Research Article

Lightweight Deep Learning for Autonomous Human Counting System on Low-Cost Hardware

Anh Vu LE[1], *Nhat Tan LE*[2], *Anh Dung NGUYEN*[2], *Ngoc Nghia NGUYEN*[2], *Hai Dang LE*[2], *Minh Dang TRAN*[2], *Bui Vu MINH*[3*], *Lam Dong HUYNH*[4], *Mohan Rajesh ELARA*[5]

[1]Advanced Intelligent Technology Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[2]Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[3]Faculty of Engineering and Technology, Nguyen Tat Thanh University, 300A - Nguyen Tat Thanh, Ward 13, District 4, Ho Chi Minh City, Vietnam
[4]Z755 Electronic Communication CO., LTD, 2A Phan Van Tri Street, Ward 10, Go Vap District, Ho Chi Minh City, Vietnam
[5]The ROAR Lab, Engineering Product Development, Singapore University of Technology and Design, Singapore 487372, Singapore

leanhvu@tdtu.edu.vn, 42001078@student.tdtu.edu.vn, 41900939@student.tdtu.edu.vn, ngocnghia0904@gmail.com, llehaidang@gmail.com, dangtranminh151@gmail.com, bvminh@ntt.edu.vn, huynhlamdong@gmail.com, rajeshelara@sutd.edu.sg

*Corresponding author: Bui Vu Minh; bvminh@ntt.edu.vn

DOI: 10.15598/aeee.v19ix.250301

Article history: Received Mar 01, 2025; Revised Apr 05, 2025; Accepted Apr 28, 2025; First Published May 05, 2025.

This is an open access article under the BY-CC license.

Abstract.

Accurate and efficient human counting is essential for optimizing public transportation and advancing smart city infrastructure. This paper evaluates proposed lightweight deep learning models for autonomous human counting system on low-cost hardware, ensuring real-time monitoring and enhanced operational efficiency. While existing methods, such as DeepSORT, Kalman Filters, and YOLO variants, are often implemented on high-end hardware, they typically prioritize accuracy over computational efficiency. Few object detection and tracking techniques can run in real-time on low-end hardware. This work advances the field by utilizing optimized deep learning models suitable for embedded systems with constrained resources. Specifically, fine-tuned YOLOv8 is employed for head detection, combined with ByteTrack for robust tracking, outperforming YOLOv5 and YOLOv11 in accuracy and efficiency. Archiving the 15 FPS and more then 90% accuracy on the real environment deployment on both RISC-V architecture with an integrated NPU (MaixCAM) and ARM v8 (Raspberry Pi), The proposed system demonstrates its suitability for real-time, cost-effective, and scalable autonomous human counting in public transit environments.

Keywords

Deep Learning, Low-cost hardware, Human counting, Human detection, Human tracking.

1. INTRODUCTION

Efficient and secure transportation systems are essential for modern urban mobility, particularly in student transport and public transit. In developing countries like Vietnam, student safety on school buses is a growing concern, especially after several incidents since 2020 where children were left unattended. In the 2023–2024 academic year alone, Vietnam had around 18.5 million students in primary to high school, underscoring the scale of the issue. Addressing these safety concerns through advanced technological solutions has become a necessity. Although regulations—including national decrees, road traffic laws, circulars, resolution numbers, and directives—provide a legal framework for transportation, they lack the integration of modern AI-driven monitoring technologies [1, 2, 5].

Beyond student transport, public transit efficiency is directly influenced by real-time passenger counting. Accurate passenger monitoring is essential for improving operational planning, optimizing resource allocation, and enhancing user experience. Traditional methods, such as manual counting or basic infrared sensors, suffer from accuracy limitations, environmental dependencies, and scalability challenges [3, 4]. Computer vision and machine learning techniques offer robust alternatives by automating the passenger counting process. However, deploying these AI-driven systems on low-cost hardware introduces challenges related to computational efficiency, power consumption, and model accuracy [6–8].

For instance, MobileNetV3 has been widely adopted for embedded AI applications due to its ability to deliver high performance with minimal computational resources [9]. Similarly, Tiny-YOLO and NanoDet models have been explored for real-time human detection on resource-constrained devices [10, 11]. These models leverage depthwise separable convolutions and quantization techniques to optimize inference speed while maintaining acceptable accuracy. Additionally, some works have proposed hybrid approaches that combine lightweight CNNs with Transformer-based architectures to further enhance detection robustness [12, 13].

Recent studies have explored the application of deep learning in human detection and counting. Convolutional Neural Networks (CNNs) and You Only Look Once (YOLO)-based models have demonstrated high accuracy in detecting people in crowded environments [14, 15]. However, these models often require substantial computational power, making them less suitable for edge deployment on low-cost hardware. To address this, researchers have investigated lightweight deep learning models, such as MobileNet, EfficientNet, and ShuffleNet, which offer a balance between accuracy and computational efficiency [16, 17].

Artificial Neural Networks (ANNs) and Deep Reinforcement Learning (DRL) have been widely applied in the field of autonomous systems, particularly for path planning and robotic control. Reinforcement learning-based approaches have shown promising results in optimizing coverage path planning (CPP) for robotic cleaning and maintenance applications. Artificial Intelligence (AI) has significantly advanced robotics, particularly in autonomous navigation, path planning, and medical imaging. Deep reinforcement learning (DRL) has been successfully applied for complete coverage path planning in reconfigurable robots [18, 19], while deep learning techniques have enabled autonomous operations such as staircase cleaning [20]. Additionally, convolutional neural networks (CNNs) have enhanced medical image segmentation, improving diagnostic accuracy in applications like breast nodule detection and intracranial hemorrhage segmentation [21, 22]. Lakshmanan et al. (2020) proposed a DRL-based CPP strategy for tetromino-based cleaning and maintenance robots, demonstrating improved efficiency in large-scale environments [23]. Similarly, Kyaw et al. (2020) employed DRL for solving the Traveling Salesman Problem (TSP) in grid-based decomposition maps, enhancing the coverage efficiency of reconfigurable robots [24]. These studies highlight the effectiveness of DRL in dynamically generating optimal coverage paths while adapting to different environmental constraints. Prabakaran et al. (2020) [25] introduced Hornbill, a self-evaluating hydro-blasting robot for ship hull maintenance. The work in [26] improvements could integrate deep learning and fuzzy logic classifier for defect detection. The Tetris-inspired reconfigurable floor-cleaning robot [27, 28], the polyiamond inspired self-reconfigurable floor tiling robot [29] and the Panthera reconfigurable pavement-cleaning robot [30, 31] present promising platforms for future research, particularly in integrating reinforcement learning and computer vision techniques to enable adaptive and intelligent navigation.

