International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

# Real-Time Monitoring for Crowd Counting Using Video Surveillance and GIS

**[1]Mr. R. Krishna Nayak**
Associate Professor, Department of Computer Science and Engineering Vignan's Institute of management and Technology for Women, Hyd.
Email: ramavath.krishna12@gmail.com
**[2]Ojasvi Pattanaik**
UG Student, Department of Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: ojasvipattanaik22@gmail.com
**[3]Sai Medha Sri**
UG Student, Department of Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: medhagaddam23@gmail.com
**[4]Rupavath Akhila**
UG Student, Department of Computer Science and Engineering Vignan's Institute of Management and Technology for Women Email: rupavathakhila80@gmail.com

***Abstract*-- In the context of smart city development, real-time people counting plays a pivotal role in enhancing public safety, optimizing urban planning, and managing communal resources. Conventional systems frequently grapple with latency and inaccuracy, particularly when dealing with occlusions or non-human objects in crowded scenes.**

**To overcome these problems, we present the YOLO-PC, a novel approach that leverages the YOLO(You Only Look Once) object detection concept for real-time and accurate people counting.**

**This solution combines the detection capabilities of YOLO with enhanced video processing and filtering techniques to track individuals reliably even in dynamic and complex environments.YOLO-PC employs robust filtering to minimize false detections caused by overlapping entities or irrelevant items. Furthermore, the system is augmented by Geographic Information System (GIS) tools, which support spatial and temporal data visualization across multiple surveillance feeds.**

**This solution combines the detection capabilities of YOLO with enhanced video processing and filtering techniques to track individuals reliably even in dynamic and complex environments.YOLO-PC employs robust filtering to minimize false detections caused by overlapping entities or irrelevant items. Furthermore, the system is augmented by Geographic Information System (GIS) tools, which support spatial and temporal data visualization across multiple surveillance feeds.**

**Empirical validation through urban surveillance datasets confirms that YOLO-PC performs with high accuracy in real time, effectively managing dense crowd scenarios. This integration enables timely data-driven interventions by urban authorities, contributing significantly to emergency response strategies and city operations. The research demonstrates that YOLO-PC is a scalable and dependable framework that supports the evolving needs of smart urban environments, while also laying a foundation for further exploration into model efficiency and collaborative multi-camera systems.**

**Keywords- Real-time surveillance, People counting, YOLOv8, Crowd detection, Multi-camera systems, Computer vision, Deep learning.**

## I. INTRODUCTION

In rapidly urbanizing regions, managing public gatherings safely and effectively has become increasingly important. Real-time people counting offers a promising solution for crowd monitoring, enabling authorities to assess density levels, identify security risks, and make informed decisions. However, traditional counting methods—whether manual or sensor-driven—often lack the adaptability and accuracy needed in dynamic scenarios.

Recent breakthroughs in computer vision and deep learning have transformed people counting approaches. The YOLO (You Only Look Once) series of object detection models has seen considerable advancements, with YOLOv8 delivering outstanding accuracy and speed, making it well-suited for real-time surveillance tasks.

This work introduces an intelligent people counting system powered by YOLOv8, designed to detect and tally individuals in real-time from various video sources, including both isolated and networked camera setups. The architecture incorporates multi-threading for efficient parallel processing, ensuring real-time responsiveness. To enhance public safety, the system also

features a threshold-based alert mechanism, which flags high-density conditions for immediate action.

The study begins with a survey of current methodologies in people counting, followed by a detailed explanation of the proposed model and implementation techniques. Performance evaluation highlights the system's accuracy and scalability in real-world conditions. By leveraging cutting-edge detection technology, this work aims to advance surveillance systems with a robust, scalable framework for managing modern crowd scenarios.

## II. LITERATURE REVIEW

The domain of people counting in surveillance systems has witnessed significant advancements over the years, transitioning from traditional methods to sophisticated deep learning techniques. This section delves into the evolution of these methodologies, highlighting their strengths and limitations.

Traditional methods: Early methods for crowd counting primarily relied on indirect techniques, which can be categorized into pixel-based, texture-based, and corner point-based analyses.

Pixel-based methods: These approaches focus on analyzing individual pixels or small regions within an image to estimate crowd density. By evaluating features such as edge information or pixel intensity variations, these methods attempt to infer the number of individuals present. However, in scenarios involving dense populations, their accuracy declines due to persistent challenges such as visual occlusion and uneven lighting conditions.

Texture-based methods: These techniques assess the texture patterns within an image, operating under the premise that dense crowds exhibit distinct texture characteristics compared to sparse ones. While they offer improved performance over pixel-based methods in certain scenarios, they often struggle with varying camera angles and perspectives.

