CCTVIntelliGuard: An Intelligent Edge-Based Human Motion Detection and Alerting System for CCTV Surveillance

V. Nithiyaraj
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
vithursanaa.22@cse.mrt.ac.lk

S.J. Sooriyaarachchi
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
sulochanas@cse.mrt.ac.lk

I. ABSTRACT

Conventional CCTV (Closed-Circuit Television) surveillance systems capture continuous video streams without intelligence, resulting in storage overhead, delayed incident response, and inability to differentiate human activity from irrelevant motion. This paper presents CCTV IntelliGuard, a lightweight edge-based module that enhances traditional CCTV systems with real-time human motion detection and logging. The system employs a hybrid approach combining OpenCV based motion detection with YOLOv8 (You Only Look Once) object detection model [1] , deployed on a Raspberry Pi 5 edge device. Upon detecting human activity and the system records timestamped video clips. This retrofit solution transforms conventional CCTV into intelligent surveillance with minimal additional infrastructure. And this is optimized computation through two-stage detection.

II. INTRODUCTION

Security surveillance has become a critical component in safeguarding institutions, workplaces, and public spaces. Existing CCTV systems act as passive recorders, generating large volumes of video that require manual inspection. This causes storage overhead, delayed recognition of incidents, and reduced efficiency in responding to threats. Modern intelligent surveillance can address these limitations by providing real-time detection, classification, and alerting. However, most AI-powered systems require high performance GPUs or cloud connectivity, restricting adoption in low-cost deployments. CCTV IntelliGuard offers a retrofit solution by combining lightweight classical motion detection with optimized YOLOv8 inference, enabling cost-effective and resource efficient intelligence on Raspberry Pi 5 edge devices.

III. LITERATURE REVIEW

Traditional motion detection techniques such as frame differencing and background subtraction are computationally inexpensive but highly sensitive to lighting changes, weather, and environmental noise. This often results in false alarms [2]. Deep learning approaches (YOLO, Faster R-CNN, SSD) achieve superior accuracy but require GPUs or servers. Edge deployments face challenges of limited memory, CPU capacity, and energy budgets [3].

To overcome these limitations, hybrid frameworks that combine classical motion detection with deep learning inference have been proposed [4]. In such pipelines, motion detection serves as a trigger, activating the more computationally intensive human detection model only when relevant activity occurs. This approach reduces unnecessary processing, lowers power consumption, and enables deployment on resource-constrained edge devices. Intelli Guard implements this two-stage strategy, leveraging the efficiency of motion detection and the accuracy of YOLOv8 [1] to provide reliable real-time human monitoring.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Hardware Setup:

The prototype uses existing CCTV cameras and a Raspberry Pi 5.

B. Software Components:

- **OpenCV (Open Source Computer Vision Library) for motion detection** – Uses background subtraction and frame differencing to detect movement.
- **YOLOv8 for human classification** – Runs only when motion is detected, confirming if the object is a person.
- **FFmpeg for video clipping** – Decodes the RTSP stream and saves event-based recordings through OpenCV's writer.

