Journal of Science, Computing and Engineering Research (JSCER)
Volume-9, Issue- 3, 2026.
DOI: https:jscer.org/doi.org/10.46379/jscer.20.26

AI BASED REAL-TIME OBJECT DETECTION AND COUNTING SYSTEM

[1] J. R. Arun Kumar [2] Lavish Saini [3] Hitesh Kumar [4] Menaka Sharma

[1] Professor, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India

[2,3,4] UG Students, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India

Article Information

Received :26/03/2026

Revised :26/03/2026

Accepted :27/03/2026

Published :29/03/2026

Corresponding Author:

Lavish Saini

***Abstract*—** In today's digital era, real-time monitoring systems play an important role in areas such as traffic management, security, and crowd control. This paper introduces an AI-based system for real-time object detection and counting using deep learning methods. The system uses the YOLOv8 model to identify different objects in video frames and applies a Kalman Filter to track their movement over time. A mobile application is developed using Flutter to provide a simple and user-friendly interface, while the backend is built using Python and FastAPI for efficient processing. The system is capable of handling both live video from a camera and recorded video files, delivering results instantly. WebSocket communication is used to maintain fast and continuous data exchange between the mobile application and the server. The experimental results show that the system performs well in detecting and counting objects with good accuracy and speed. Overall, the proposed system is useful for real-world applications that require quick, reliable, and automated monitoring.

Keywords: Object Detection; YOLOv8; Computer Vision; Deep Learning; Real-Time System; Object Counting

Citation: J. R. Arun Kumar, Lavish Saini, Hitesh Kumar, Menaka Sharma. "AI-based real-time object detection and counting system", Journal of Science, Computing and Engineering Research, 9(3), March 2026.

I. INTRODUCTION

In today's digital world, monitoring objects in real time has become essential for applications such as traffic management, public safety, crowd monitoring, retail analytics, and industrial automation. Traditional methods, including manual counting and simple sensor-based systems, often fail to meet modern requirements due to their slow processing, high cost, and limited scalability. Advances in artificial intelligence, particularly deep learning and computer vision, have enabled more accurate, automated, and real-time visual analysis. This section introduces the key concepts and significance of real-time object detection and counting.

A. Overview of Object Detection and Counting

Real-time object detection and counting is a crucial task in computer vision. It involves identifying and tracking objects in video feeds or live camera input, providing immediate information about their number and location. Traditional approaches relied on manual counting or rule-based systems, which were slow, prone to errors, and unsuitable for large or dynamic environments. The development of deep learning models, such as YOLO, has significantly improved detection speed and accuracy. With AI, it is now possible to monitor traffic, crowds, and other environments automatically, reducing the need for human supervision.

B. Importance and Applications

Real-time object detection and counting plays a vital role in improving safety, efficiency, and decision-making across various fields. Key applications include traffic monitoring to prevent congestion and accidents, crowd management in public events, surveillance for security purposes, and analytics in retail stores to optimize customer service and resource management. These systems allow organizations to respond quickly to dynamic situations and make informed, data-driven decisions. As technology advances, such systems are becoming more accessible, scalable, and essential for modern monitoring and automation tasks.

II. PROBLEM STATEMENT

In the modern digital era, accurate and real-time object detection and counting has become critical across areas such as traffic management, public safety, retail analytics, industrial monitoring, and smart surveillance. Rapid urbanization and growing population densities create a need for intelligent monitoring systems capable of delivering reliable, data-driven insights. Despite technological advancements, many existing solutions still face technical, operational, and economic limitations, particularly when applied in dynamic real-world environments or on mobile platforms.

A. Importance of Object Counting

Knowing the number and location of objects or people is essential for efficient monitoring and decision-making. Applications include traffic monitoring, crowd management, security surveillance, and retail analytics. Traditional approaches relied on manual counting or simple rule-based algorithms, which were slow, error-prone, and unsuitable for large or complex environments. Modern systems must be able to process video streams efficiently and provide real-time feedback to be effective in practical scenarios.

B. Challenges in Traditional Methods

- ***Limitations of Manual Monitoring and Counting***

  Manual observation is inefficient, labor-intensive, and prone to human error, especially in large-scale areas such as highways, shopping centers, or industrial facilities. Fatigue and inconsistency often result in inaccurate counts and delayed responses, making manual monitoring unsuitable for real-time applications.

