Journal of Information Technology and Computer Science (INTECOMS)
Volume 9 Nomor 1, Tahun 2026
e-ISSN : 2614-1574
p-ISSN : 2621-3249

SISTEM PENDUKUNG KEPUTUSAN PENGIRIMAN BUS BERBASIS AIOT MENGGUNAKAN INTEGRASI YOLOV8 DAN GOOGLE DISTANCE MATRIX API

AIOT-BASED BUS DISPATCH DECISION-SUPPORT SYSTEM USING YOLOV8 AND GOOGLE DISTANCE MATRIX API INTEGRATION

Arinal Dzikrul Haqqy Amir[1], Raihan Evanza[2], Vitri Tundjungsari[3], Eric Julianto[4]
Department of Computer Science, Esa Unggul University[1,2,3,4]
vitri.tundjungsari@esaunggul.ac.id[3]

ABSTRACT

Accurate passenger counting and crowd density estimation play a critical role in smart public transportation systems to enhance safety, service quality, and operational efficiency. Vision-based approaches leveraging deep learning models, particularly You Only Look Once (YOLO), have been widely adopted for real-time passenger detection and tracking due to their high detection speed and accuracy. However, challenges such as occlusion, scale variation, and limited camera viewpoints remain significant issues, especially in densely crowded bus stops and public transport environments. To address these limitations, recent studies have integrated Internet of Things (IoT) architectures with video analytics to enable continuous monitoring of passenger flow. In this study, YOLO-based crowd detection is combined with the Google Distance Matrix API to estimate travel time and distance between bus stops, enabling automatic dispatch recommendations based on real-time overcrowding conditions. The proposed AIoT-based framework supports data-driven decision-making for smart bus operations, improving responsiveness, reducing passenger congestion, and optimizing public transportation scheduling.

Keywords: *AIoT-Enabled Public Transport Surveillance, YOLOv8, Intelligent Transportation Systems, Real-Time Crowd Density Estimation, Google Distance Matrix API.*

ABSTRAK

Kalkulasi penumpang dan estimasi kepadatan kerumunan yang akurat memainkan peran penting dalam sistem transportasi publik cerdas untuk meningkatkan keselamatan, kualitas layanan, dan efisiensi operasional. Pendekatan berbasis visi yang memanfaatkan model deep learning, khususnya You Only Look Once (YOLO), telah diadopsi secara luas untuk deteksi dan pelacakan penumpang secara real-time karena kecepatan deteksi dan akurasinya yang tinggi. Namun, tantangan seperti oklusi, variasi skala, dan sudut pandang kamera yang terbatas tetap menjadi kendala signifikan, terutama di halte bus yang padat dan lingkungan transportasi umum. Untuk mengatasi keterbatasan ini, studi terbaru telah mengintegrasikan arsitektur Internet of Things (IoT) dengan analitik video untuk memungkinkan pemantauan arus penumpang secara berkelanjutan. Dalam penelitian ini, deteksi kerumunan berbasis YOLO dikombinasikan dengan Google Distance Matrix API untuk mengestimasi waktu tempuh dan jarak antar halte bus, sehingga memungkinkan rekomendasi pengiriman armada otomatis berdasarkan kondisi kepadatan secara real-time. Kerangka kerja berbasis AIoT yang diusulkan mendukung pengambilan keputusan berbasis data untuk operasional bus cerdas, meningkatkan responsivitas, mengurangi penumpukan penumpang, serta mengoptimalkan penjadwalan transportasi publik.

Kata Kunci: AIoT-Enabled Public Transport Surveillance, YOLOv8, Sistem Transportasi Cerdas, Estimasi Kepadatan Kerumunan Real-Time, Google Distance Matrix API.

INTRODUCTION

The rapid development of Artificial Intelligence of Things (AIoT) has significantly transformed various sectors, including public transportation systems. AIoT integrates artificial intelligence with Internet of Things (IoT) infrastructures to enable real-time data acquisition, analysis, and decision-making based on sensor information. In urban public transportation, AIoT plays a crucial role in improving service efficiency, reducing passenger waiting times, and supporting adaptive operational strategies at smart bus stops through continuous monitoring and intelligent control mechanisms [1].

