

AI-Based Smart Bus Passenger Counting and Alert System for Bus Capacity

Mr. Abhale.B. A¹, Miss. Aware.S. P², Miss. Babar.P. L³, Miss.Bhadane A. D⁴, Miss.Ugalmugale.S. B⁵
^{1,2,3,4,5}S.N. D College of engineering and research centre, yeola Savitribai Phule Pune University

Abstract—Effective management of public transportation is crucial for enhancing urban mobility, alleviating traffic congestion, and guaranteeing commuter safety. Conventional methods for counting passengers on buses, like manual monitoring or infrared sensors, often face issues such as inaccuracy, restricted coverage, and operational difficulties. This review paper explores AI-driven smart passenger counting technologies that leverage deep learning, video analysis, and IoT frameworks to track passenger entry and exit in real time. It emphasizes important methods including object detection models (like YOLO), tracking algorithms (such as OpenCV-based tracking), and embedded systems. Furthermore, this research investigates alert systems that inform drivers or authorities when bus capacity limits are surpassed, thereby improving operational efficiency and passenger safety, particularly in the context of post-pandemic smart cities. The review also addresses performance evaluation metrics, challenges such as occlusion and low visibility, and potential future research paths aimed at creating more resilient, privacy-conscious, and energy-efficient systems. In conclusion, AI-driven automated counting and alert systems offer a promising solution for optimizing public transport operations and improving the commuter experience.

Index Terms—Artificial Intelligence (AI), YOLO, Object Detection, Public Transportation, Real Time Tracking, Crowd Density Analysis, Alert System.

I. INTRODUCTION

Urban public transportation systems play a crucial role in modern infrastructure, providing an affordable and efficient means of mobility for millions of daily commuters. As cities grow and densities increase, ensuring safe and comfortable bus travel has become more challenging. One significant issue faced by transit authorities is the difficulty of accurately tracking passenger load, resulting in regular

overcrowding, diminished comfort, longer boarding times, and heightened operational strain on transit systems. Traditional methods such as manual head counting, ticket-based estimates, or infrared counter sensors are often unreliable, susceptible to errors, and incapable of delivering continuous real-time data, especially during peak periods or in highly variable environments.

Recent advancements in Artificial Intelligence (AI) and computer vision technology have led to the development of smart passenger counting systems that offer a novel solution to these issues. Deep learning frameworks such as Convolutional Neural Networks (CNNs), YOLO-based object detectors, and sophisticated tracking algorithms have markedly enhanced the precision and speed of human detection in real-world contexts. When combined with IoT devices and platforms like Arduino or specialized AI processors, these systems can effortlessly tally passengers boarding or alighting from a bus and consistently monitor occupancy levels. In addition to counting, AI-driven systems can also implement automated alert features that inform drivers or control units when a bus surpasses its seating or standing limits, thereby encouraging safer travel and improving fleet management.

This survey paper seeks to deliver a thorough examination of recent advances in AI-based bus passenger counting and alert systems. The review emphasizes vision-based methodologies, sensor-assisted hybrid systems, and IoT-integrated smart solutions that improve monitoring efficiency. Furthermore, the paper assesses existing research in relation to accuracy, computational efficiency, practicality of deployment, hardware limitations, and adaptability to various environmental factors such as occlusion, low-light situations, and differing camera perspectives. By synthesizing current research,

pinpointing gaps, and showcasing emerging trends, this study offers valuable perspectives for researchers and industry practitioners aiming to create robust and scalable intelligent passenger counting systems for smart public transportation networks.

II. PROBLEM DEFINITION

Public transportation systems often face issues with overcrowding and inefficient passenger flow due to the absence of precise real-time monitoring techniques. Conventional counting methods such as manual counting, ticket-based approximations, and simple sensor-based devices are generally unreliable, susceptible to human mistakes, and struggle to perform effectively during peak times or in crowded situations. Consequently, transport authorities encounter difficulties in maintaining passenger safety, ensuring comfort, overseeing fleet operations, and avoiding situations of overcapacity.

