2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

Enhancing Indoor Auto-Steering for AMRs through RGB and Depth Fusion

Chi-Hsuan Lee and Chih-Hung G. Li*, *Member, IEEE*

***Abstract*— This research presents a method for improving the navigation capabilities of autonomous mobile robots (AMRs) in indoor environments. Indoor navigation is challenging due to the presence of various obstacles such as floors, walls, furniture, and doors. While depth sensing devices can effectively recognize geometric conditions in corridor environments, they struggle with reflective surfaces and slim objects. Our proposed solution is to fuse depth and RGB inspections using a dual-ResNet architecture in the visual detection ConvNet. This improves performance compared to traditional depth-only approaches. Field tests have shown that our system operates at a speed of 30 frames per second and guides the AMR through various corridor routes at 2 m/s, all on an embedded PC.**

I. Introduction

Autonomous Mobile Robots (AMRs) hold great potential for logistics automation and flexible service. In particular, their ability to navigate autonomously in indoor environments is crucial as these scenarios are prevalent in human-centered work and living spaces. Previous research has developed an end-to-end auto-steering system for a self-balanced two-wheeled mobile robot using a deep Convolutional Neural Network (ConvNet) to map the depth image of the front view to direct steering actions [1, 2]. The approach showed preliminary success in navigating through office corridors and fleets of machine tools. However, it was found that the depth camera had limitations, such as the difficulty of properly capturing glass doors and mirror walls and a higher failure rate when encountering lower fences or slim structures that are perceivable in color but easily dictated by depth noise. Additionally, the previous ConvNet controller, which was trained to classify only three types of scenarios [2], had difficulty recognizing and properly reacting to more complex real-world situations.

To address these limitations, this study proposes a four-layer framework including the inspection, intelligence, task, and command layers to enhance the performance of the auto-steering system. By recruiting both RGB and depth pixels, as shown in Fig. 1, and designing the ConvNet architecture to accommodate two independent visual inputs for convolutional feature extraction, the input dimensionality is increased. The network then fuses the extracted features and performs integrated network classification for an output of approximately twelve typical visual scenarios. The more stable RGB input helps to capture presentations of untraversable

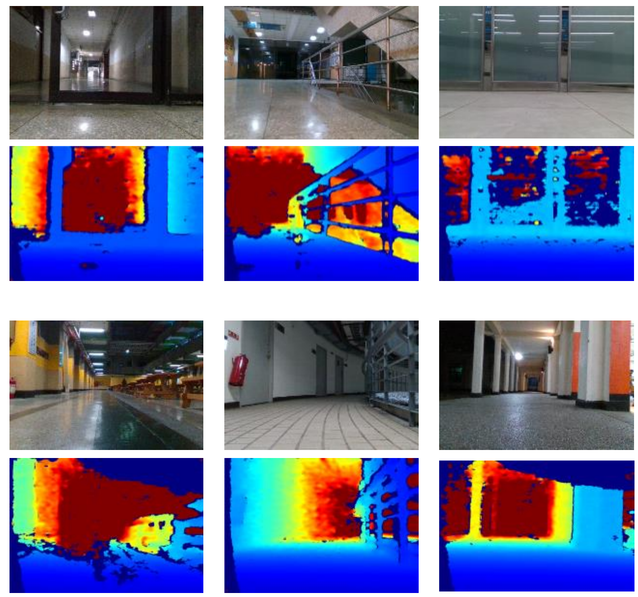


Fig. 1. The proposed end-to-end auto-steering system adopted both RGB and depth inputs for an integrated deep ConvNet which improves the AMR's navigation performance.

obstacles such as low decorations and glass structures as well as identify slim structures such as iron railings. As the intelligence layer accurately inspects and characterizes the field situations, more delicate manipulation of the robot is possible, resulting in more successful navigations in various environments. The proposed framework was implemented on a mobile robot J4.β [3], and a series of field tests were conducted to verify its performance.

The main contributions of this research are:

(1) A new RGB-D fusion ConvNet for end-to-end auto-steering in indoor corridor environments, which addresses the limitations of previous depth-only approaches.

