

Weapon Detected: Weapon 0.56

Figure 5: LCD screen showing the message “Weapon Detected: Knife”.

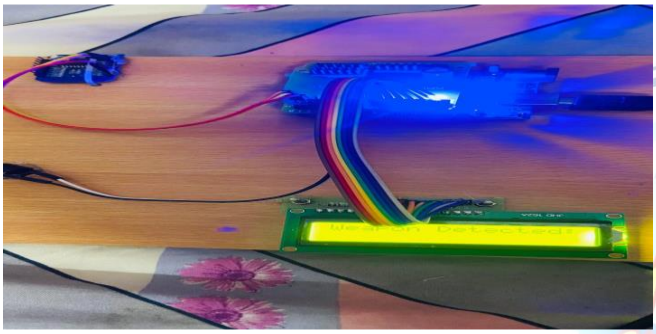


Figure 6: Arduino system in active alert state with LCD and buzzer triggered.

4.5 System Integration and Functional Validation

The combined system exhibited:

Seamless communication between AI model and hardware

Reliable dual-mode notification (remote + local)

Efficient response time under real-world test conditions

High classification accuracy for critical activities

4.6 Summary of Results

Metric / Feature	Result
mAP\@0.5	94.6%
mAP@\[0.5:0.95	76.3%
Real-Time FPS	~23
Telegram Alerts	<2 sec delay with annotated images
On-Device Alert Time (IoT)	Immediate (within 1 second)
Local Alert Mediums	LCD display, buzzer
Hardware Used	Arduino Uno, ESP8266, LCD 16×2, Buzzer

These results confirm the efficacy of the proposed system as a scalable real-time surveillance solution that merges AI-based vision capabilities with practical IoT-based responses for urban safety enhancement.

V. CONCLUSION AND FUTURE ENHANCEMENTS

The proposed intelligent surveillance system successfully integrates YOLOv8 deep learning algorithms with IoT-based alert mechanisms to enable real-time detection of weapons and criminal activities such as chain snatching, fighting, kidnapping, and road accidents. Leveraging a custom-trained YOLOv8 model, the system processes live video feeds from webcams or CCTV sources and rapidly identifies potential threats with high accuracy and low latency.

Upon detection of suspicious or violent activities, the system performs dual-mode alerting:

- **Remote Notification**: Annotated images and event details (including class label, confidence score, and timestamp) are sent automatically to law enforcement or predefined recipients via a **Telegram bot**.
- **Local Public Alert**: An **Arduino-based IoT module** equipped with a **buzzer** and **16×2 LCD display** is triggered to alert nearby individuals with real-time messages like “Weapon Detected: Knife” or “Crime Alert: Fighting”.

The system achieved a mean Average Precision (mAP) of 94.6% at IoU=0.5, and a mAP@[0.5:0.95] of 76.3%, confirming its strong capability to detect threats under various environmental conditions including partial occlusion and low light. With real-time video processing at ~23 FPS on an RTX-powered system, the solution demonstrates practical deployment readiness.

Overall, the system successfully combines the power of AI-based computer vision, wireless communication, and microcontroller-based alerting, offering a scalable and effective tool for urban safety, especially in high-risk zones like ATM centers, school premises, public roads, and residential areas.

5.1 Future Enhancements

To further improve scalability, adaptability, and functionality, the following enhancements are proposed for future iterations of the system:

1. Multi-Camera Integration

Incorporating support for multiple camera feeds from various locations will enable centralized monitoring of broader regions and allow real-time detection across distributed zones.

- **Synchronization**:
  - o *Software Synchronization*: Suitable for static environments, this method uses software to align frames from different cameras.
  - o *Hardware Synchronization*: For dynamic or fast-moving objects, precise frame-level alignment is critical. Hardware triggers such as **PWM-based signals** can initiate frame capture simultaneously across multiple cameras.
- **Applications**:
  - o Robotics (e.g., stereo vision using parallax)
  - o Automated surveillance of large-scale infrastructures
  - o Autonomous vehicles and sports analytics
- **Camera Interface Considerations**:
  - o For high-resolution and high-frame-rate requirements, interfaces like MIPI, GMSL2, or FPD-Link III are recommended over USB due to better bandwidth, stability, and longer-range data transfer capabilities.

