Intelligent Video Surveillance using Deep Learning

Abhishek Sunil Sarjine[1], Pramod R. Jadhao[2].

[1]Master of Computer Application, Trinity Academy of Engineering, Pune.

[2]Assistant Professor (Dept of MCA), Trinity Academy of Engineering, Pune.

ABSTRACT

In recent years, the demand for automated and intelligent surveillance systems has increased significantly due to rising concerns over public safety and security. Traditional surveillance systems rely heavily on manual monitoring, which is time-consuming, error-prone, and lacks real-time responsiveness. This research presents the development of an intelligent video surveillance system using deep learning techniques, specifically employing the YOLOv5s object detection model for efficient and accurate real-time detection of objects such as people, vehicles, and animals. The system is implemented as a web-based application using Flask, with functionalities for both live surveillance through a webcam and video upload analysis. The YOLOv5s model ensures a balanced trade-off between speed and detection accuracy, enabling smooth operation even on limited hardware. The application also incorporates user authentication and object counting, offering a complete end-to-end solution. Experimental results demonstrate that the system performs well in diverse environments, making it suitable for use in schools, offices, and public places. This project highlights the effectiveness of deep learning in enhancing modern surveillance solutions.

Keywords:- Deep Learning, YOLO (You Only Look Once), Object Detection, Real-Time Tracking, Flask Web Application, Object Counting, Automated Surveillance System

1. INTRODUCTION

Video surveillance has become a crucial tool in ensuring safety and security across a variety of public and private domains. However, traditional surveillance systems rely heavily on manual monitoring by human operators, which is not only inefficient but also prone to fatigue and oversight. As the volume of video data increases, the limitations of manual surveillance become more evident, necessitating the need for intelligent and automated systems.With advancements in artificial intelligence, particularly deep learning, there has been a significant shift toward automated surveillance solutions. Object detection models such as YOLO (You Only Look Once) offer real-time performance and high accuracy, making them ideal for smart surveillance systems. These models can detect and classify multiple objects in a single video frame, enabling proactive responses to security threats.

This research focuses on developing an intelligent video surveillance system using the YOLOv5s model integrated with a Flask-based web application. The system supports both live webcam monitoring and video file upload, with features such as real-time detection, object annotation, and user authentication. Designed for simplicity and efficiency, this system demonstrates how deep learning can enhance surveillance by reducing human workload and improving response time in real-world scenarios.

2. LITERATURE SURVEY / BACKGROUND

The evolution of video surveillance has seen a transition from analog systems to IP-based digital solutions, but most traditional systems still rely on manual monitoring and rule-based video analytics. These methods often suffer from limited scalability, delayed responses, and reduced accuracy due to human fatigue. Recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs), have significantly improved object detection and scene understanding. Among the most influential models is YOLO (You Only Look Once), known for its balance between speed and accuracy. The YOLOv5 series, developed using PyTorch, offers various versions (s, m, l, x) suited for different resource constraints and performance needs. YOLOv5s, for instance, provides faster inference with lightweight computation, making it ideal for real-time surveillance on modest hardware. Prior works have integrated YOLO models into surveillance applications for tasks such as pedestrian detection, vehicle tracking, and abnormal activity recognition. However, many of these implementations lack a user-friendly interface or real-time streaming support.

Our project bridges this gap by combining YOLOv5s with a Flask web framework to deliver a responsive, intelligent surveillance platform that supports both live feed and uploaded video analysis. This integration addresses both usability and performance, creating a more practical solution for real-world surveillance challenges.

3. PROPOSED WORK / SYSTEM

The proposed system is an intelligent, real-time video surveillance application that integrates deep learning for automated object detection and monitoring. Built using the YOLOv5s object detection model, the system is capable of processing both live and uploaded video streams, identifying objects such as persons, bicycles, and vehicles. It includes secure admin authentication, video input management, object annotation and counting, and real-time result display through a user-friendly web interface.