(2) An evaluation of the proposed visual detection framework, which demonstrates its superiority over previous approaches.

*This work was supported by the Ministry of Science and Technology of the Republic of China, Taiwan, under Contract No.: MOST 111-2221-E-027-106-MY2.

Chi-Hsuan Lee is with the Graduate Institute of Mechatronic Engineering, National Taipei University of Technology, Taipei, 10608 Taiwan ROC (phone: +886-2-2771-2171 ext. 2080; fax: +886-2-2776-4889; e-mail: t110408096@ntut.org.tw@ntut.org.tw).

Chih-Hung G. Li is with the Graduate Institute of Manufacturing Technology, National Taipei University of Technology, Taipei, 10608 Taiwan ROC (phone: +886-2-2771-2171 ext. 2092; fax: +886-2-2776-4889; e-mail: cL4e@ntut.edu.tw).

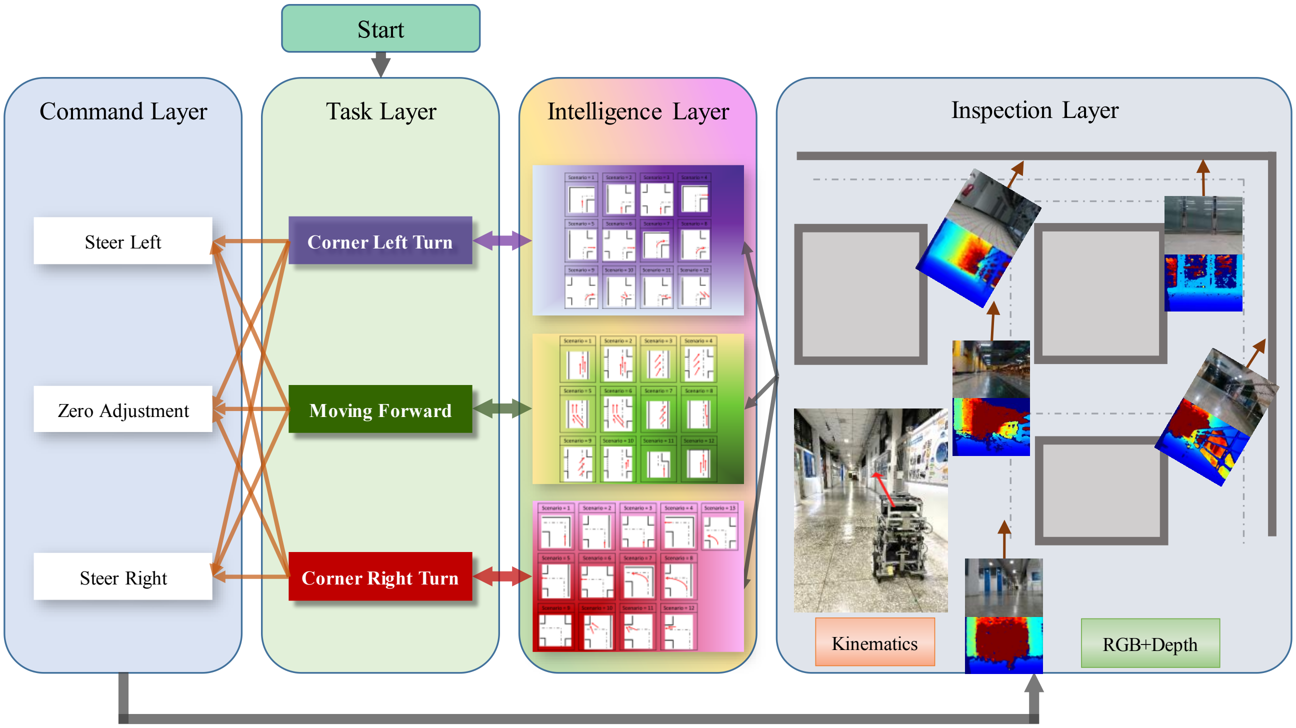


Fig. 2. Illustration of the four-layer task-oriented auto-steering framework.

(3) A thorough examination of the performance of the proposed AMR navigation system through extensive field tests.