While several AI-based solutions have emerged in recent years, no standardized or universally recognized approach exists for real-time passenger counting and bus occupancy monitoring. The effectiveness, processing speed, hardware needs, and adaptability to real-world conditions such as obstructions, lighting changes, and movement within the vehicle vary significantly across different models and systems. As a result, there is a necessity to systematically evaluate and contrast existing AI-driven methodologies to pinpoint the most effective techniques, comprehend their shortcomings, and recommend potential improvements.

Public bus systems often struggle with overcrowding and poor passenger flow because traditional counting methods are manual, inaccurate, and not suitable for real-time monitoring. These outdated techniques lead to unsafe travel conditions, inefficient bus usage, and inconvenience for passengers. Although AI and computer-vision-based solutions are emerging, existing models still differ in precision, speed, cost, and performance in real-world environments like crowded or low-light buses. Therefore, there is a need to evaluate current AI-based passenger counting technologies, identify their drawbacks, and explore better techniques that ensure accurate, real-time, and scalable bus occupancy management for smart transportation.

This survey seeks to fill this research void by

reviewing current state-of-the-art passenger counting systems, assessing their performance, and identifying technological obstacles and future pathways for developing dependable, scalable, and real-time AI-based monitoring and alert systems for bus passengers.

III. LITERATURE SURVEY

Sr no	Paper title	Author name	year
1	Passenger Counting in Public Transport using IR Sensors	Sharma et al.	2021
2	Real-Time Bus Occupancy Monitoring using YOLO-based Computer Vision	Zhang et al.	2022
3	IoT-Based Smart Passenger Tracking System	Kumar & Reddy	2023
4	RFID-Based Passenger Identification and Counting	Lopez et al.	2023
5	Arduino-Ultrasonic Sensor Based Bus Overcrowding Alert System	Singh & Patel	2024
6	Automatic Passenger Counting System on Public Buses Using CNN: YOLOv8 Model for Passenger Capacity Optimization	Prastyo et al	2024
7	Development of a Real-Time Automatic Passenger Counting System	Hyunduk Kim, Myoung-Kyu Sohn, Sang-Heon Lee	2022
8	BPCS: Multi-view Bus Passenger Counting System	A. K. Mokin et al.	2024
9	A Review of Passenger Counting in Public Transport Concepts with Solution Proposal based on Image Processing & ML	A. Radovan	2024

IV. METHODOLOGY

The proposed system adopts a hybrid approach combining embedded sensing hardware with intelligent computer-vision software to accurately

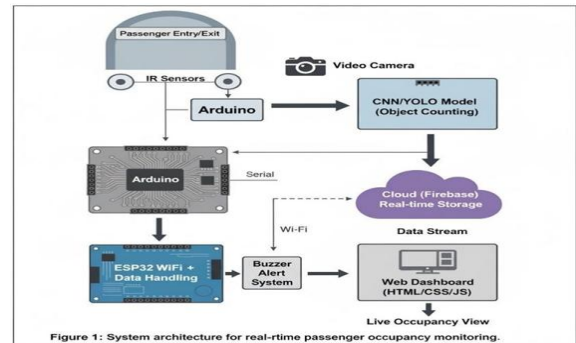
monitor passenger flow and detect overcrowding in public buses. The hardware module consists of an Arduino controller interfaced with IR sensors placed at the bus doors to count passenger entry and exit events. An ESP32 microcontroller is used for wireless data transmission and additional on-device processing. When the IR sensor detects a crossing, the system updates the passenger count and checks it against a preset threshold. If the maximum occupancy limit is reached, the buzzer is triggered to provide real-time alerts to the driver and passengers, ensuring safety and compliance with capacity norms.

Simultaneously, a vision-based passenger recognition model is deployed using CNN and YOLO architectures to enhance count reliability, particularly in crowded or rapid-movement scenarios. The captured video frames are processed to detect and track individuals, reducing false counts and improving system robustness. The processed data and occupancy values are uploaded to Firebase Cloud in real time, enabling secure storage, monitoring, and record keeping. An interactive web dashboard created using HTML, CSS, and JavaScript displays live bus occupancy status, alerts, and analytics for operators and users. This integrated architecture ensures accurate crowd tracking, automated alerts, seamless data access, and scalability for smart public transport systems.