In addition to DRL, heuristic and evolutionary algorithms have also been explored for autonomous systems. Cheng et al. (2020) proposed a multi-objective genetic algorithm (GA)-based autonomous path planning approach for reconfigurable tiling robots, balancing multiple objectives such as energy efficiency and coverage completeness [32]. Similarly, Le et al. (2020) utilized an evolutionary algorithm for complete coverage path planning, optimizing tiling-based surface coverage [33]. Graph-based and learning-based models have also been considered for CPP tasks. Cheng et al. (2019) introduced a graph-theoretic approach for accomplishing complete coverage in reconfigurable robotic systems [34], while Yin et al. (2020) applied deep learning techniques for robotic table-cleaning tasks, demonstrating the potential of neural networks in adaptive coverage solutions [35]. These works [36–38] demonstrate the application of reinforcement learning in robot path planning, energy efficiency, and autonomous coverage. By leveraging adaptive learning mechanisms, RL helps improve decision-making in coverage path planning, energy-aware navigation, and environment perception, making robotic systems more efficient and autonomous.

Deploying lightweight AI models for people counting in transportation systems presents several challenges, including occlusions, varying lighting conditions, and real-time inference on low-power hardware. Studies have shown that pruning and knowledge distillation techniques can reduce model size while maintaining efficiency [39,40]. Additionally, advancements in edge AI hardware—such as NVIDIA Jetson Nano, Coral Edge TPU, and Raspberry Pi-based accelerators—have enabled real-time deep learning inference for transportation applications [41,42].

To successfully implement a lightweight AI model for human counting, several key challenges must be addressed: Applying quantization and pruning techniques to reduce model size while preserving accuracy. Evaluating real-world performance across diverse environmental conditions, including varying lighting and occlusions. Conducting a comparative analysis of different lightweight AI architectures for human detection in transportation scenarios.

Basing on the problem formulation, in this work, we propose an optimized lightweight deep learning model for real-time human counting in transportation systems. Our approach aims to achieve over 93% accuracy, a frame rate exceeding 15 FPS, and stable operation for at least 12 hours while keeping the total system cost under 150. Development of an efficient deep learning model optimized for low-cost hardware deployment.

The study contributions are summarized as follows:

- Design and develop a complete system for counting students on schoolbus.
- Provide a proposed lightweight model combing object detection and object tracking suitable for embedded applications, ensuring up to 93% accuracy, developing a multiobject counting algorithm for moving objects with FPS up to 15 FPS, and web-based monitoring.
- Evaluate the proposed model and present results from the real-world deployment using various models and hardware.

The paper is structured as follows: Section 1 delves into introduction, while Section 2 presents the proposed system to evaluation the light deep-learning model based human counting for low cost hardware implemenation. Experimental results are detailed in Section 3, then followed by the conclusion and remarks in the final section.

2. Proposed System

2.1. Hardware and software components

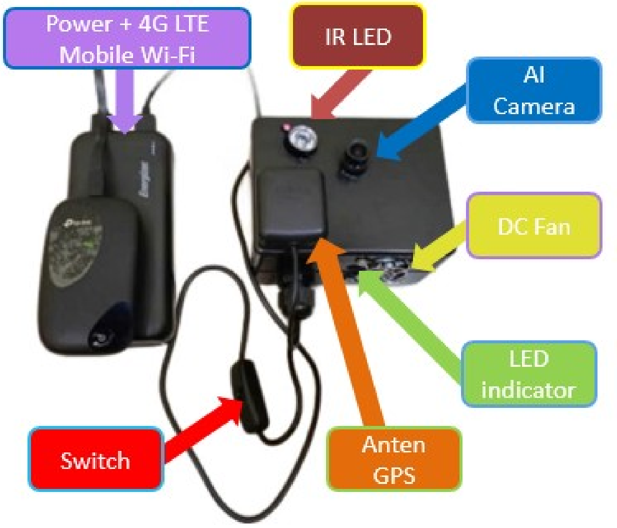
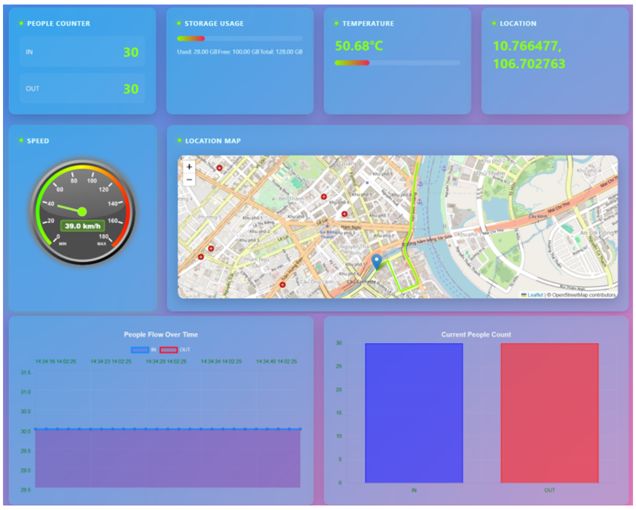


Fig. 1: Overview hardware components of the proposed system


PEOPLE COUNTER
IN 30
OUT 30

STORAGE USAGE
Used: 28.00 GB Free: 100.00 GB Total: 128.00 GB

TEMPERATURE
50.68°C

LOCATION
10.766477, 106.702763

SPEED
39.0 km/h

LOCATION MAP

Fig. 2: Overview software of the proposed system

Fig.1 illustrates the system components, used alongside the website in Fig.2. Housed in a 115×90×55 mm enclosure, the system includes a MaixCAM, GPS module, antennas, LED indicators, IR LED, and a DC fan for active cooling.

The GPS module is used to locate the position of the bus while it is moving, allowing for accurate route tracking. The green LED indicator will be on when the system is active and off when the system is inactive.MaixCAM supports camera modules with resolutions up to 5 megapixels (MP), specifically GC4653 and OS04A10 sensors with 4 MP resolution. The GC4653 sensor has a size of 1/3" and supports a maximum frame rate of 60 FPS at 720P resolution, while the OS04A10 sensor is larger (1/1.79") and optimized for low-light conditions, supporting a maximum frame rate of 90 FPS at 720P resolution. During experiments, the configuration used includes: Resolution: 416 × 416 pixels, Frame rate: 5 FPS (frames per second). MaixCAM uses the RISC-V C906 chip with a clock speed of 1 GHz for the main core and 700 MHz for the secondary core, enabling smooth execution of AI and IoT tasks.

The system transmits the real-time counting results over Wi-Fi using the MQTT protocol to a central server. A TP-Link router with a SIM attached is used to provide a stable internet connection to the system during the bus's travel. At the same time, the resulting image is stored locally on a microSD card for further analysis. This hybrid approach ensures both instant data reporting and offline data backup, enabling flexible deployment in a variety of environments.

The system is installed at the bus entrance (assuming a single-door layout), with hardware costs totaling around $150. As shown in Fig. 3, this placement allows the camera to capture passengers' heads and movement area, enabling accurate tracking and direction inference.