II. Related Work

Autonomous navigation is a crucial capability for AMRs. Among various approaches, Simultaneous Localization And Mapping (SLAM) has received significant attention and has led to the development of numerous models [4 - 7]. SLAM systems use sensors to perceive the environment and estimate the robot's position simultaneously. Mapping is a critical step in SLAM, as an accurate geometric map helps reduce the localization error of the robot and plan the navigation path. However, one drawback of this approach is that it can require a large amount of data storage and significant computational resources.

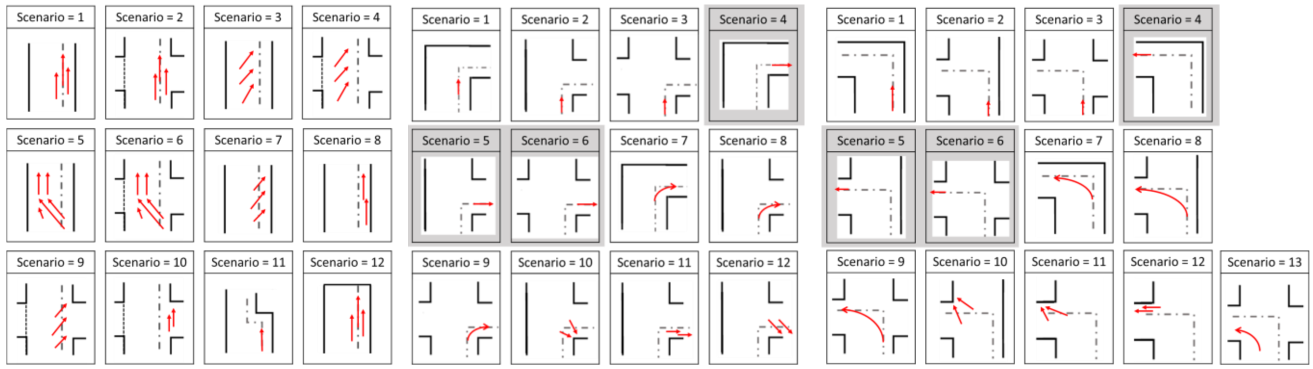
Recent advances in deep learning have opened up opportunities for more accurate perception through computer vision. End-to-end actions from vision to steering have become more promising [8-12]. The autonomous navigation task was divided into two parallel modules - topological localization and automatic steering [1, 2]. While the automatic steering module focuses on making sound judgments and actions in response to various field conditions, the topological localization module [13, 14] uses ConvNet-based scene recognition to indicate the robot's location. As the topological map does not carry detailed geometric features, the model is much lighter and easier to operate in real-time.

Chen and Li [15] developed a task-oriented steering framework to generate suitable actions for more general scenarios. Navigation tasks were categorized as moving forward, corner left turn, and corner right turn. When a specific task is issued, the ConvNet-based visual detector examines the depth condition in front of the robot and outputs one of the three possible actions, steering left, zero adjustments, and steering right. Each navigation task has its ConvNet detector, which was trained with abundant field images reflecting all possible navigation scenarios in indoor corridor environments.

Our work aims to improve the model's performance developed by [15] by incorporating two major differences. First, we extended the navigation system to a four-layer architecture, with the intelligence layer focusing on inspecting more specific scenarios. Instead of directly outputting three action options [15], the new framework classifies the view into approximately 12 scenario types, which later prompt suitable action commands by the action maker. Secondly, we included both RGB and depth signals to increase the input dimensionality. The new ConvNet architecture involved two ResNet-18 [16] pipelines, each extracting designated features before merging for final classification. By separating action-making from the ConvNet, the visual detector is able to better focus on visual classification and improve detection results.

III. Method

In this study, we proposed a four-layer task-oriented framework for autonomous navigation of AMRs, which aims to enable the robot to divide a routing job into fundamental navigation tasks such as Moving Forward, Corner Left Turn,



(a) Moving Forward (b) Corner Right Turn (c) Corner Left Turn

Fig. 3. Scenario classifications of the three navigation tasks. Indistinguishable scenarios, indicated by the grey background, are grouped under a single label.

and Corner Right Turn, where sound judgments and can be made in response to various field conditions (see Fig. 2).