A primary challenge in public transportation is the inability to promptly detect overcrowding, leading to a mismatch between passenger demand and fleet availability. Recent studies highlight that vision-based deep learning models, particularly You Only Look Once (YOLO), offer robust solutions for dense transportation environments due to their flexibility in handling occlusion and scale variation [2], [3]. Specifically, the RLGS-YOLO architecture, an enhanced YOLOv8-based model, demonstrated a 2.2% increase in mean Average Precision (mAP) over the baseline YOLOv8n in crowded metro scenarios [2]. Beyond high-end performance, the feasibility of edge-based deployment has been confirmed by YOLOv5 implementations on low-cost Raspberry Pi platforms, which achieved over 70% accuracy in real-world bus operations [5]. Consequently, occlusion-aware passenger detection has become a pivotal research focus, outperforming traditional commercial systems in both cost-efficiency and detection accuracy [4], [5].

Vision-based approaches, particularly the YOLO (You Only Look Once) family, have evolved into sophisticated architectures such as YOLOv8, which deliver high detection accuracy and real-time performance in complex, crowded environments [6]. Current research emphasizes the efficacy of head-based detection and multi-stage frameworks—integrating YOLO-based detection with machine learning regression—to accurately estimate both visible and occluded passengers, achieving accuracy rates exceeding 96% [7]. In this study, YOLO serves as the primary visual sensing component within an AIoT-based smart bus stop system to extract real-time passenger counts from CCTV footage [1]. These detection outputs are processed through a rule-based thresholding mechanism to categorize crowd density into low, medium, and high levels, transforming raw video data into structured, actionable congestion insights.

However, passenger density detection alone is insufficient to achieve a fully adaptive public transportation system. Additional integration is required in the form of an automatic bus dispatch recommendation mechanism capable of estimating the fastest arrival time from nearby depots or available buses to congested stops. To address this need, this research integrates the Google Distance Matrix API to estimate real-time travel distance and duration between bus depots and overcrowded bus stops, enabling data-driven dispatch decisions based on both passenger density and traffic conditions [8], [9], [10].

Therefore, this research focuses on the implementation of a YOLO-based AIoT framework for passenger density detection and automatic notification at smart bus stops, combined with Google Distance Matrix API–based dispatch recommendations. The proposed system not only detects overcrowding conditions in real time but also supports intelligent operational decision-making by recommending the fastest available bus deployment to mitigate congestion at high-demand locations [11].

METHODOLOGY

This study proposes an AIoT-based intelligent passenger density monitoring and bus dispatch recommendation system that integrates YOLO-based computer vision and Google Distance Matrix API-based travel time estimation to support real-time decision-making at smart bus stops.

The proposed methodology consists of five main stages:

1. Data acquisition
2. Visual passenger detection
3. Feature construction
4. Passenger density classification
5. Bus dispatch recommendation based on estimated travel time.

The overall workflow is designed for edge-based execution to ensure real-time responsiveness while maintaining scalability through IoT integration [1].

1. Data Acquisition

Passenger data are collected using CCTV cameras installed at bus stops, forming the primary sensing layer of the AIoT framework. Video streams are captured continuously and processed frame-by-frame for real-time analysis. Vision-based passenger monitoring has been shown to outperform traditional flow-based or sensor-based approaches, especially under dense and occluded conditions [5].

Recent studies demonstrate the effectiveness of YOLO-based models in public transportation environments. RLGS-YOLO, an enhanced YOLOv8-based architecture, achieved a 2.2% improvement in mean Average Precision (mAP) over the baseline YOLOv8n in metro station passenger detection, confirming YOLO's suitability for crowded scenarios [2]. Similarly, Pronello and Garzón Ruiz showed that a low-cost Raspberry Pi–based YOLOv5 system can achieve over 70% accuracy in real-world bus operations, highlighting the feasibility of edge-based deployments [5].

To support dispatch recommendation, traffic-related metadata (distance and travel duration) are obtained via the Google Distance Matrix API, which provides real-time estimates based on current road conditions [12],[13]. These data are used exclusively for Estimated Time of Arrival (ETA) computation, not for passenger detection. Edge-based passenger counting systems enable real-time processing with low latency, making them suitable for smart bus stop deployment [7]. Data used in this research consist of: Passenger data are collected using CCTV cameras installed at bus stops, forming the primary sensing layer of the AIoT framework. Video streams are captured continuously and processed frame-by-frame for real-time analysis. Vision-based passenger monitoring has been shown to outperform traditional flow-based or sensor-based approaches, especially under dense and occluded conditions [5].