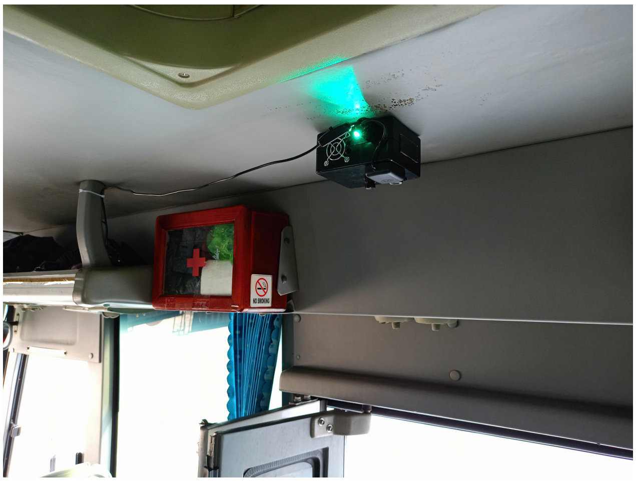
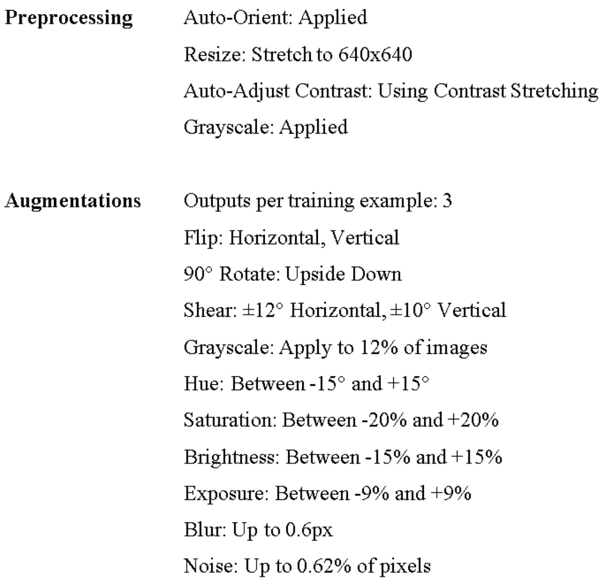


Fig. 3: System is deployed on schoolbus at real environment

2.2. Light Deeplearning Model based Human Counting

The AI model for human counting is fine-tuned from YOLOv8n using a dataset of 5,000 annotated images, specifically curated to enhance detection accuracy in crowded and dynamic environments such as school buses and public transit. This dataset includes diverse scenarios with varying lighting conditions, occlusions, and different passenger postures to ensure robust performance. Preprocessing and augmentation methods, as shown in Fig. 4, were also applied to enhance the uniqueness of the data. To optimize the model for real-time inference on low-cost hardware, we employed integer quantization (INT8), resulting in a model size of approximately 3.28 MB. We did not utilize pruning techniques in this process.



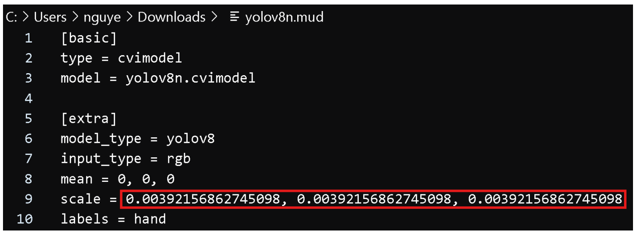
Preprocessing	Auto-Orient: Applied Resize: Stretch to 640x640 Auto-Adjust Contrast: Using Contrast Stretching Grayscale: Applied
Augmentations	Outputs per training example: 3 Flip: Horizontal, Vertical 90° Rotate: Upside Down Shear: ±12° Horizontal, ±10° Vertical Grayscale: Apply to 12% of images Hue: Between -15° and +15° Saturation: Between -20% and +20% Brightness: Between -15% and +15% Exposure: Between -9% and +9% Blur: Up to 0.6px Noise: Up to 0.62% of pixels

Fig. 4: Image preprocessing and augmentation methods

To optimize the model for real-time inference on low-cost hardware, the trained YOLOv8n model is converted into the .cvimodel format, a lightweight structure designed for efficient deployment on MaixCAM. This conversion significantly reduces the computational overhead, enabling smooth operation on resource-constrained devices while maintaining high detection accuracy. The .cvimodel format is particularly suited for embedded AI applications as it minimizes latency and optimizes memory usage. Fig. **??** illustrates the .mud file, which encapsulates critical model information, including architecture details, quantization settings, and hardware compatibility.

Further improvements include the application of post-processing techniques to refine detection accuracy and reduce false positives. Non-Maximum Suppression (NMS) thresholds are adjusted to enhance multi-object detection, ensuring that overlapping passenger detections are accurately counted. Additionally, inference speed is optimized through model pruning and quantization, which lower computational complexity without significantly affecting accuracy.

To validate the model's effectiveness, extensive testing was conducted under real-world conditions, including varied illumination and different passenger densities. Fig. 5 and Fig. 6 presents several examples and size of the Yolov8n.cvimodel where students are successfully detected using the custom-trained YOLOv8n model. These results demonstrate the system's ability to provide reliable and consistent human counting, reinforcing its suitability for autonomous monitoring in transportation environments.

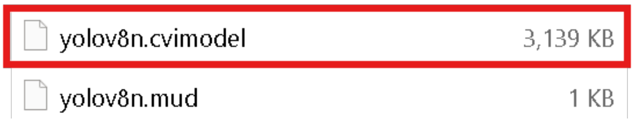


C: > Users > nguye > Downloads > yolov8n.mud

```
[basic]
type = cvimodel
model = yolov8n.cvimodel

[extra]
model_type = yolov8
input_type = rgb
mean = 0, 0, 0
scale = 0.00392156862745098, 0.00392156862745098, 0.00392156862745098
labels = hand
```

Fig. 5: The File .mud structures archived after fine-tun the lightweight model



yolov8n.cvimodel	3,139 KB
yolov8n.mud	1 KB

Fig. 6: File size of YOLOv8.cvimodel deployed on MaixCAM

3. Experimental Results

3.1. Evaluation metrics

The performance of lightweight deep learning models is evaluated using standard object detection metrics.

Intersection over Union (IoU) quantifies the overlap between a predicted bounding box and the ground truth, defined as:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{1}$$

IoU ranges from 0 (no overlap) to 1 (perfect overlap).

The **accuracy** A represents the proportion of correct predictions:

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

The **F1-score** is the harmonic mean of precision and recall:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Precision P measures the proportion of true positive predictions among all positive predictions:

$$P = \frac{TP}{TP + FP} \tag{4}$$

In object detection, a predicted bounding box is considered a true positive (TP) if its IoU with the ground truth exceeds a defined threshold; otherwise, it is a false positive (FP).

Recall R measures the proportion of correctly predicted positive instances:

$$R = \frac{TP}{TP + FN} \tag{5}$$

False negatives (FN) represent undetected objects.