- ***Inaccuracy in Conventional Automated Systems***

  Rule-based or non-AI automated systems often produce duplicate counts, missed detections, and false positives. They struggle with overlapping, fast-moving, or variably shaped objects and are heavily dependent on predefined thresholds, reducing adaptability in dynamic environments.

- ***Poor Adaptability to Real-World Conditions***

  Environmental factors such as lighting variations, weather changes, occlusions, and cluttered backgrounds can significantly reduce the accuracy of traditional systems. Fixed camera or sensor setups often fail to maintain reliable performance in such situations.

- ***Limited Real-Time Access***

  Many legacy systems are designed for desktop or server-based operation and lack mobile accessibility, limiting their use in field-level operations and real-time decision-making.

- ***High Infrastructure and Maintenance Costs***

  Conventional monitoring relies on specialized hardware, sensor networks, and complex installations, increasing deployment and maintenance costs and limiting adoption for smaller organizations.

- ***Missing AI Mobile Solution***

  There is a clear gap in systems that combine real-time deep learning detection, object tracking, mobile accessibility, and scalable backend processing into a single architecture. Current solutions rarely integrate advanced AI models with lightweight mobile deployment, which is essential for modern, portable monitoring.

C. Objective of the Proposed System

To address these challenges, this project aims to develop an AI-based mobile-enabled system for real-time object detection, tracking, and counting. It employs YOLOv8 for fast and accurate object detection and a Kalman Filter for tracking objects across video frames. The system works with live camera feeds or recorded videos and provides instant results. The primary goal is to deliver a reliable, efficient, and scalable solution that can operate effectively in real-world environments while being easy to deploy and maintain.

III. PROPOSED METHOD

The goal of this project is to create a real-time system that can detect, track, and count objects using deep learning and computer vision. The system works on video streams or live camera feeds. To do this, we use the YOLOv8 model for detecting objects and a Kalman Filter to track them across frames. A mobile app is built with Flutter, and the backend is implemented in Python with FastAPI. The results are shown in real time on the app using WebSocket communication.

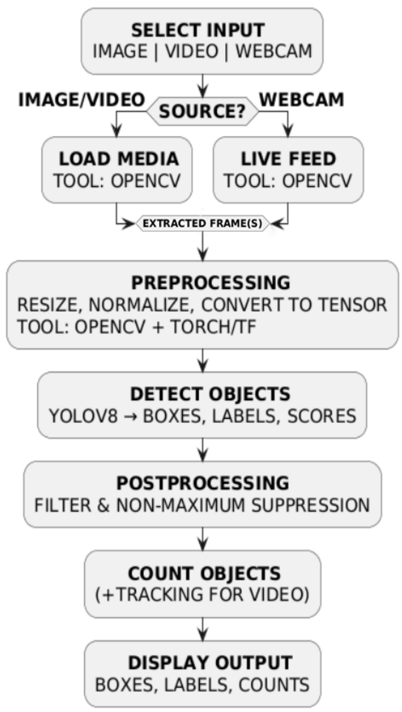


Fig. 1. Proposed System

A. System Overview

This system is designed to detect, track, and count objects in real time. YOLOv8 detects objects in each frame, and the Kalman Filter keeps track of them across multiple frames. The mobile app is built with Flutter, and the backend uses Python and FastAPI. The system can work with both live camera feeds and saved video files.

B. Dataset and Input

The system can process video streams from cameras or video files. For testing, we use sample videos containing objects like cars, people, and animals. Each video frame is resized and prepared before sending it to YOLOv8 to make sure the detection is accurate and consistent.

C. Model Architecture

YOLOv8 is a neural network that can detect many objects at the same time with good speed and accuracy. It predicts both the type of object and its position in the frame. After detection, the Kalman Filter tracks each object and gives it a unique ID. This combination makes the system reliable, even when there are many objects close together.