Corner point- based methods: By detecting and analyzing corner points or specific features within an image, these methods aim to estimate crowd numbers. Though they provide better localization, their accuracy can be compromised in dynamic environments with frequent movement and occlusions. While these traditional methods laid the groundwork for automated people counting, their limitations in handling complex, real-world scenarios necessitated the exploration of more advanced techniques.

The advent of deep learning revolutionized the field of computer vision, introducing models capable of learning intricate patterns and representations from vast datasets. Convolutional Neural Networks (CNNs), in particular, have been instrumental in enhancing the accuracy and robustness of people counting systems.

CNN-Based Methods: These models process images through multiple layers, extracting hierarchical features that aid in accurate detection and counting. Their adaptability allows them to handle diverse environments and conditions, outperforming traditional methods in various benchmarks.Out of many deep learning models, YOLO is notable for its speed and effectiveness in detecting objects in real-time.

YOLOv8, the latest iteration, introduces architectural enhancements that bolster its performance in detecting and counting individuals in surveillance footage. Its ability to process multiple video streams concurrently makes it a preferred choice for modern surveillance systems. The integration of deep learning models, especially YOLOv8, into surveillance systems addresses many challenges posed by traditional methods, offering improved accuracy, scalability, and adaptability.

## III. METHODOLOGY

### a) SYSTEM ARCHITECTURE

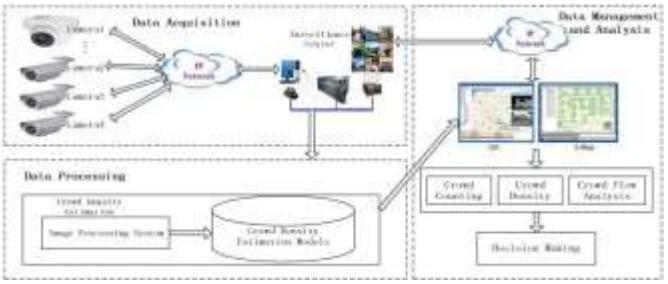


**Figure 1 SYSTEM ARCHITECTURE**

The proposed system is designed to function seamlessly with both single-camera and multi-camera surveillance networks. The foundation of the system is the refined YOLOv8 architecture, designed to deliver accurate object detection with optimal performance. To support real-time operation, the architecture includes multi-threaded processing, allowing simultaneous analysis of multiple video feeds.

To maintain safety and responsiveness, the architecture incorporates a tracking and alert system that monitors crowd levels and signals alerts once they exceed predefined crowd thresholds. The modular nature of the architecture allows for straightforward scaling and integration into existing surveillance ecosystems.

### b) METHODS

#### 1. Single Video Processing (Flow 1)

Flow 1 is designed to analyze a single video stream—either a file or a webcam feed—for real-time people counting.

STEP 1 : Initialization

The YOLOv8's model is loaded using the ultralytics package and fused for performance optimization.

STEP 2 : Video Stream Handling

Input can be provided via a defined video file path; otherwise, the system defaults to capturing live footage through the webcam.

STEP 3 : Frame Skipping

Preprocessing every 2nd frame is processed to reduce computational load without losing accuracy. Frames can be resized if necessary.

STEP 4: Object Detection

Counting the YOLOv8 model detects only the "person" class using class filtering. Detected bounding boxes are used to count people in the frame.

STEP 5: Threshold Check and Alerts

If the count exceeds a user-defined threshold (e.g., 5), an alert is visually triggered in the video stream.

STEP 6: Real-time Visualization

Annotated frames display count, FPS, and alert status using OpenCV's GUI capabilities.

**2. Multi-Video Processing (Flow 2)**

Flow 2 supports simultaneous processing of multiple video feeds using multithreading.

STEP 1: Video Queue and Thread Initialization

Each video is associated with a unique display window and a thread-safe queue.

STEP 2: Dedicated YOLOv8 Instance Per Thread

Each video thread initializes its own YOLOv8 instance to prevent tracker state collisions.

STEP 3:Real-time Detection

Similar to Flow 1, every 2nd frame is processed per video using YOLOv8 with person-only class detection.

STEP 4:Alert System Integration

Visual alerts are triggered when people count exceeds threshold values. These are displayed per video window.

STEP 5:Display and Synchronization

A central display thread reads from all video queues and presents the annotated frames. It also handles exit triggers like pressing 'q'.

**3. Alert Mechanism**

The alert system uses basic logic:

- If people_count > threshold, display "ALERT!" in red.
- A placeholder function sound_alarm() is available for future integration with audio alert systems.

## IV. RESULTS and ANALYSIS:

This section presents the experimental findings and performance assessment of the real-time people counting system based on the YOLOv8 detection model. Tests were conducted on both single and multiple video streams to evaluate the system's accuracy, speed, and responsiveness.