The **Precision–Recall Curve (PRC)** illustrates the trade-off between precision and recall at varying confidence thresholds and is especially useful for evaluating performance on imbalanced datasets.

Average Precision (AP) is the area under the PRC and summarizes the performance for a specific class. The **mean Average Precision (mAP)** is the average of AP across all classes. Common benchmarks include **mAP@50** (IoU = 0.50) and **mAP@50:95** (averaged across IoU thresholds from 0.50 to 0.95 in steps of 0.05).

Frames per Second (FPS) quantifies the model's processing speed. Higher FPS values indicate better real-time performance, which is crucial for embedded and edge AI applications.

Here, TP, FP, TN, and FN follow standard confusion matrix terminology [43, 44].

3.2. Results and analysis

Initial tests using traditional models on a Raspberry Pi 4 Model B (8GB RAM) showed poor performance, as detailed in Table I. The hardware proved insufficient for real-world scenarios, especially with fast-moving subjects.

We then switched to hardware with a RISC-V CPU architecture featuring an integrated NPU, MaixCAM, with a performance of 1 TOPS@INT8. The experimental results, as shown in Table II, demonstrate more promising performance, making it suitable for practical deployment. To further improve the performance of the model, we utilized a dataset consisting of 10,128 images, divided into a training set (9,990 images) and a validation set (138 images). To enhance the generalization ability of the model, data augmentation techniques such as image rotation, brightness adjustment (from -15% to +15%), noise addition, Flip (horizontal, vertical), 90-degree rotation, Shear, Hue change, Saturation, and Exposure were applied. All original images are collected during the day; the use of light augmentations helps the model better adapt to different lighting conditions in practice. The photo is resized to 640x640, turned into grayscale, and the contrast is automatically adjusted. During development and testing, various real-world challenges were observed, including

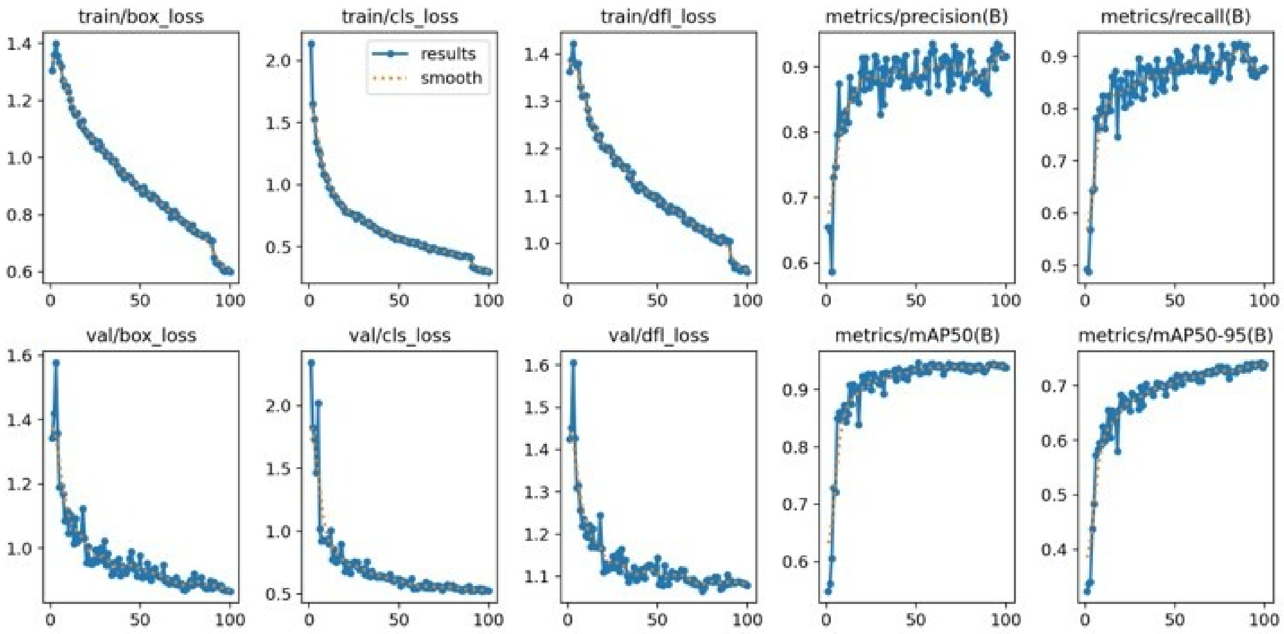


Fig. 7: Results after training 100 epochs

motion blur, partial occlusions, and students moving in groups. However, by leveraging the lightweight deep learning model, particularly the fine-tuned YOLOv8n for head detection combined with ByteTrack for tracking, the system consistently achieved precise counting. The real-time tracking mechanism ensures that each student is only counted once when crossing the respective ROI lines, preventing duplication or discounts.

In training process, we employ image preprocessing and augmentation methods then train that dataset for 100 epochs with YOLOv8n. This approach ensures consistency and enables direct comparison with the results outlined in the paper. The F1 curve and overview of the results after training the model with YOLOv8n in Fig. 7, on the other hand, is a graphical representation that shows how the F1 Score varies with different classification and provide visualizations of the result charts from the training process. The loss graphs (train/box-loss, train/cls-loss, train/dfl-loss) show how the model is learning and improving over training iterations. Initially, the losses decreased rapidly, indicating that the model learned important characteristics effectively. After that, the rate of loss reduction slows down and goes into a stable state, indicating that the model is fine-tuning the parameters. This indicates that the pattern is converging, i.e., gradually achieving the best possible performance.

To evaluate the generalizability of the model, we compare the losses on the training and validation sets. The val/box-loss, val/cls-loss, val/dfl-loss graphs show losses on the test set similar to those on the training set. The small gap between the losses on these two data sets shows that the model has good generalization ability and is not overfitting, that is, it not only learns well on the training data, but also performs well on the new data.

The evaluation metrics (metrics/precision(B), metrics/recall(B), metrics/mAP50(B), metrics/mAP50-95(B)) show how well the model performs in object recognition. The graphs show that these metrics have increased over time and reached a stable level. This proves that the model has learned to recognize objects accurately and completely, with high precision and recall.

Fig. 8and Fig. 9illustrate the Precision (P) curve, Recall (R) curve, Precision-Recall (PR) curve, and the confusion matrix. [44]

Fig.10 shows detections using the pretrained YOLOv8n, while Fig.12 shows results from proposed custom YOLOv11n. As seen in Fig.5 and Fig.11, head-focused model outperforms the baseline, detecting students more reliably.

After installing the hardware components and load to the model to main processing unit, the system is underwent a deployment he system operated for 14 days at 6 hours per day with a 0% failure rate. Startup takes about 30 seconds to power on, connect to Wi-Fi, begin counting, send data via MQTT, and initialize GPS. GPS acquisition, averaged over 10 trials, takes approximately 5 minutes depending on the infrastructure.