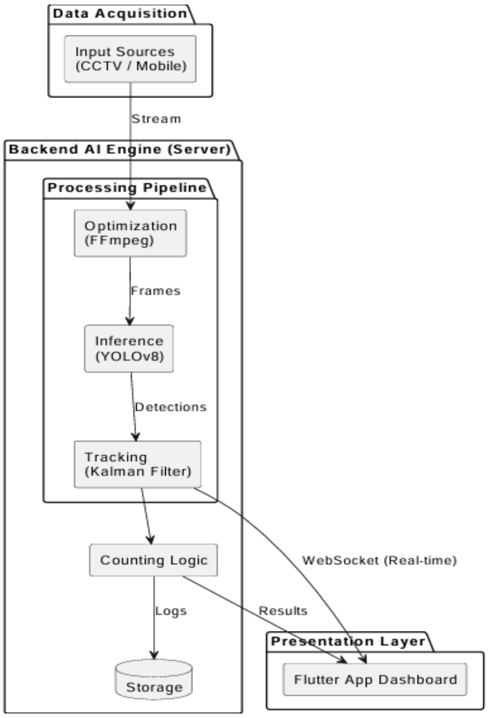


Fig. 2. Architecture Model

D. Object Detection

YOLOv8 looks at each frame and identifies objects by class and position. It is fast and accurate, which makes it suitable for real-time applications where multiple objects appear at once.

E. Object Tracking

The Kalman Filter keeps track of objects across frames, even if they move or disappear temporarily from the camera view. This prevents double counting and ensures every object is counted only once.

F. Counting Objects

Each tracked object gets a unique ID. The system automatically updates the total number of objects as they enter or leave the frame, providing a real-time count.

G. System Deployment

The backend processes all video frames and sends the results to the mobile app via WebSocket. The app shows the live video with bounding boxes, object IDs, and total counts in a simple and easy-to-understand interface.

IV. IMPLEMENTATION

The implementation of the real-time object detection and counting system involves preparing the dataset, preprocessing video frames, training and deploying the YOLOv8 and Kalman Filter models, and building a user-friendly mobile application.

A. Dataset Preparation

The system uses video streams or recorded video files containing objects like people, vehicles, and animals. The dataset is divided into training, validation, and testing sets. Each frame is extracted from videos and labelled with object classes and bounding boxes for supervised learning. Data augmentation techniques such as rotation, scaling, and flipping are applied to improve model robustness and generalization.

B. Preprocessing

Before feeding the video frames into YOLOv8, frames are resized to a fixed resolution and normalized to maintain consistency. Noise reduction and contrast adjustment are applied when needed. Preprocessing ensures that the model receives high-quality input and maintains detection accuracy in varying lighting conditions and environments.

C. Model Training

YOLOv8 is trained on the prepared dataset to detect multiple objects in each frame. The model learns to predict bounding boxes, object classes, and confidence scores simultaneously. Hyperparameters such as learning rate, batch size, and number of epochs are tuned for optimal performance. The Kalman Filter parameters are also configured to track objects effectively across consecutive frames, even when they temporarily disappear from view.

D. Backend Development

The backend is implemented using Python and FastAPI. It handles frame processing, object detection, tracking, and counting. Detection results, including bounding box coordinates and object IDs, are sent to the mobile application in real time via WebSocket communication.

E. Mobile Application

The mobile app is developed using Flutter. It receives real-time detection and counting results from the backend and displays them with an intuitive interface. Users can view the live feed with bounding boxes, object IDs, and the total count.

F. Technology Stack

- ***Frontend*:** Flutter for building the mobile interface.
- ***Machine Learning:*** YOLOv8 for object detection, Kalman Filter for tracking.
- ***Backend:*** Python and FastAPI for processing and communication.
- ***Video Processing:*** OpenCV for frame extraction and preprocessing.
- ***Data Handling:*** NumPy for efficient matrix operations and image manipulation.
- ***Real-Time Communication:*** WebSocket for sending detection results to the mobile app.

V. RESULTS

The proposed real-time object detection and counting system was tested using live camera feeds and recorded video datasets containing multiple objects such as cars, people, and animals. The results demonstrate that the system can detect, track, and count objects accurately and efficiently in real time.