We identified approximately a dozen general scenarios for each navigation task, which would lead to specific actions. To achieve accurate visual inspection, we collected up to 250k training images and built an RGB-D fusion ConvNet using a dual-ResNet-18 architecture. This architecture allows for increased input dimensionality and improved feature extraction, resulting in more accurate visual classification.

An action maker then determines the steering actions based on the inspection results and the robot's kinematics. The commands issued by the action maker activate the robot's actuators for propulsion, thus forming a closed-loop control scheme for autonomous visual-servo navigation.

A. *Task-Oriented Auto-Steering*

The proposed system is primarily intended for indoor corridor environments, typically consisting of passages guided by walls, interiors, or large furniture. The navigation pattern in such environments can be broken down into three basic tasks: moving forward, turning left at corners, and turning right at corners. We approach the execution of a complete routing job as a series of fundamental tasks. Thus, the action policy π_l, which determines the proper reactions u_i to respond to certain environmental and robot conditions under the current navigation task, depends on the navigation task l and includes an action maker $\tilde{\pi}_l$, and two sensory units Γ_l and K. Specifically, Γ_l receives the front views I_i and classifies them into one of the visual categories ω_i, while K measures the robot's kinematic characteristics k_i at the current state s_i,

$$\pi_l(u_i|I_i,s_i) = \tilde{\pi}_l(u_i|\omega_i,k_i)\Gamma_l(\omega_i|I_i)\mathrm{K}(k_i|s_i) \quad (1)$$

where i denotes any instance during the routing job.

The trajectory τ of a routing job is the result of the successive execution of navigation tasks, with each task taking into account the sensory probabilities,

$$\pi(\tau) = p(x_1)\prod_{i=1}^{N} p(x_{i+1}|x_i,u_i) \iint \pi_l(u_i|I_i,s_i)p(I_i,s_i|x_i)dI_i ds_i \quad (2)$$

where x_i denotes the robot's pose at the current instance.

In the four-layer design illustrated in Fig. 2, the first step is to select one of the three navigation tasks to be executed. The interoceptive and exteroceptive devices on the robot then send signals to the intelligence layer, where scenarios are classified under the designated task. Steering actions are generated according to the intelligence outputs; depending on the scenario and the robot's current speed, $\tilde{\pi}_l$ generates commands of different steering actions with various amplitudes.

Γ_l classifies N scenarios for each navigation task, with 12 scenarios defined for Moving Forward, 13 for Corner Left Turn, and 12 for Corner Right Turn. These scenarios, illustrated in Fig. 3, include a target path for the robot to follow and adhere to social movement norms, such as staying on the right side of a corridor. Four common structures are present in the scenarios: closed passages, T-junctions, crossroads, and dead-end corners. The robot is placed in different poses within these structures, and similar scenarios are grouped into a single category. For example, Scenario #5 of Moving Forward depicts the robot veering to the left of the target path, necessitating a corrective steering to the right. A detection label is established within the ConvNet for each scenario, with indistinguishable scenarios from the front view grouped together under a single label, as seen in the Corner Left and Right Turn categories (see Fig. 3).

B. *RGB-D Fusion ConvNet*

We adopted an RGB-D camera to capture RGB and depth images simultaneously, which serve as the visual input for the intelligence layer in Fig. 2 to generate an accurate interpretation of the front-view condition. Each type of image is then inputted into a ResNet-18 pipeline for feature extraction, as shown in Fig. 4. The resulting vectors are concatenated and passed to a fully connected layer, producing an output of N categories.