By tracking student heads as they pass through the predefined Region of Interest (ROI) 1 and ROI 2 lines, the system effectively distinguishes between individuals entering and exiting the bus. Figure 11 illus-

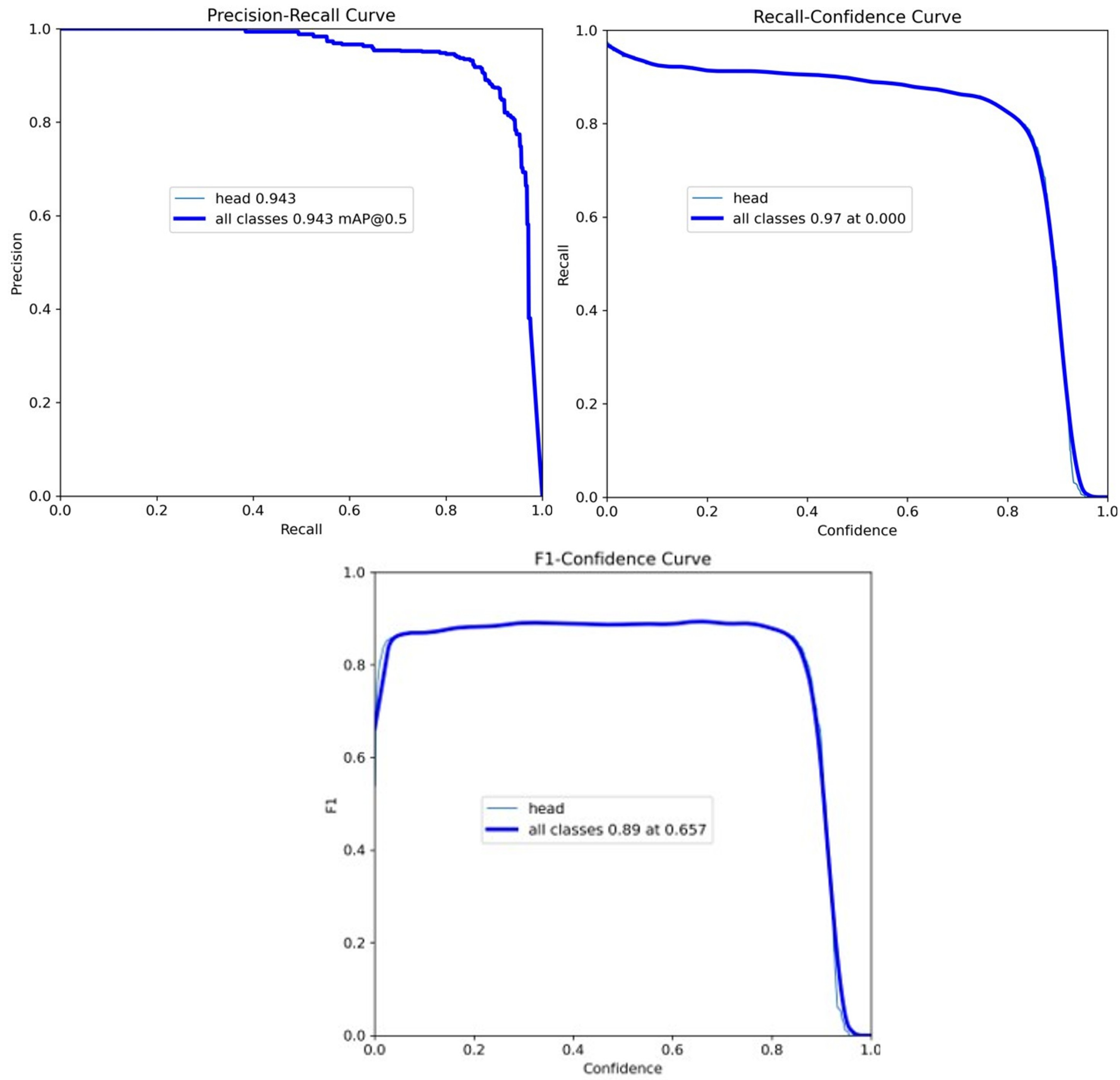


Fig. 8: (P) curve, Recall (R) curve and F1 curve

trates the output of the human counting system during real-time deployment, maintaining a stable frame rate of 15 FPS. This high processing speed ensures that even when students board and disembark rapidly, their movements are accurately tracked without missing detections. During development and testing, various real-world challenges were observed, including motion blur, partial occlusions, and students moving in groups. However, by leveraging the lightweight deep learning model, particularly the fine-tuned YOLOv8n for head detection combined with ByteTrack for tracking, the system consistently achieved precise counting. The real-time tracking mechanism ensures that each student is only counted once when crossing the respective ROI lines, preventing duplication or discounts. Furthermore, the system demonstrates robust performance under different environmental conditions, including varying lighting and passenger density. This accuracy is critical for ensuring student safety, as it provides reliable data on the exact number of passengers on board at any given time. The integration of efficient AI-based tracking with optimized model deployment enables the system to function effectively on low-cost hardware, making it a scalable and practical solution for real-world transportation monitoring.

Without cooling, the system stabilizes at 77°C after 12 hours in a non-air-conditioned environment. With cooling and air conditioning, the temperature stays below 55°C—a reduction of over 22°C. In terms of energy