Fig. 1 Raspberry Pi 5 setup

ERU Symposium 2025 - DOI 10.31705/ERU.2025.26

C. *Workflow:*

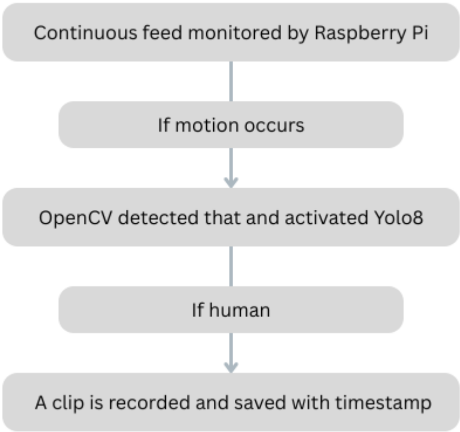

Fig.2 Workflow chart

V. Results and Discussion

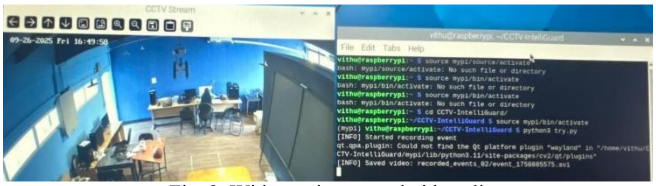

Fig. 3. With previous saved video clip

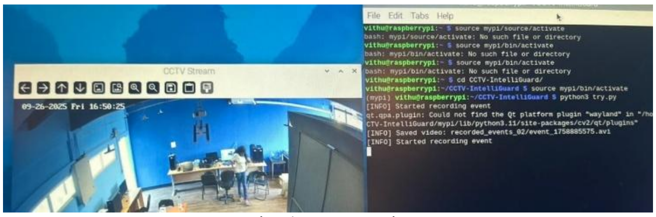

Fig. 4. Start moving

Fig. 5. Continue moving

Fig. 6. Continue moving

Fig. 7. Finish moving

Fig. 8. Save new video clip after the cooldown period

A. *Discussion*

As illustrated in Figure 3, the system initially records a video clip and then continues monitoring the feed without creating a new clip until motion is detected. In Figure 4, a human motion is observed, triggering the start of a new video recording. Subsequent frames in Figures 5 and 6 show continued movement, with the recording ongoing. Even after the subject stops moving, as seen in Figure 7, the system does not immediately stop recording due to a predefined cooldown period. This design accommodates scenarios where motion resumes shortly after stopping and prevents the creation of multiple short clips. Additionally, to ensure seamless capture, the system includes a few pre-event frames from a buffer , producing video clips (Figure 8) composed of pre-event frames, the detected event, and post-event frames.

B. *Challenges*

The system occasionally reports duplicate frames in the video sequence, indicating misordered or repeated frames due to encoding or stream inconsistencies. Intermittent frame drops are also observed, caused by slow frame reads, network latency, or system resource limitations. These issues can lead to inaccurate object detection, lag, and unreliable performance.

VI. Conclusion

CCTV IntelliGuard enhances passive CCTV systems with intelligent, edge-based human detection and real-time alerting. By combining OpenCV motion detection with optimized YOLOv8 inference, it delivers high accuracy with low resource consumption. This makes IntelliGuard a scalable, affordable solution for institutions seeking enhanced security without replacing infrastructure.

Acknowledgment

I thank the University of Moratuwa and the Department of Computer Science and Engineering for their support and guidance throughout this project. And I would like to express my sincere gratitude to Dr. (Ms.) S. J. Sooriyaarachchi, Eng. Prof. Chathura R. De Silva, and the senior students for their valuable guidance, and assistance.

References

[1] Yaseen, Muhammad. “What Is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector.” *ArXiv.org*, 2024, arxiv.org/abs/2408.15857.

[2] Smith, Sarah. “When Weather Defeats Surveillance: Why Most Security Systems Fail in Real-World Conditions.” *ArcadianAI*, 4 Aug. 2025, www.arcadian.ai/blogs/blogs/when-weather-defeats-surveillance-why-most-security-systems-fail-in-real-world-conditions?srsltid=AfmBOopYTINgXaQHLIahyR8AfKdcjqCS fHaQKyZgG0We2kdL-jEbTypv. Accessed 30 Sept. 2025.

[3] Vina, Abirami. “Ultralytics YOLO11 for Real-Time Security Monitoring | Ultralytics.” *Ultralytics.com*, 2025, www.ultralytics.com/blog/real-time-security-monitoring-with-ai-and-ultralytics-yolo11. Accessed 30 Sept. 2025.

[4] Kim, Chulyeon, et al. “A Hybrid Framework Combining Background Subtraction and Deep Neural Networks for Rapid Person Detection.” *Journal of Big Data*, vol. 5, no. 1, 10 July 2018, https://doi.org/10.1186/s40537-018-0131-x.