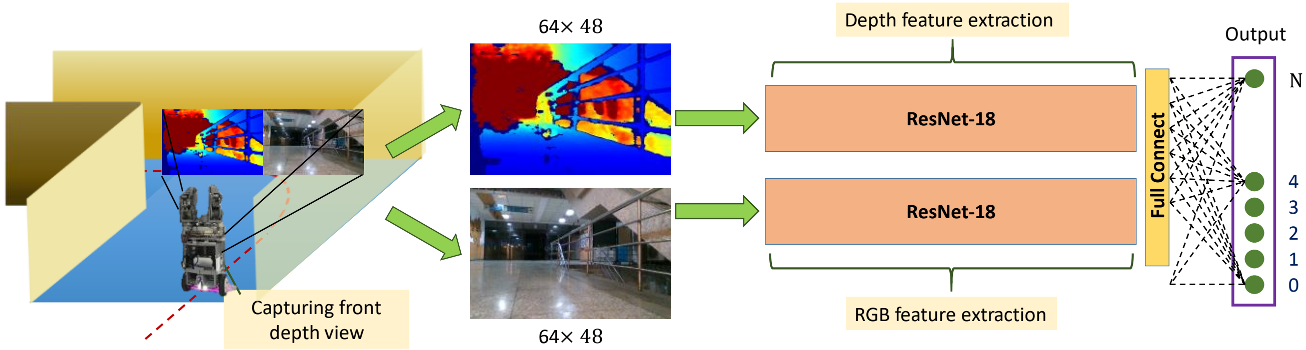


Fig. 4. The architecture of the dual-input ConvNet designed for combining RGB and depth information.

To acquire training images for the ConvNet, we manually operated the robot in various corridor environments and recorded the front views at different scenarios, as shown in Fig. 5. The images were then divided into N groups, each belonging to a specific scenario. The training image set, V contains a total of m sets of images, with each set containing an RGB and a depth image that can be categorized in N scenarios,

$$V : (\boldsymbol{v}_1^c, \boldsymbol{v}_1^d, y_1), \cdots, (\boldsymbol{v}_m^c, \boldsymbol{v}_m^d, y_m) \in \mathbb{R}^{2q} \times \{\sigma_1, \cdots, \sigma_N\}. \quad (3)$$

where $\boldsymbol{v}_j^c$ denotes a vector of q features corresponding to the jth RGB image, $\boldsymbol{v}_j^d$ denotes a vector of q features corresponding to the jth depth image, and y_j denotes the class of the jth image. After training, the model assigns each image to its corresponding category,

$$\Gamma : \mathbb{R}^{2q} \longmapsto \{\sigma_1, \cdots, \sigma_N\}. \quad (4)$$

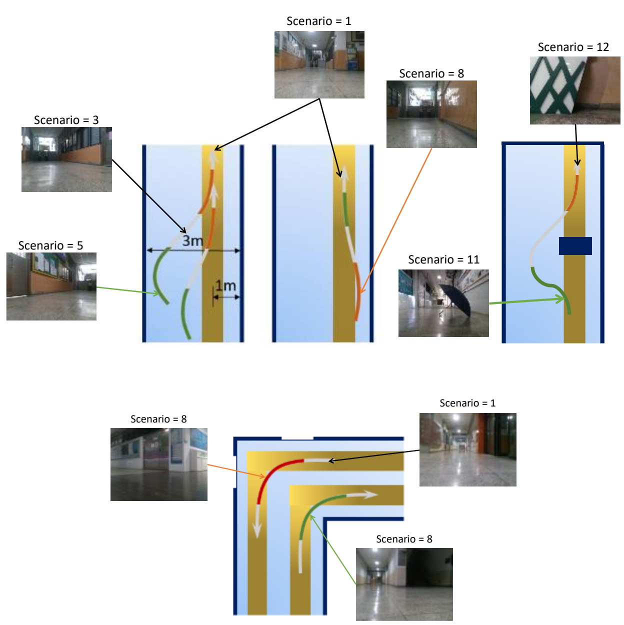


Fig. 5. Samples of the training annotations.