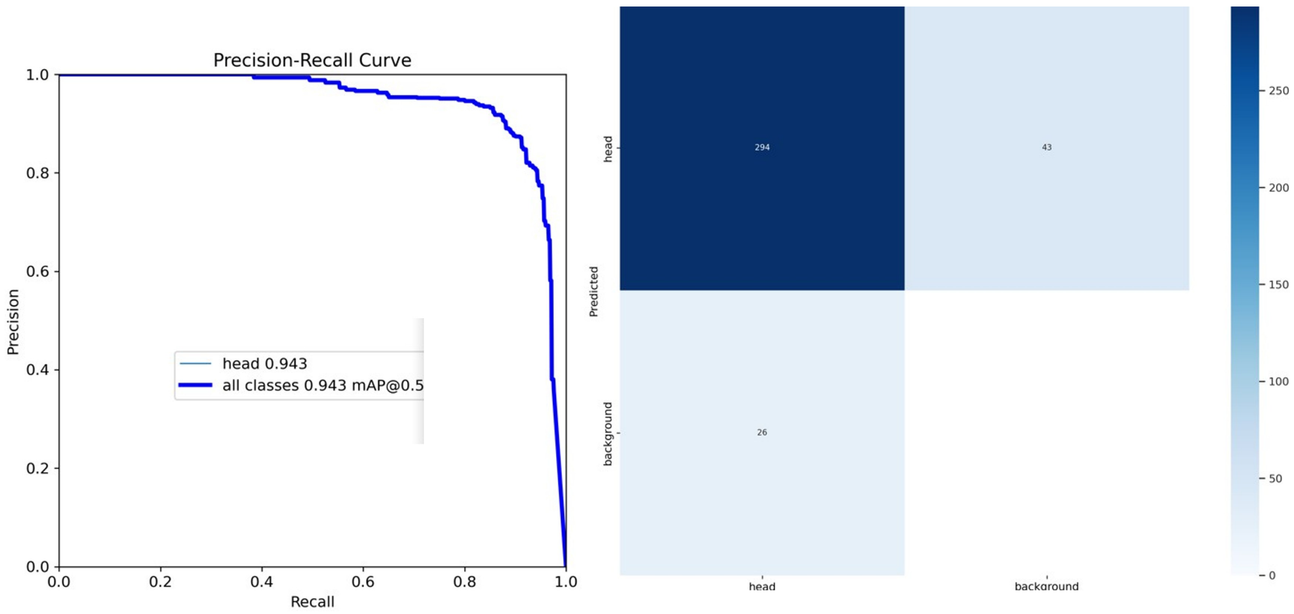


Fig. 9: Precision-Recall (PR) curve and the confusion matrix

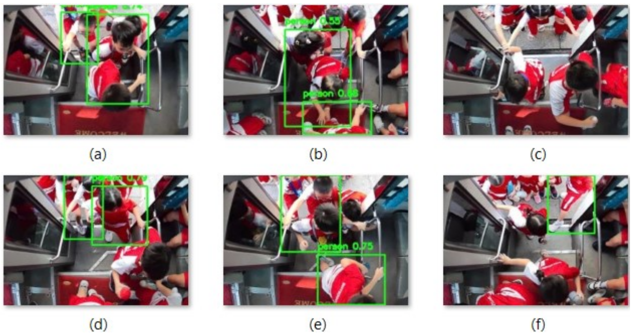


Fig. 10: Students are detected using a pretrained YOLOv8n model in the following cases: **(a)** 1; **(b)** 2; **(c)** 3; **(d)** 4; **(e)** 5; **(f)** 6.

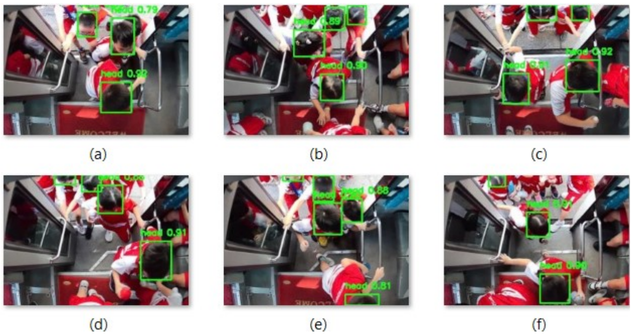


Fig. 11: TStudents are detected using proposed custom YOLOv11n model. **(a)** 1; **(b)** 2; **(c)** 3; **(d)** 4; **(e)** 5; **(f)** 6.

usage, the system runs on a 20,000 mAh portable battery, supporting about 6 hours of daily use for up to 3 days.

Metric	Run 1	Run 2	Run 3
Total Images	2540	4609	10,128
Training Set	2112	4578	9,990
Validation Set	428	31	138
Lighting Augmentation	Yes	Yes	Yes

Tab. 1: Dataset Summary Across Experimental Runs

4. Conclusion and Future Works

This paper presents a cost-effective AI-driven people-counting and location-tracking system designed for public transportation. Despite operating on low-cost hardware priced under 150, the system achieves a frame rate exceeding 15 FPS and an accuracy of over 93%, with potential for further improvement. By optimizing power consumption and leveraging efficient AI models, the system aligns with Green IT principles, contributing to advancements in multimedia processing, communication technology, and IoT-powered smart city initiatives.

The proposed solution integrates real-time AI-based passenger monitoring with seamless IoT connectivity, ensuring scalability and sustainability while enhancing public transit efficiency. If widely adopted, it could significantly contribute to IoT innovation and national development, particularly in improving commuter safety. Moreover, this system holds substantial societal value, especially in safeguarding children and vulnerable passengers.

To further improve the system, future research will focus on: Enhancing Accuracy and Robustness – In-

Input	**Model**	**Other Components**	**People Counting**	**FPS**	**Performance Evaluation**
Webcam 300x300	pre-trained mobilenet_v1 (tflite)	-	Yes	6-8	Stable for individual counting
Webcam 640x480	pre-trained mobilenet_v1 (tflite)	-	Yes	5-7	Stable for individual counting
Video 1920x1080 & 640x480	pre-trained mobilenet_v1 (tflite)	-	Yes	∼5	Counting errors, incorrect detection
Video 1920x1080 & 640x480	Fine-tuning Yolov9 (350 images, 79 epochs, mAP = 0.96) tflite16/32	-	Yes	<1	Poor performance
Webcam 640x480	pre-trained yolov8n (tflite16/32)	-	No	<1	Poor performance
Webcam 640x480	pre-trained yolov8n (tflite)	Deep sort realtime	No	<1	Poor tracking, errors
Webcam 640x480	pre-trained mobilenet_v2	pytorch + FasterRCNN	No	<1	Poor performance
Webcam 640x480	pre-trained SSD300_VGG16	pytorch	No	<1	Poor performance
Webcam 1920x1080	openCV color detection	-	Yes	∼30	Webcam testing only
Video 640x480	MOG2 background subtraction	-	Yes	∼12	Suitable for ideal input conditions only
Video 1080x1920	pre-trained yolov8s	Centroid tracking, Euclidean distance, ID management (1/3 frame rate)	Yes	0-6	∼50% error rate
Video 1080x1920	pre-trained yolov8n	Centroid tracking, Euclidean distance, ID management	Yes	0-6	Accurate
Webcam 1080x1920	pre-trained yolov8n	Centroid tracking, Euclidean distance, ID management	Yes	<1	Poor performance
Webcam 640x640	Track objects with Camshift using OpenCV	-	No	12-16	Inaccurate tracking

Tab. 2: Performance Evaluation of Various Person Detection and Tracking Models on Raspberry Pi

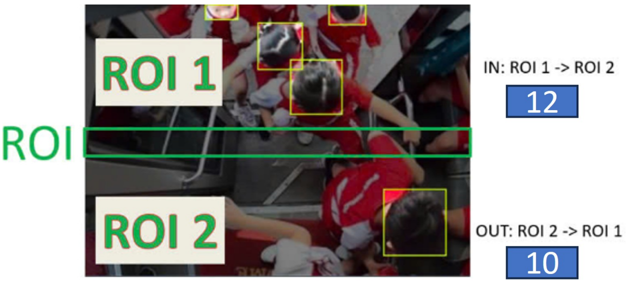


Fig. 12: the output of counting the students get in and out the school bus when deploying the system on realtime.