The proposed system follows a structured methodology designed to ensure accurate passenger detection and real-time crowd monitoring inside buses. Initially, depth-sensing cameras near entry and exit points is continuously captured. Each frame undergoes preprocessing steps such as noise reduction, resizing, and background filtering to enhance clarity and eliminate irrelevant details. Once the frames are refined, advanced AI-based object detection models, such as YOLO or MobileNet-SSD, are applied to identify and track passengers entering or exiting the bus. A unique tracking ID is assigned to each individual to avoid repeated counting and to maintain precise tracking across frames.

The system then updates passenger counts dynamically by incrementing for entry events and decrementing for exit events. Subsequently, the live passenger count is compared with the predefined vehicle capacity. If the number exceeds the set threshold, an alert notification is instantly sent to the bus driver, bus control unit, or transportation authority

through IoT-based connectivity. This approach ensures accurate passenger management, enhances safety, and provides real-time overcrowding alerts to support smart public transport operations.



Fig(a): Real-time Passenger Occupancy Monitoring

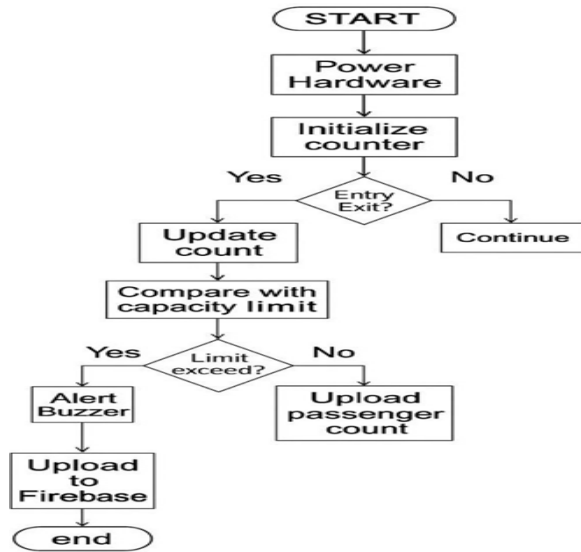
V. OBJECTIVE

Public bus networks often face issues with overcrowding and inefficient passenger movement due to inadequate real-time monitoring solutions. Conventional counting methods like manual observations, estimates based on ticket sales, and simple sensor-based counters tend to be unreliable, susceptible to human error, and ineffective during peak times or in crowded situations. Consequently, transportation authority's encounter difficulties in maintaining passenger safety, ensuring comfort, managing fleet operations, and avoiding situations of overcapacity.

While several AI-based solutions have emerged in recent years, there is currently no standard or widely accepted method for real-time passenger counting and monitoring of bus occupancy. Various models and systems differ in terms of accuracy, processing speed, hardware requirements, and their ability to adapt to real-world conditions such as occlusion, lighting changes, and movements within the vehicle. Thus, there is a necessity to conduct a systematic analysis and comparison of existing AI-driven methods to pinpoint the most effective techniques, comprehend their limitations, and propose future improvements.

This survey seeks to fill this research gap by reviewing contemporary advanced passenger counting systems, assessing their effectiveness, and emphasizing the technological obstacles and future pathways in developing dependable, scalable, and real-time AI-based monitoring and alert systems for bus passengers.

VI. FLOW CHART



Fig(b): Flowchart

VII. SYSTEM REQUIREMENTS

A. Functional Requirements

Real-time Passenger Detection

The system must continuously monitor passengers entering or leaving the bus using AI-powered vision modules. It should identify individuals accurately through live camera streaming and execute counting logic instantly. Real-time operations are critical to avoid lag during peak boarding hours. The system should automatically reset the count when the bus completes a trip cycle.

Processing must occur without manual supervision to maintain automation reliability. The detection algorithm should work under varying light and movement conditions. Any counting error should trigger correction logic using frame-to-frame tracking. This requirement ensures passenger records remain precise throughout the journey. Ultimately, the purpose is to maintain an accurate real-time passenger count at all times.

Overcrowding Alert Generation

The system must generate automatic alerts when the passenger count approaches or exceeds bus capacity. Alerts should be pushed to the driver dashboard and optionally to a central transport unit.