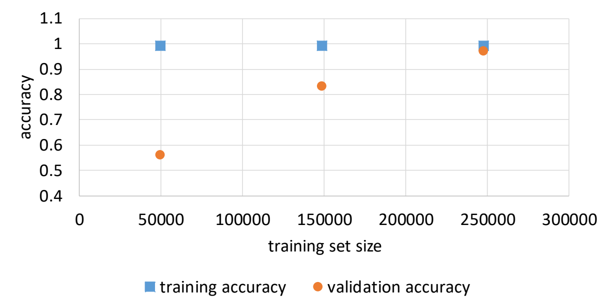
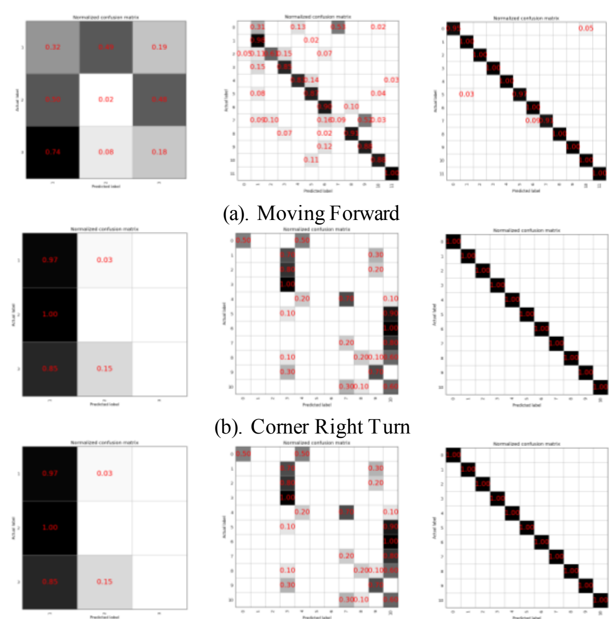


Fig. 6. Training results of the ConvNet classifier as a function of the training set size.

The goal of the model training is for the predicted category at any instance i to be as close as possible to the ground truth $y_{i/g}$,



(a). Moving Forward

(b). Corner Right Turn

(c). Corner Left Turn

Fig. 7. A comparison of scenario detection results by various models. From left to right: model of [2], single input of depth images, and the proposed RGB-D framework.

TABLE I. COMPARISON OF THE SPEED PERFORMANCES

	Raspberry Pi 4 + TensorFlow	IPC + TensorFlow	IPC + IR
Photo-taking	30 fps	30 fps	33 fps
Model execution	20 fps	43 fps	145 fps
Whole process	2 fps	20 fps	30 fps

$$y^{i^*} = \underset{y^i}{argmin}\left\| \underset{n}{argmax}\left(\Gamma_l\left(v_i^f \oplus v_i^d, \sigma_n\right)\right) - y^{i/g}\right\|_2. \quad (5)$$

An examination of multiple training iterations showed that the validation accuracy improves with an increase in the size of the training set. As seen in Fig. 6, when the total number of training images was 50,000, the validation accuracy of the model was 60%, indicating a 40% gap in relation to the training accuracy. However, when the total number of training images increased to 250,000, the validation accuracy became nearly equal to the training accuracy, approaching 100%.

C. The Robot

We implemented the proposed navigation system on the mobile robot, J4.β [2, 3, 15], which has a compact design, dynamic balancing capabilities, high velocity, and adaptability to various terrains. As a self-balanced two-wheeled mobile robot, J4.β changes its direction through differential speed. The output signal of the action maker was fed to the controller of the mobile platform, and the original handlebar was replaced by an electronic signal generator. The original version of J4.β adopted a Raspberry Pi 4 for the computation of the deep neural networks. However, the original depth-only scheme generated outputs at approximately 3 fps, while the dual ResNet-18 framework produced a lower output rate of approximately 2 fps. To overcome this limitation, we adopted an embedded system with an Intel 9th gen i7 CPU and 16 GB DDR4 for the RGB-D fusion ConvNet, resulting in a higher inference speed of 20 fps. By converting the TensorFlow model to an IR model using the Intel OpenVINO Toolkits, the inference speed was further increased to around 30 fps. The performance comparison of different embedded systems is presented in Table I.

IV. EXPERIMENTS

The aim of this experiment is twofold: firstly, to assess the accuracy of the proposed framework in recognizing objects in diverse scenarios, and secondly, to determine its practicality for deployment on an AMR in real-world situations. The neural network's ability to generalize was evaluated by collecting photographs of road segments distinct from those in the training set and using them as the test set. Furthermore, to evaluate the framework's actual ability during navigation, we identified and tested it on 10 unvisited road segments.