Model	Tracking	Objects	FPS
yolov5s	ByteTrack	0	34
		1	30
		1-5	19
		5-10	14
		>10	12.5
yolov8n	ByteTrack	0	35
		1	29
		1-5	19
		5-10	16
		>10	14
yolov11_int8	ByteTrack	0	34
		1	29
		1-5	20
		5-10	18
		>10	14

Tab. 3: Performance Evaluation of Pretrained Models on MaixCAM

corporating advanced deep learning architectures and sensor fusion techniques to improve tracking precision under varying lighting and environmental conditions. Optimizing Energy Efficiency – Exploring low-power AI hardware and model compression techniques to extend operational hours while maintaining high performance. Integrating Edge Computing – Leveraging edge AI to minimize latency and enhance real-time decision-making without relying on cloud-based processing. Implementing Privacy-Preserving AI – Adopting federated learning and anonymization techniques to protect user data while ensuring system effectiveness. Scaling and Real-World Deployment – Conducting large-scale field trials across multiple cities to validate system performance in diverse public transportation networks.

Acknowledgment

This research is funded by Ton Duc Thang University under grant number FOSTECT.2025.02

Author Contributions

Anh Vu LE, Nhat Tan LE, Anh Dung NGUYEN developed the theoretical formalism. Anh Vu LE, Ngoc Nghia NGUYEN, Hai Dang LE, Minh Dang TRAN performed the analytic calculations and performed the numerical simulations. Bui Vu MINH, Lam Dong HUYNH contributed to the software and the daft and final versions of the manuscript. Anh Vu Le. Mohan Rajesh ELARA. supervised the project.

References

[1] HSU, Y.-W., CHEN, Y.-W., and PERNG, J.-W. Estimation of the number of passengers in a bus using deep learning, *Sensors*, vol. 20, no. 8, p. 2178, 2020. DOI: 10.3390/s20082178.

[2] RAWAT, N., RAI, A., and AGARWAL, A. Deep learning-based passenger counting system using surveillance cameras, *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pp. 234–239, 2024. DOI: 10.1109/COMSNETS59351.2024.10426937.

[3] CHENG, Y., ZHAO, M., LIU, X., et al. Smart Public Transportation with AI-Based Passenger Counting. *IEEE Internet of Things Journal*, 2021, vol. 8, no. 5, pp. 3275–3286. DOI: 10.1109/JIOT.2020.3036852.

[4] MEHMOOD, Y., SHAH, R., KHAN, S., et al. Deep Learning for Intelligent Transportation Systems: A Review. *IEEE Transactions on Intelligent Transportation Systems*, 2019, vol. 20, no. 10, pp. 3828–3845. DOI: 10.1109/TITS.2019.2929020.

[5] LIYANAGE, S., et al. AI-based neural network models for bus passenger demand forecasting using smart card data. *Journal of Urban Management*, vol. 11, no. 3, pp. 365–380, 2022. DOI: 10.1016/j.jum.2022.05.002.

[6] YANG, B., et al. Edge computing-based real-time passenger counting using a compact convolutional neural network. *Neural Computing and Applications*, vol. 32, no. 9, pp. 4919–4931, 2020. DOI: 10.1007/s00521-018-3894-2.

[7] LI, F., et al. Occlusion handling and multi-scale pedestrian detection based on deep learning: A review. *IEEE Access*, vol. 10, pp. 19937–19957, 2022. DOI: 10.1109/ACCESS.2022.3150988.

[8] NGUYEN, D. T., PHAN, M. K., TRAN, P.-N., and MINH, D. D. N. Vietnamese Traffic Sign Recognition Using Deep Learning. In: *Proceedings of the 2024 9th International Conference on*

Intelligent Information Technology. ACM, 2024, pp. 30–35. DOI: 10.1145/3654522.3654528.

[9] HOWARD, A., SANDLER, M., CHU, G., et al. Searching for MobileNetV3. In: *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324. DOI: 10.1109/ICCV.2019.00140.

[10] BOCHKOVSKIY, A., WANG, C., and LIAO, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*, 2020. DOI: 10.48550/arXiv.2004.10934.

[11] GE, Z., LIU, S., WANG, F., et al. YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. DOI: 10.48550/arXiv.2107.08430.

[12] HAN, K., WANG, Y., TIAN, Q., et al. Vision Transformers for Dense Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, vol. 44, no. 9, pp. 6425–6440. DOI: 10.48550/arXiv.2103.13413.

[13] IKARAM, S., et al. A Transformer-Based Multimodal Object Detection System for Real-World Applications. *IEEE Access*, vol. 13, pp. 29162–29176, 2025. DOI: 10.1109/ACCESS.2025.3539569.

[14] NGUYEN, H. H., et al. YOLO based real-time human detection for smart video surveillance at the edge. In: *Proc. 2020 IEEE 8th Int. Conf. Commun. Electron. (ICCE)*, Phu Quoc, Vietnam, Jan. 2021, pp. 439–444. DOI: 10.1109/ICCE48956.2021.9352144.

[15] RADOVAN, A., et al. A review of passenger counting in public transport concepts with solution proposal based on image processing and machine learning. *Eng*, vol. 5, no. 4, pp. 3284–3315, 2024. DOI: 10.3390/eng5040172.

[16] NIKOUEI, S. Y., et al. Real-time human detection as an edge service enabled by a lightweight CNN. In: *Proc. 2018 IEEE Int. Conf. Edge Comput. (EDGE)*, pp. 125–129, 2018. DOI: 10.1109/EDGE.2018.00025.

[17] PRONELLO, C., et al. A low-cost automatic people-counting system at bus stops using Wi-Fi probe requests and deep learning. *Public Transport*, pp. 1–30, 2024. DOI: 10.1007/s12469-023-00349-0.

[18] LE, A. V., VO, D. T., DAT, N. T., VU, M. B., and ELARA, M. R. Complete Coverage Planning Using Deep Reinforcement Learning for Polyiamonds-Based Reconfigurable Robot. *Engineering Applications of Artificial Intelligence*, 2024, vol. 138, p.109424. DOI: 10.1016/j.engappai.2024.109424.

[19] VO, D. T., LE, A. V., et al. Toward Complete Coverage Planning Using Deep Reinforcement Learning by Trapezoid-Based Transformable Robot. *Engineering Applications of Artificial Intelligence*, 2023, vol. 122, p.105999. DOI: 10.1016/j.engappai.2023.105999.

[20] PRABAKARAN, V., LE, A. V., et al. sTetro-D: A Deep Learning Based Autonomous Descending-Stair Cleaning Robot. *Engineering Applications of Artificial Intelligence*, 2023, vol. 120, p.105844. DOI:10.1016/j.engappai.2023.105844.

[21] HOANG, Q. T., PHAM, X. H., LE, A. V., and BUI, T. T. Artificial Intelligence-Based Breast Nodule Segmentation Using Multi-Scale Images and Convolutional Network. *KSII Transactions on Internet and Information Systems (TIIS)*, 2023, vol. 17, no. 3, pp.678–700. DOI:10.3837/tiis.2023.03.001.