Notifications may include sound, text, or visual warning signals. The alert mechanism must activate

instantly to support real-time decision-making. Capacity values must be configurable based on bus type and guidelines. The system must store alert events for future analysis and reporting. Alerts should differentiate between “near-capacity” and “maximum limit reached.” This helps in preventing potential safety hazards and overcrowding issues. In summary, timely alerting enhances passenger safety and operational control.

Data Logging & Reporting

The system must record passenger count data, timestamps, and alert events. Data logs should be structured for ease of retrieval and transportation analytics. Storage may be cloud-based or local depending on deployment settings. Historical passenger patterns must be accessible for planning transport capacity. Daily, weekly, and route-specific reports should be generated automatically. Logs must support export formats such as CSV or PDF. The system should provide graphical visualization of load patterns. This enables administrators to analyze peak hours, service demand, and occupancy rates.

Integration With Existing Bus Infrastructure

The solution must be compatible with existing bus camera setup. It should allow seamless deployment without extensive hardware replacement. Integration layers must support real-time video feeds and control signals. The driver dashboard app must integrate with bus notification consoles. The system should also interact with GPS units if installed. Hardware configuration must support plug-and-play functionality. APIs should be provided for backend communication and transport control systems. Flexible architecture ensures smooth adoption without operational disruption. Such compatibility minimizes cost and enhances practicality.

Multi-Condition Performance

The system must perform reliably across different environments and passenger conditions. Low-light, crowd density, and rapid movement must not affect accuracy. The model should detect individuals with bags, masks, or obstructed visibility. The logic must be robust against false positives from shadows or objects. Counting must remain stable in motion, including sudden stops and turns. The AI model must self-adapt to variable frame rates during bus vibration. System should re-synchronize if video feed

interruptions occur. Scalability should support increased passengers in highly crowded cities. These conditions ensure field readiness and performance consistency.

B. Non Functional Requirements

Performance & Efficiency

The system must deliver high-speed processing with minimal latency for real-time tracking. Time complexity must support continuous video stream analysis without frame drops. Low computational delay ensures timely decision-making for alerts. The model must be optimized to run on lightweight hardware (Raspberry Pi/Edge AI). Memory and CPU utilization should maintain stability under heavy load. Backup algorithms may be triggered during overload scenarios. Power consumption must be efficient to match onboard battery support. These performance metrics guarantee smooth and dependable operations. Fast response enhances system reliability and public safety.

Accuracy & Reliability

System must ensure a minimum of 90–95% accuracy in passenger counting. False detection rates should be minimized through machine-learning refinement. Algorithm should be validated against diverse bus types and passenger behaviour. Reliability must hold during continuous long-duration runs. Error logs should be maintained for learning and improvement. The system must recover gracefully from temporary faults or network interruptions. Stability testing should confirm no operational failures during peak hours. High reliability builds trust for transport authorities. Accuracy ensures project usefulness and real-world acceptance.

Algorithm

Step 1:

Start system and initialize variables (entry count, exit count, total passengers).

Step 2:

Activate IR sensors and camera at bus doors.

Step 3:

When IR sensor beam breaks:

If direction = entry → increment count. If direction = exit → decrement count. Step 4:

Capture video frames and run YOLO + CNN model to validate count.

Step 5:

Compare IR count and AI count and use corrected value.

Step 6:

Upload updated passenger count to Firebase Cloud.

Step 7:

If count $\geq$ capacityLimit:

Activate buzzer alert.

Log alert event in Firebase.

Step 8:

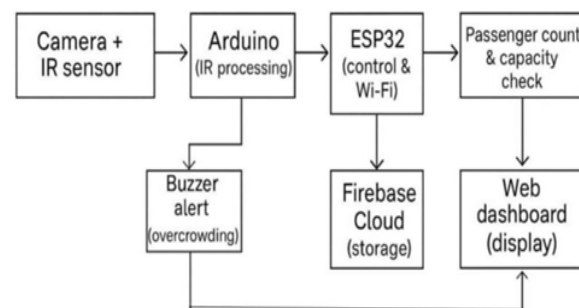
Update web dashboard with live count and alert status.

Step 9:

Continuously repeat steps until bus trip ends.