A. Detection Accuracy

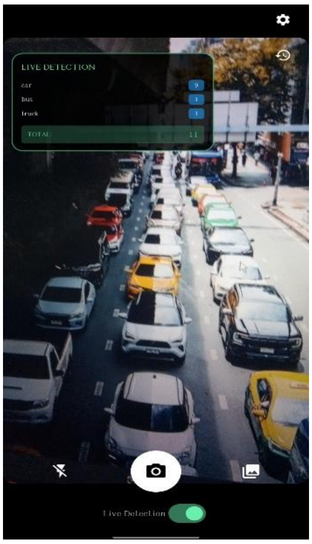
The YOLOv8 model successfully identified objects in each frame with high precision. Detected objects were classified correctly, and bounding boxes closely matched the object locations. Accuracy was measured using metrics showing consistent performance across different scenarios.

B. Tracking Performance

The Kalman Filter effectively maintained the identity of objects across frames. Even when objects moved quickly, partially occluded, or temporarily left the frame, the tracking algorithm preserved unique IDs, preventing duplicate counting.

C. Counting Results

The system automatically updated object counts in real time. When new objects entered the frame or existing objects exited, the total count was adjusted accurately. This demonstrates that the system is reliable for applications like traffic monitoring or crowd analysis.



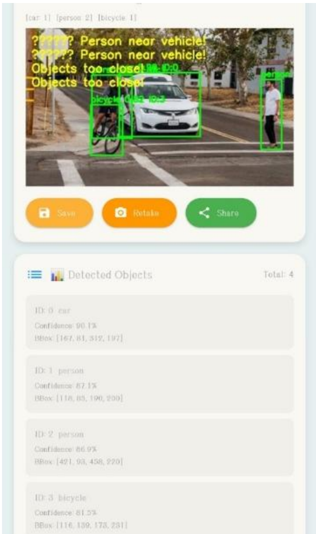


Fig. 3. Real-time Counting ***Fig. 4. Media detected Result***

D. User Interface Output

The mobile application displays the live video feed with overlaid bounding boxes for each detected object and the current total count. Users can easily observe real-time detection and tracking, and visually verify the correctness of the results.

E. Summary

Overall, the system achieved a balance of accuracy, speed, and real-time performance, making it suitable for practical deployment in surveillance, traffic monitoring, and retail analytics. Visual examples of detection and counting are shown in Figures 3 and 4, highlighting both the detected objects and their corresponding counts in real time.

VI. CONCLUSIONS

The real-time object detection and counting system developed in this project demonstrates reliable performance in identifying, tracking, and counting multiple objects in both live and recorded video streams. By leveraging YOLOv8 for detection and a Kalman Filter for tracking. The integration of a mobile application using Flutter, with a Python-FastAPI backend, ensures a seamless and user-friendly interface for monitoring results in real time.
This system is highly adaptable, making it suitable for applications such as traffic control, crowd monitoring, surveillance, and retail analytics. Overall, the project underscores the potential of combining deep learning and computer vision techniques to automate monitoring tasks and provide actionable insights across multiple domains.

Key Highlights:

A. Accurate Object Detection: Using YOLOv8, the system reliably detects multiple objects in real time with high precision, even in crowded or complex scenes.

B. Robust Tracking: The Kalman Filter ensures consistent tracking of objects across frames, maintaining unique IDs and avoiding duplicate counts.

C. Real-Time Counting: Objects are counted accurately as they appear or leave the frame, providing live quantitative insights for applications such as traffic monitoring, crowd management, and surveillance.

D. User-Friendly Deployment: The Flutter-based mobile app, connected via the FastAPI backend, allows real-time intuitive for end users.

E. Scalability and Future Enhancements: The system's lightweight and modular architecture supports future improvements, including higher-resolution input, advanced AI models, and alert-based monitoring.

REFERENCES

[1] A. Dirir, H. Ignatious, H. Elsayed, M. Khan, M. Adib, A. Mahmoud, and M. Al-Gunaid, "An efficient multi-object tracking and counting system using deep learning in urban vehicular environments," *Future Internet*, vol. 13, no. 12, 2021.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, realtime object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016.

[4] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2017.

[5] G. Jocher, et al., "YOLOv8: The latest generation of You Only Look Once real-time object detection," *Ultralytics*, 2023.

[6] L. Bochkovskiy, Y. Wang, and H. Liao, "Recent advances in YOLO object detection for embedded and mobile platforms," *Journal of Real-Time Image Processing*, vol. 20, 2024.