2. Face Recognition and Criminal Identification

Integrating face recognition capabilities will allow real-time identification of known offenders by comparing detected faces with a criminal database.

- **Use Cases**:
  - o Automatic flagging of repeat offenders
  - o Identification from CCTV footage
  - o Cross-referencing individuals from multiple crime scenes

Face recognition adds an additional layer of biometric intelligence to the system, enabling more personalized and proactive intervention in criminal detection.

5.2 Summary of Proposed Enhancements

Enhancement	**Objective**	**Technology Involved**
Multi-Camera System	Broaden surveillance coverage	Hardware/software sync, camera interfacing
Face Recognition Module	Identify suspects from criminal database	FaceNet, OpenCV, Dlib, YOLO + facial features
Improved Camera Interfaces	Enhance resolution and real-time frame accuracy	MIPI, GMSL2, FPD-Link III

These enhancements will further position the proposed system as a next-generation smart surveillance platform, capable of not only detecting but also predicting and identifying criminal behaviors in real time.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] M Mudgal, D Punj, A Pillai, "Theoretical and Empirical Analysis of Crime Data", Journal of Web Engineering (Vol:20, Issue: 1, Jan 2021)

[2] M. T. Bhatti, M. G. Khan, M. Aslam, and M. J. Fiaz, "Weapon detection in real-time CCTV videos using deep learning," IEEE Access, vol. 9, pp. 34366–34382, 2021, doi: 10.1109/ACCESS.2021.3059170.

[3] Smith, John; Johnson, Mary, "Real-Time Surveillance for Crime Detection Using Deep Learning", IEEE Transactions on Image Processing, 2020

[4] L Abdul Saleem; E Venkateswara Reddy, "A Survey on Deep Learning based Video Surveillance Framework", 2023 International Conference on Computer Communication and Informatics 2023, DOI: 10.1109/ICCCI56745.2023.10128302 5.

[5] Abdul Rehman; Labiba Gillani Fahad, "Real-Time Detection of Knives and Firearms using Deep Learning", 2022 24th International Multitopic Conference (INMIC), DOI: 10.1109/INMIC56986.2022.9972915

[6] V. Mandalapu et al.: Crime Prediction Using ML and DL:A Systematic Review and Future Directions

[7] Chhaya Gupta, Nasib Singh Gill, "A Real Time 3- Dimensional Object Detection using Human Action Recognition Model", IEEE Open Journal of the Computer Society (Vol: 5), 2022, DOI: 10.1109/OJCS.2023.3334528

[8] A Singh; T Anand; S Sharma; P Singh, "IoT Based Weapons Detection System for Surveillance and Security Using YOLOV4", 2021 6th International Conference on Communication and Electronics Systems, DOI: 10.1109/ICCES51350.2021.9489224

[9] P Akshaya; P B Reddy; P Panuganti; P. Gurusai; A Subhahan, " Automatic weapon detection using Deep Learning", 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering. DOI: 10.1109/RMKMATE59243.2023.10369889

[10] Rakesh Garg; Someet Singh, "Intelligent Video Surveillance Based on YOLO: A Comparative Study.", 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), 978-1-6654-2634-3,DOI: 10.1109/ICAC353642.2021.9697321

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convo lutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), May 2016, pp. 779–788. [Online].

[13] S. W. Shah and S. S. Kanhere, "Recent trends in user authentication— A survey," IEEE Access, vol. 7, pp.112505–112519, 2019, doi: 10.1109/ACCESS.2019.2932400.

[14] K. M. Renuka, S. Kumari, D. Zhao, and L. Li, "Designof a secure password-based au thentication scheme for M2M networks in IoT enabled cyber-physical systems," IEEE Access, vol. 7, pp. 51014-51027, 2019, doi:10.1109/ACCESS.2019.2908499.

[15] H.-J. Mun, "Biometric Information and OTP based on authentication mechanism using blockchain", J. Converg.Inf.Technol., vol. 8, no. 3, pp. 85-90,2018, doi: 10.22156/CS4SMB.2018.8.2.085.