A. Scenario Prediction Accuracy

To validate the effectiveness of the proposed method, we further compared the performance of the proposed framework with a single-input ConvNet and the three-label architecture described in [2]. The comparison was conducted using the same test set. The single-input ConvNet possesses the same output structure, but it only receives depth images as input. The model in [2] collapsed all scenarios into three action labels: steer left, zero adjustments, and steer right. The results are summarized in Fig. 7, where the proposed framework demonstrated superior performance compared to the other two models. It is worth noting that the models in [2] were trained with approximately 1000 depth images, significantly less than what was used for the proposed model. The single-input models were trained with the same training sets as the proposed model, except the RGB inputs were removed.

B. Navigation Success Rate

To evaluate the feasibility of AMR navigation in various corridor environments, we examined a total of 10 road sections, each with different widths and features and requiring correct detections of various scenarios to complete a route (see Fig. 8). Each section was tested ten times; a successful case was recorded when the robot completed the entire route without deviating significantly from the target path. The results are summarized in Fig. 9, revealing a 100% success rate in ordinary walled corridors, such as sections B, C, and D. Even when encountering an intersection or T-junction, the robot can smoothly pass through the area. Sections A and H presented more challenges, with the presence of railings in A and a narrow aisle in H. Nevertheless, J4.β successfully

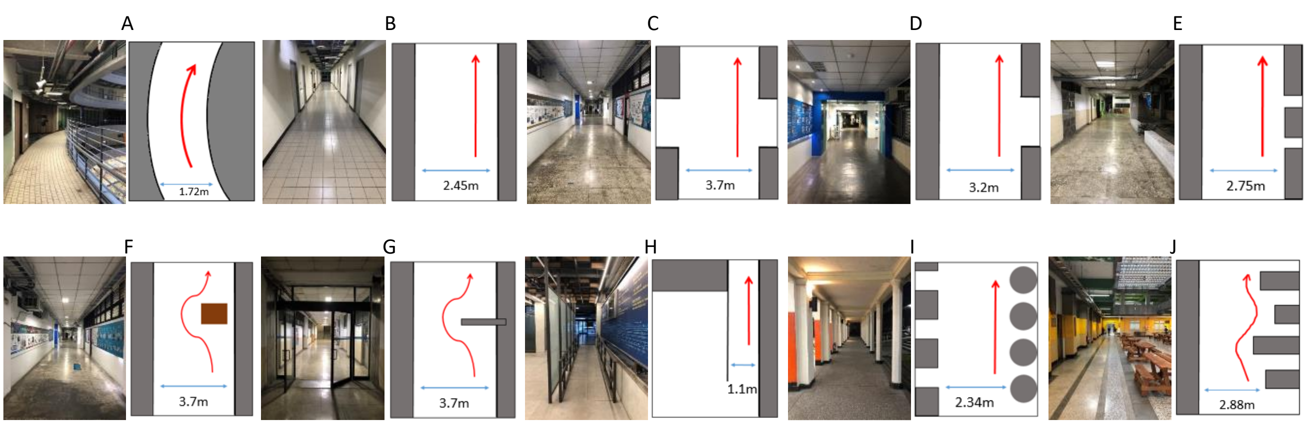


Fig. 8. Illustrations of the testing fields and routes.

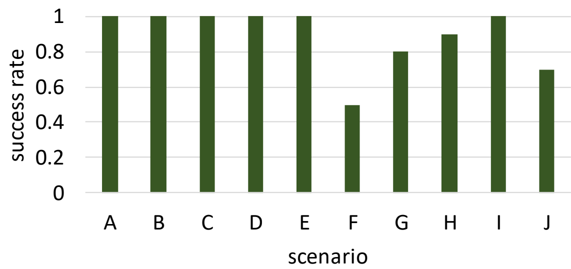


Fig. 9. Success rates of various testing routes.

navigated the circular arc of section A, although it was never trained with scenarios of circular passages. With a 10% probability, failure occurred in H due to its low fault tolerance; however, this is a significant improvement from previous models.