Recent studies demonstrate the effectiveness of YOLO-based models in public transportation environments. RLGS-YOLO, an enhanced YOLOv8-based architecture, achieved a 2.2% improvement in mean Average Precision (mAP) over the baseline YOLOv8n in metro station passenger detection, confirming YOLO's suitability for crowded scenarios [2]. Similarly, Pronello and Garzón Ruiz showed that a low-cost Raspberry Pi–based YOLOv5 system can achieve over 70% accuracy in real-world bus operations, highlighting the feasibility of edge-based deployments [5].

To support dispatch recommendation, traffic-related metadata (distance and travel duration) are obtained via the Google Distance Matrix API, which provides real-time estimates based on current road conditions [12],[13]. These data are used exclusively for Estimated Time of Arrival (ETA) computation, not for passenger detection. Edge-based passenger counting systems enable real-time processing with low latency, making them suitable for smart bus stop deployment [7]. Data used in this research consist of:

Table 1. Data Research

Data Type	***Source***	***Purpose***
CCTV video frames from bus stops	AIoT camera sensors installled at bus corridors	Passenger detection and density estimation
Traffic metadata (speed, congestion, distance, time)	Public traffic API/simulated traffic data	Bus travel time prediction
Historical passenger density levels	Calibration of *rule-based* density thresholds	Calibration of *rule-based* density thresholds

The video is captured in real-time and processed per frame for object detection. Traffic data is recorded at periodic time intervals to support travel time prediction modeling [8]. Edge-based passenger counting systems have been shown to support low-latency real-time processing, making them suitable for deployment in smart bus stop environments with limited computational resources [7].

2. Visual Detection Using YOLO

YOLO (You Only Look Once) is employed as the visual detection backbone due to its high inference speed and strong performance in crowded environments. YOLO-based detection has been widely adopted for passenger counting in buses, metro stations, and railway vehicles [2], [4]. The selection of YOLOv8 in this study is motivated by its architectural improvements over previous versions, including enhanced feature fusion and improved small-object detection, as summarized in recent YOLO architecture reviews [1]. The detection process consists of:

a. Each CCTV frame is passed to a YOLO model (YOLOv8 or lightweight edge variant).
b. The model detects bounding boxes corresponding to individual passengers (head-based or full-body depending on configuration).
c. The total number of detected passengers per frame is extracted.

YOLO is used only for detection, not for density classification, aligning with prior studies that separate detection from higher-level inference to improve accuracy in dense environments [5].

3. Feature Construction for Density Classification

Feature-based representations have been shown to improve robustness by supporting rule-based density thresholding under occlusion and partial visibility conditions [5], [7].

Table 2. Feature Construction

Feature	*Description*
N_people	Number of detected persons in the frame
box_density	Ratio of total bounding box area over total frame area
avg_box_scale	Avergae size of detected bounding
box_overlap_rate	Intersection intensity between bounding boxes, indicating
crowd_growth_rate	Rate of increase in passenger count per second

These features are used as indicators to support rule-based density thresholding:

- Low
- Medium
- High

4. Passenger Density Classification

YOLO (You Only Look Once) is employed as the visual detection backbone due to its high inference speed and strong performance in crowded environments.

5. Bus Dispatch Time Prediction

Passenger density detection alone is insufficient for adaptive public transportation management. Therefore, an automatic dispatch recommendation module is introduced. This module uses the Google Distance Matrix API to estimate the fastest arrival time from nearby bus depots or available buses to congested bus stops [9]. Travel time estimation and routing decisions can be formulated as graph-based problems, where weighted paths determine optimal route selection, aligning with reduced Google Matrix computation methods [14].

Table 3. Bus Dispatch

Feature	***Description***
route_distance	Distance from bus
avg_traffic_speed	Mean speed
current_congestion_level	Congestion index from
time_of_day	Peak / non-peak time
weather_factor	Optional external

The bus with the minimum Estimated Time of Arrival (ETA) is selected for dispatch. Similar ETA-driven decision mechanisms have been shown to improve operational responsiveness in intelligent transport systems [1], [11], [15].