[22] HOANG, Q. T., PHAM, X. H., et al. An Efficient CNN-Based Method for Intracranial Hemorrhage Segmentation from Computerized Tomography Imaging. *Journal of Imaging*, 2024, vol. 10, no. 4, p.77. DOI:10.3390/jimaging10040077.

[23] LAKSHMANAN, A. K., MOHAN, R. E., RAMALINGAM, B., and LE, A. V. Complete Coverage Path Planning Using Reinforcement Learning for Tetromino-Based Cleaning and Maintenance Robot. *Automation in Construction*, 2020, vol. 112, p.103078. DOI:10.1016/j.autcon.2020.103078.

[24] KYAW, P. T., et al. Coverage Path Planning for Decomposition Reconfigurable Grid-Maps Using Deep Reinforcement Learning Based Travelling Salesman Problem. *IEEE Access*, 2020, vol. 8, pp.225945–225956. DOI:10.1109/ACCESS.2020.3045027.

[25] PRABAKARAN, V., LE, A. V., et al. Hornbill: A Self-Evaluating Hydro-Blasting Reconfigurable Robot for Ship Hull Maintenance. *IEEE Access*, 2020, vol. 8, pp.193790–193800. DOI:10.1109/ACCESS.2020.3033290.

[26] MUTHUGALA, M. A. V. J., et al. A Self-Organizing Fuzzy Logic Classifier for Benchmarking Robot-Aided Blasting of Ship Hulls. *Sensors*, 2020, vol. 20, no. 11, p.3215. DOI:10.3390/s20113215.

[27] VEERAJAGADHESWAR, P., PING-CHENG, K., ELARA, M. R., LE, A. V.,

and IWASE, M. Motion Planner for a Tetris-Inspired Reconfigurable Floor Cleaning Robot. *International Journal of Advanced Robotic Systems*, 2020, vol. 17, no. 2, p.1729881420914441. DOI:10.1177/1729881420914441.

[28] MANIMUTHU, A., LE, A. V., et al. Energy Consumption Estimation Model for Complete Coverage of a Tetromino Inspired Reconfigurable Surface Tiling Robot. *Energies*, 2019, vol. 12, no. 12, p.2257. DOI:10.3390/en12122257.

[29] PARWEEN, R., LE, A. V., SHI, Y., and ELARA, M. R. System Level Modeling and Control Design of hTetrakis—A Polyiamond Inspired Self-Reconfigurable Floor Tiling Robot. *IEEE Access*, 2020, vol. 8, pp.88177–88187. DOI:10.1109/ACCESS.2020.2992333.

[30] YI, L., et al. Reconfiguration During Locomotion by Pavement Sweeping Robot with Feedback Control from Vision System. *IEEE Access*, 2020, vol. 8, pp.113355–113370. DOI:10.1109/ACCESS.2020.3003376.

[31] DO, H., LE, A. V., YI, L., HOONG, J. C. C., TRAN, M., DUC, P. V., VU, M. B., WEEGER, O., and MOHAN, R. E. Heat Conduction Combined Grid-Based Optimization Method for Reconfigurable Pavement Sweeping Robot Path Planning. *Robotics and Autonomous Systems*, 2022, vol. 152, p.104063. DOI:10.1016/j.robot.2022.104063.

[32] CHENG, K. P., MOHAN, R. E., NHAN, N. H. K., and LE, A. V. Multi-objective Genetic Algorithm-Based Autonomous Path Planning for Hinged-Tetro Reconfigurable Tiling Robot. *IEEE Access*, 2020, vol. 8, pp.121267–121284. DOI:10.1109/ACCESS.2020.3006579.

[33] LE, A. V., NHAN, N. H. K., and MOHAN, R. E. Evolutionary Algorithm-Based Complete Coverage Path Planning for Tetriamond Tiling Robots. *Sensors*, 2020, vol. 20, no. 2, p.445. DOI:10.3390/s20020445.

[34] CHENG, K. P., MOHAN, R. E., NHAN, N. H. K., and LE, A. V. Graph Theory-Based Approach to Accomplish Complete Coverage Path Planning Tasks for Reconfigurable Robots. *IEEE Access*, 2019, vol. 7, pp.94642–94657. DOI:10.1109/ACCESS.2019.2928467.

[35] YIN, J., et al. Table Cleaning Task by Human Support Robot Using Deep Learning Technique. *Sensors*, 2020, vol. 20, no. 6, p.1698. DOI:10.3390/s20061698.

[36] LE, A. V., et al. Coverage Path Planning Using Reinforcement Learning-Based TSP for hTetran—A Polyabolo-Inspired Self-Reconfigurable Tiling Robot. *Sensors*, 2021, vol. 21, no. 8, p.2577. DOI:10.3390/s21082577.

[37] LE, A. V., et al. Reinforcement Learning-Based Energy-Aware Area Coverage for Reconfigurable hRombo Tiling Robot. *IEEE Access*, 2020, vol. 8, pp.209750–209761. DOI:10.1109/ACCESS.2020.3038905.

[38] LE, A. V., et al. Autonomous Floor and Staircase Cleaning Framework by Reconfigurable Stetro Robot with Perception Sensors. *Journal of Intelligent & Robotic Systems*, 2021, vol. 101, pp.1–19. DOI:10.1007/s10846-020-01281-2.

[39] VAN, V.-A., et al. Design of Deep Learning Model Applied for Smart Parking System. *Advances in Electrical and Electronic Engineering*, 2023, vol. 21, no. 4, pp.258–267. DOI:10.15598/aeee.v21i4.5366.

[40] GOROKHOVATSKYI, et al. Search for Visual Objects by Request in the Form of a Cluster Representation for the Structural Image Description. *Advances in Electrical and Electronic Engineering*, 2023, vol. 21, no. 1, p.19. DOI:10.15598/aeee.v21i1.4661.

[41] CHAVHAN, S., et al. Edge Computing AI-IoT Integrated Energy-Efficient Intelligent Transportation System for Smart Cities. *ACM Transactions on Internet Technology*, 2022, vol. 22, no. 2, pp.1–15. DOI:10.1145/3422473.

[42] MURSHED, G. S., et al. “Machine learning at the network edge: A survey,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–37, 2021. DOI:10.1145/3469029.

[43] RAMALINGAM, B., LE, A. V., LIN, Z., WENG, R. E., MOHAN, S., and POOKKUTTATH, S. Optimal Selective Floor Cleaning Using Deep Learning Algorithms and Reconfigurable Robot hTetro. *Scientific Reports*, 2022, vol. 12. DOI:10.1038/s41598-022-19249-7.

[44] NGUYEN, D. T., PHAN, M. K., TRAN, P.-N., and MINH, D. D. N. Vietnamese Traffic Sign Recognition Using Deep Learning. In: *Proceedings of the 2024 9th International Conference on Intelligent Information Technology*. ACM, 2024, pp. 30–35. DOI:10.1145/3654522.3654528.