VIII. BLOCK-DIAGRAM

Passenger Counting System Block Diagram



Fig(c): Passenger Counting System Block Diagram

a. Camera + IR Sensor

The system begins with a camera and infrared (IR) sensor placed at the bus entrance.

The IR sensor detects movement when a person crosses the entry point.

The camera assists in capturing visual data to improve counting accuracy and avoid false detection.

b. Arduino (IR Processing)

The sensor data is sent to the Arduino microcontroller, which processes input from the IR sensor.

Arduino identifies entry and exit events.

It determines whether a person is entering or leaving the bus and updates the count locally.

c. Buzzer Alert (Overcrowding)

If the passenger count exceeds the predefined bus capacity,

The Arduino triggers a buzzer as an overcrowding alert.

This warns the driver/authority to control the

passenger load.

d. ESP32 (Control & Wi-Fi)

The processed count is then transmitted to the ESP32, which acts as the communication and control unit. It connects to the internet using built-in Wi-Fi. Sends live data to cloud storage and dashboard applications.

e. Firebase Cloud (Storage)

The ESP32 sends passenger count data to Google Firebase Cloud.

Firebase stores real-time passenger information. Provides backend support for monitoring from anywhere.

f. Web Dashboard (Display)

A web-based dashboard displays real-time analytics such as:

Current passenger count
Total capacity and remaining seats
Alerts for overcrowding
Historical data logs from cloud

IX. CONCLUSION

In conclusion, an inventive way to improve the effectiveness and safety of public transportation is the AI-Based Smart Bus Passenger Counting and Alert System. The technology tracks bus occupancy levels and counts people precisely in real time using sensors and artificial intelligence. This prevents crowding, encourages drivers to make better decisions, and guarantees a more enjoyable ride for passengers. When the bus reaches or surpasses its capacity, the warning mechanism instantly notifies the driver, assisting in upholding safety standards and legal compliance. The device also helps transportation authority's analyze passenger traffic and optimize schedules or routes.

All things considered, this technology enhances passenger management, minimizes human mistake, and decreases manual labor. By incorporating intelligent automation into the transportation system, it also aids in the creation of smart cities.

The technology has the ability to revolutionize public transportation operations and provide a safer, smarter, and more effective commute with further advancements like cloud connectivity and enhanced analytics.

REFERENCES

- [1] Kumar, R., & Singh, P. (2023). IoT and AI-based Smart Transport Management System. *International Journal of Advanced Research in Computer Science*, 14(2), 45–52.
- [2] Patel, M., & Sharma, K. (2022). AI-Enabled Passenger Safety and Monitoring in Public Transport. *Journal of Intelligent Transportation Systems*, 26(4), 356–367.
- [3] Verma, A., & Gupta, N. (2021). Real-Time Fuel Monitoring and Efficiency Analysis in IoT-enabled Vehicles. *International Journal of Emerging Technology and Advanced Engineering*, 11(6), 112–119.
- [4] Ali, H., & Nair, S. (2021). Smart Ticketing and Passenger Counting Using IoT Sensors. *International Conference on Smart Cities and Automation*, 2021.
- [5] Sahu, R., & Pandey, V. (2019). Intelligent Transport Management Using AI and IoT. *International Journal of Engineering Trends and Technology*, 67(5), 112–117.
- [6] Ahmed, F., & Rao, A. (2019). IoT-based Safety and Security System for Public Transport Vehicles. *Journal of Automation and Smart Systems*, 3(2), 55–64.
- [7] Mishra, S., & Tiwari, P. (2018). Smart Bus Information and Tracking System. *International Journal of Computer Applications*, 180(19), 23–27.
- [8] Yadav, D., & Patel, R. (2018). Automated Fare Collection and Monitoring System Using IoT. *Journal of Electrical and Computer Engineering Innovations*, 12(3), 42–49.
- [9] Khatri, P., & Mehta, J. (2020). Cloud- Enabled Real-Time Public Transport Tracking System. *International Journal of Cloud Computing and Services Science*, 9(1), 72–80.
- [10] Tangudu Yoshitha, Nimmakayala Aadi Kishore, Andhavarapu Thapaswi, Pusarla Venkatesh, Ram Kishor Ponderti Real-time Passenger Counting and Occupancy Optimization using IOT and Deep Learning.