6. Feature Construction for Density Classification

Trigger Conditions:

Table 4. System Decision

Condition	***Action***
Density = High	Send alert to control center
ETA Prediction completed	Recommended fastest bus to
Multiple bus pools available	Choose minimum ETA result

Notification outputs include:

- Passenger density alert
- Estimated crowd severity
- Recommended bus ID to deploy
- Estimated arrival time

Feature Construction for Density Classification

Table 5. Method Summary

Stage	*Technique*
Visual Detection	YOLO (v8 / Tiny / Edge variant)
Density Feature Extraction	Spatial & bounding box analytics
Density Classification	Rule-based thresholding derived from YOLO detection output
ETA Prediction	Google Matrix API
Deployment	Edge AI + IoT integrated pipeline

RESULT AND DISCUSSION

1. System Stability and Performance Definition

A system is considered stable if and only if it provides reliable and accurate real-time decision support for operational efficiency, validated through consistent performance in passenger density classification and bus dispatch recommendation accuracy. However, challenges such as occlusion, scale variation, and limited camera viewpoints remain significant issues in densely crowded bus stops. To address these challenges Recent YOLO-based approaches incorporating feature enhancement and attention mechanisms have demonstrated improved robustness in dense and occluded environments [1].

In the proposed AIoT framework, System stability is evaluated based on the consistency of YOLO-based passenger detection, system response latency, and the correctness of bus dispatch recommendations generated using the Google Distance Matrix API, as this component directly influences downstream operational decisions such as alert triggering and bus dispatch recommendations [10]. Previous studies have demonstrated that inaccuracies at the perception level (e.g., passenger detection errors) can propagate and degrade system-level decision quality, especially in dense and highly dynamic environments [5], [7].

The use of YOLO-based object detection as the visual sensing layer ensures real-time performance while maintaining sufficient detection accuracy in crowded public transportation scenarios [2], [5]. Furthermore, decoupling visual detection from high-level density inference improves system robustness by decoupling raw detection from high-level density inference, as suggested in recent passenger counting and congestion estimation studies [6], [7]. Recent YOLO-based approaches incorporating feature enhancement and attention mechanisms have demonstrated improved robustness in dense and occluded environments, reinforcing the suitability of YOLO for complex public transportation scenarios [1].

2. Implications for Bus Dispatch Recommendation

Accurate passenger count estimation obtained from YOLO-based detection enables the timely activation of the bus dispatch recommendation module. When predefined overcrowding thresholds are exceeded, the system estimates the fastest arrival time using the Google Distance Matrix API and recommends dispatching the most suitable bus from the nearest depot to the congested bus stop [9].

From a conceptual perspective, travel time estimation and routing decisions in transportation systems can be formulated as graph-based problems, where weighted paths and node connectivity play a critical role in determining optimal routes [16]. Network-based distance estimation has been extensively studied using Google Matrix–based analysis in directed graphs, providing a theoretical foundation for distance-based dispatch decision modeling [8], [10], [17].

This approach aligns with integrated passenger flow and operational optimization frameworks, in which perception-driven alerts are combined with travel-time estimation to support adaptive fleet management strategies [1]. By incorporating real-time distance and duration data, dispatch decisions become not only reactive to congestion conditions but also optimized with respect to current traffic situations [13]. Such integration has been shown to significantly improve system responsiveness and reduce passenger waiting times in intelligent transportation systems [1].

Furthermore, traffic-aware dispatch strategies that leverage real-time travel conditions have been demonstrated to enhance fleet responsiveness and reduce passenger waiting times in urban public transportation environments [1], [18]. Graph-based routing approaches are therefore widely adopted in intelligent transportation systems to support optimal vehicle allocation and dispatch decisions under dynamic traffic conditions [9], [14], [15]. Collectively, these network-based distance estimation and routing methods provide a scalable framework for integrating spatial relationships and operational constraints in transportation decision-support systems [14].

CONCLUSION

This study successfully implemented an AIoT-based intelligent transportation framework by integrating YOLO-based real-time visual detection with the Google Distance Matrix API. The results demonstrate that while YOLO provides reliable passenger monitoring in crowded environments, the integration of real-time travel metadata enables proactive dispatch recommendations [8], [9]. By shifting from passive monitoring to an active decision-support tool, the proposed system facilitates data-driven bus deployment to effectively mitigate overcrowding at smart bus stops [1].