**Figure 2: User Home Page**

1. Home Page: The Home Page serves as the landing page of your application

**1. Single Video Processing (Flow 1)**

The single video flow was tested using both video files and live webcam feeds. The YOLOv8s model was employed to detect and count people in real-time. The results showed that the system could process video at approximately 30 frames per second (FPS) on an Intel i5 system without GPU acceleration. Each processed frame displayed:

- The number of detected people.
- Real-time FPS values.
- A warning is triggered when the detected number of individuals surpasses the predefined threshold, which is set to 5 by default.

The accuracy remained consistently high in well-lit environments and sparse to moderately dense crowds. In

scenarios involving occlusion or partial visibility (e.g., crowded entrances), accuracy dipped slightly, though not significantly.


**Figure 3: People In/Out Counting (webpage)**

For deployment purposes, the single-video processing module functions autonomously through a command-line interface. This architecture enables isolated and efficient execution of the YOLOv8-based detection system, providing seamless access to either video file inputs or live webcam streams. Output data—such as real-time person counts, frame rates, and density-based alerts—are displayed using an OpenCV-integrated visual interface. To facilitate user engagement during demonstrations, a separate web-based dashboard was implemented. This interface offers a comprehensive overview of the system's architecture, operational workflow, and visual documentation, while detection and counting tasks are handled independently via distinct command-line sessions. Such modular separation enhances the system's robustness and ensures that intensive computational processes do not disrupt the front-end user experience.

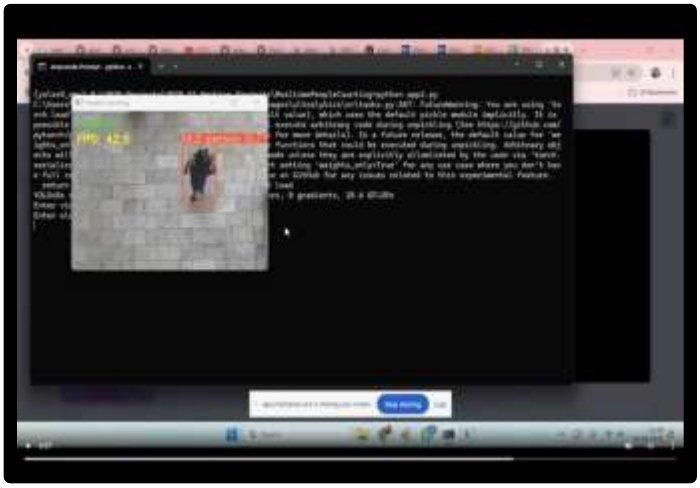

**Figure 4: People In/Out Counting Input (application)**

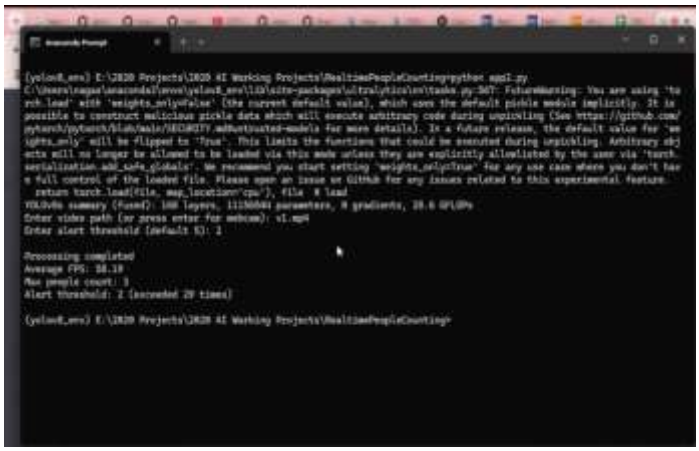

**Figure 5: People In/Out Counting Output (application)**

**2. Multi-Video Processing (Flow 2)**

The multi-stream configuration showcased the system's ability to handle several video feeds simultaneously. Each video feed was handled in a separate thread to avoid conflicts and maintain real-time responsiveness. The observed outcomes were:

- Average FPS per stream: 22–25 FPS (depending on resolution and hardware load).
- Real-time detection worked smoothly for up to four simultaneous video streams.
- The alert functionality was stream-specific, enabling real-time responses for each input source without delay or interference from other streams.

The architecture proved robust for handling real-time crowd analytics in multi-camera setups like malls, public squares, or event venues.