The entire architecture is developed using Python, Flask, OpenCV, and SQLite, making it lightweight, fast, and easy to deploy on modest hardware. The modular design allows for feature scalability and future improvements, including facial recognition, behavioral analysis, and suspicious activity detection.

Key System Modules:

1. Authentication Module – Verifies admin login credentials and manages session state securely.
2. Forgot Username/Password – Allows account recovery via security questions stored in the database.
3. Video Input Module – Accepts real-time webcam feeds and uploaded video files.
4. YOLOv5 Detection Engine – Performs frame-by-frame object detection with high accuracy and speed.
5. Object Count Module – Maintains count of detected objects and updates it in real time.
6. Logging & Analytics Module – Stores detection logs and statistics for review and reporting.
7. Web Interface – Built using Flask, HTML/CSS, and Bootstrap for responsive admin control.
8. Data Storage – SQLite database manages user credentials, recovery info, and detection logs.

Additional System Characteristics:

1. Works in real-time with frame rates ranging from 20–30 FPS depending on hardware.

2. Lightweight backend with no need for high-end GPU (but GPU accelerates detection).
3. Supports scalability: the modular nature makes it ready for multi-camera support and cloud deployment.
4. Security: user access is restricted to verified admins, and recovery features ensure account continuity.
5. Accessibility: web interface is accessible on any device through a browser, with responsive design.

Future Scope:

1. Add real-time alerts for unusual activity.
2. Integrate face recognition using a pretrained deep learning model.
3. Implement cloud logging and remote dashboard access.
4. Provide multi-user support with role-based access control (admin/viewer/operator).

4. RESULT AND DISCUSSIONS

The developed intelligent video surveillance system successfully detects and annotates objects in both live and pre-recorded video streams using the YOLOv5s model. The system demonstrates real-time processing capabilities, maintaining a balance between detection speed and accuracy, which is essential for surveillance applications. During testing, the system was evaluated under various lighting conditions and environments. It consistently identified common objects such as people, vehicles, and bags with minimal false positives. The average processing speed was found to be satisfactory, enabling live surveillance without noticeable lag on standard hardware configurations.

Additionally, the user interface, built with Flask and Bootstrap, provides a smooth and interactive experience. Features such as login authentication and recovery mechanisms enhance the security and usability of the system.

The results confirm that the system is efficient for real-world deployment, though the accuracy could be further improved by training on domain-specific datasets or using more advanced YOLO models such as YOLOv5m or YOLOv8x depending on the hardware support.

5. CONCLUSION

The development of an intelligent video surveillance system using deep learning represents a significant advancement in the field of security and automated monitoring. This project successfully demonstrates how convolutional neural networks and object detection algorithms, particularly YOLOv5s, can be integrated into a real-time surveillance application that offers both efficiency and accuracy. The system provides robust functionalities, including live video monitoring, pre-recorded video analysis, object detection, and object counting, all within a user-friendly web interface.

Throughout the project, we focused on balancing detection accuracy with processing speed to ensure that the system operates smoothly without sacrificing performance. YOLOv5s, being lightweight and fast, served as a practical choice for real-time inference, although it was observed that detection accuracy might slightly drop in complex scenes. This trade-off was considered acceptable for general surveillance purposes, but future implementations may adopt more advanced variants or ensemble techniques for critical applications.

The system is designed with simplicity and security in mind, featuring basic authentication and recovery modules. While it currently supports only an admin user, the modular structure allows for future expansion into multi-user environments with access control levels. The integration of Flask for the backend and OpenCV for video handling proved effective in delivering a scalable and responsive web-based application.

REFERECNES

[1] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” arXiv preprint arXiv:1804.02767, 2018.

[2] G. Jocher et al., “YOLOv5 by Ultralytics,” GitHub repository, 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[3] A. Rosebrock, *Deep Learning for Computer Vision with Python*, PyImageSearch, 2017.

[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[6] OpenCV, “Open Source Computer Vision Library,” [Online]. Available: https://opencv.org

[7] Flask Documentation, “Flask Web Framework,” [Online]. Available: https://flask.palletsprojects.com/