Overall, the proposed framework demonstrates that combining YOLO-based perception, rule-based density estimation, and real-time travel analytics provides an effective and scalable solution for smart bus stop management in urban transportation systems. Recent advances in YOLO-based detection have demonstrated increased robustness in dense and occluded environments, reinforcing its suitability for smart transportation applications [1]. This integration positions the proposed system as an effective decision-support tool for public transportation management, enabling data-driven operational planning and congestion mitigation [1].

REFERENCE

Paganelli, M. G., Gallo, M., Massenio, P. R., & Naso, D. (2025). Integrated passenger flow analysis and street-level monitoring for public transport management using deep learning and IoT. IEEE Access, 13, 143401–143413.

Qin, Y. D., Li, X. W., He, D., Zhou, Y., & Li, L. (2024). RLGS-YOLO: An improved algorithm for metro station passenger detection based on YOLOv8. Engineering Research Express, 6(4), 045095.

Pronello, C., & Garzón Ruiz, X. R. (2023). Evaluating the performance of video-based automated passenger counting systems in real-world conditions: A comparative study. Sensors, 23(18), 7719.

Rakhymova, A., Mussina, A., Aubakirov, S., & Trigo, P. (2024). Creation of a high-precision, single-board computer-based intelligent passenger counting system. Procedia Computer Science, 241, 212–219.

Kim, H., Sohn, M. K., & Lee, S. H. (2022). Development of a real-time automatic passenger counting system using head detection based on deep learning. Journal of Information Processing Systems, 18(3), 428–442.

Kim, K. H., An, T. K., & Kim, S. (2025). Estimating invisible passenger count using CCTV footage: An approach combining object detection models and machine learning. IEEE Access, 13, 152131–152147.

Votis, K. (2024). Improving passenger detection with overhead fisheye imaging. IEEE Access, 12, 66237–66247.

Hossain, S., Islam, M., & Rahman, A. (2022). Analyzing terrific traffic in urban areas: A small step towards bringing order. Computing and Informatics, 41(2), 571–589.

Abdulhameed, O. (2025). API-based dynamic programming model and optimization of vehicle routing: Cases of fluctuations in demand, traffic, capacity, and availability. Journal of Project Management, 10(4), 613–628.

Wang, F., & Xu, Y. (2011). Estimating O–D travel time matrix by Google Maps API: Implementation, advantages, and implications. Annals of GIS, 17(4), 199–209.

Chavan, P., et al. (2025). Leveraging real-time data: A location-based ambulance booking and tracking system with geofencing. Journal of Integrated Science and Technology, 13(2), 1–9.

[Author Name Missing] (2022). Near real-time spatial and temporal distribution of traffic emissions in Bangkok using Google Maps Application Program Interface.

Shibghatullah, A. S., Jalil, A., Wahab, M. H. A., Ng, J., & Soon, P. (2022). Vehicle tracking application based on real time traffic. International Journal of Electrical and Electronic Engineering & Telecommunications, 11(1), 67–73.

Terven, J., & Cordova-Esparza, D. M. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5(4), 1680–1716.

Mishra, A., & Ray, A. K. (2020). IoT cloud-based cyber-physical system for efficient solid waste management in smart cities: A novel cost function based route optimisation technique for waste collection vehicles using dustbin sensors and real-time road traffic informatics. IET Cyber-Physical Systems: Theory & Applications, 5(4), 330–341.

Kubitza, D. O., & Weßling, K. (2025). Decisions during the transition to higher education. Journal for Educational Research Online, 17(1).

Kotelnikova, E., Frahm, K. M., & Shepelyansky, D. L. (2022). Fibrosis protein-protein interactions from Google matrix analysis of MetaCore network. PLOS ONE, 17(5), e0267425.

Mancuso, N., & Sullivan, P. S. (2025). Methods for estimating public transit travel times to healthcare services as a measure of equitable healthcare access. Annals of Epidemiology, 111, 24–29.

Yang, B., Cao, J., Liu, X., Wang, N., & Lv, J. (2023). Edge computing based real-time passenger counting using a compact convolutional neural network. IEEE Transactions on Intelligent Transportation Systems.

Delzanno, G., et al. (2023). Automatic passenger counting on the edge via unsupervised clustering. Sensors, 23(15), 6821.