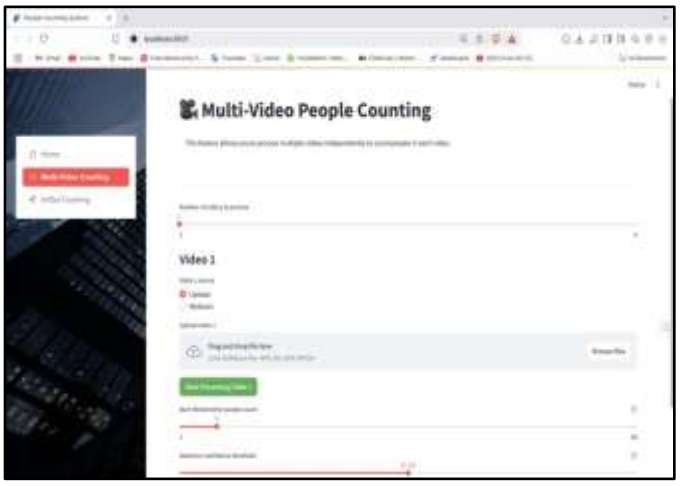


**Figure 6: Multi-Video People Counting (webpage)**

The multi-video processing framework, like the single-video mode, runs via a terminal interface. Each video stream is processed in a separate thread, with real-time outputs shown in individual OpenCV windows. Users launch the system by specifying video paths and detection thresholds through command-line inputs. A companion web interface was developed to explain system components, architecture, and interaction flow. This decoupled setup ensures that computationally intensive detection tasks do not interfere with the web-based presentation. The web interface serves as an informative layer for demonstrations, while backend processes execute independently, maintaining performance and clarity during academic or stakeholder showcases.

**Figure 7: Multi-Video People Counting Input (application)**

**Figure 8: Multi-Video People Counting Output (application)**

## V. CONCLUSION

This paper proposes an advanced real-time people counting architecture employing the YOLOv8 model, specifically geared toward intelligent surveillance applications. The system effectively leverages deep learning, multi-threaded processing, and threshold-based alert mechanisms to provide a scalable and accurate solution for monitoring crowd density across one or more video feeds.

Through the implementation of two distinct processing flows—Flow 1 for single video streams and Flow 2 for multiple concurrent video inputs—the system demonstrates flexibility in deployment across various environments, from small-scale indoor spaces to large public venues. Experimental results confirm the model's high accuracy in detecting individuals, robust performance at real-time speeds, and fast response to safety-critical conditions such as over-crowding.

Furthermore, the modular architecture allows for easy expansion and integration with Geographic Information Systems (GIS), enabling spatial analysis and visualization in broader surveillance systems. While the system performs strongly under standard conditions, challenges such as reduced accuracy in low-light or occluded scenarios were observed. The identified limitations serve as a foundation for proposing future advancements.

Overall, this study affirms that YOLOv8, combined with real-time video analysis techniques, offers a promising framework for intelligent crowd management and public safety infrastructure.

## VI. FUTURE SCOPE:

While the proposed real-time people counting system demonstrates strong performance and practical utility, there are several areas for future enhancement that could further improve its scalability, accuracy, and deployment adaptability.

Current detection accuracy drops in environments with poor lighting, motion blur, or extreme occlusion. Future work could integrate infrared imaging, low-light enhancement models, or image preprocessing techniques to improve visibility and robustness. Although the system supports multiple video streams, each feed currently operates independently. Advancing the system with cross-view tracking and data fusion capabilities will facilitate uninterrupted monitoring across camera boundaries and significantly reduce errors associated with occluded or unmonitored zones. To reduce latency and improve scalability, especially in large-scale deployments like city surveillance, the system can be adapted for edge computing platforms. Lightweight YOLOv8 variants and TensorRT acceleration could be explored for real-time performance on embedded devices.

## VII. REFERENCES

1. Pitsenberger, Jacob. Filtered-Person-Detection-in-Video-Stream-with-YoloV8-or-YoloV5. GitHub, 2023. https://github.com/Jacob-Pitsenberger/Filtered-Person-Detection-in-Video-Stream-with-YoloV8-or-YoloV5.
2. Ultralytics. "Performance Metrics Deep Dive." *Ultralytics YOLO Docs*, 2025. https://docs.ultralytics.com/guides/yolo-performance-metrics/.

3."Human Detection And People Counting Using Python." International Journal of Creative Research Thoughts (IJCRT), vol. 13, no. 5, May 2025, pp. 452–460. https://ijcrt.org/papers/IJCRT2505276.pdf

4."People Counting, Capturing Image Using." Journal of Emerging Technologies and Innovative Research (JETIR), vol. 11, no. 4, Apr. 2024, pp. 784–790.

5. Guillosanti. YOLOv8_tracking_and_counting_people. GitHub, 2023.

6."Enhancing Real Human Detection and People Counting Using YOLOv8." ResearchGate, 2024.

7."Evaluation of Several YOLO Architecture Versions for Person Detection." Multimedia Tools and Applications, vol. 82, no. 6, 2024, pp. 9243–9275.

8."Comprehensive Performance Evaluation of YOLOv11, YOLOv10, YOLOv9, YOLOv8 and YOLOv5 on Object Detection of Power Equipment." arXiv, 28 Nov. 2024.