In addition, special corridor settings such as sections E, I, and J consisting of low walls, pillars, and picnic tables were tested. The results showed that J4.β had no problem navigating through sections E and I, with a success rate of 100% for both cases. However, the success rate for section J dropped to 70%, as the robot might take a route that is too close to the picnic tables. This may be due to the low and discontinuous profiles of picnic tables being underrepresented in the input and resulting in misclassification.

The obstacle avoidance capabilities of the AMR were tested in sections F and G, where a small cardboard box and a glass door were respectively presented. Section F had a success rate of 50%, while section G had a success rate of 80%, indicating that the AMR was able to detect obstacles that were previously missed. Although the success rate was not 100%, the issue generally arose when returning to the target path after bypassing the obstacle. This problem could potentially be resolved by implementing more delicate maneuvering during that process. A video can be viewed at https://www.youtube.com/watch?v=jdfM_j9ZaD8.

V. Conclusion

this research proposes a novel visual method for enhancing the navigation of AMRs in indoor environments. The task-oriented framework breaks down the routing task into fundamental movement tasks and is supported by a four-layer structure that addresses the challenges posed by diverse features in indoor environments. By fusing depth and RGB inspections with a dual-ResNet architecture, the intelligence layer improves visual classification. The proposed framework showed superior performance over depth-only architectures and previous models, successfully completing previously impossible navigations in side-by-side evaluations and field tests.

Table II presents a concise comparison of the characteristics of RGB-D cameras, 2D LiDARs, and 3D LiDARs. RGB-D cameras are significantly less expensive than LiDARs and offer joint color and space detection capabilities, enabling them to detect meaningful color features and identify structures on the road. However, LiDARs provide wider field of views and are less susceptible to light interference.

References

[1] C. G. Li and L.-P. Zhou, "Training end-to-end steering of a self-balancing mobile robot based on RGB-D image and deep ConvNet," in *Proc. 2020 IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM 2020)*, 2020, pp. 898–903.

[2] C. G. Li, L.-P. Zhou, and Y.-H. Chao, "Self-balancing two-wheeled robot featuring intelligent end-to-end deep visual-steering," IEEE/ASME Transactions on Mechatronics, vol. 26, no. 5, pp. 2263-2273, 2021

[3] Y.-C. Hsu, M.-C. Lin, and C. G. Li, "Mobility improvement on the two-wheeled dynamically balanced robot – J4.β," in *Proc. IEEE Int. Conf. Auto. Sci. Eng., (CASE)*, 2021, pp. 442–447.

[4] S. Wang et al., "A localization and navigation method with ORB-SLAM for indoor service mobile robots," in Proc. *IEEE Int. Conf. Real-time Comput. Robot.*, 2016, pp. 443-447.

[5] A. Singandhupe and H. M. La, "A Review of SLAM Techniques and Security in Autonomous Driving," 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 2019, pp. 602-607

[6] A. R. Khairuddin, M. S. Talib and H. Haron, "Review on simultaneous localization and mapping (SLAM)," 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2015, pp. 85-90

[7] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. Robotics Research*, vol. 21, pp. 735–758, 2002.

[8] Dean A. Pomerleau, "Knowledge-based training of artificial neural networks for autonomous robot driving," In J. Connell and S. Mahadevan, editors, Robot Learning. Kluwer Academic Publishing, 1993.

[9] T. Jochem, D. Pomerleau, and C. Thorpe, "Vision-based neural network road and intersection detection and traversal," in *Proc. IEEE Conf. Intelligent Robots and Systems*, 1995, vol. 3, pp. 344–349.

[10] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 739–746.

[11] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: learning affordance for direct perception in autonomous driving," in *Proc. Int. Conf. Comput. Vision*, 2015, pp. 2722–2730.

[12] Tai, L., Liu, M. Mobile robots exploration through cnn-based reinforcement learning. Robot. Biomim, vol. 3, 24, 2016.

[13] Y.-F. Hong, Y.-M. Chang, and C. G. Li, "Real-time visual-based localization for mobile robot using structured-view deep learning," in *Proc. IEEE Int. Conf. Auto. Sci. Eng., (CASE)*, 2019, pp. 1353–1358.

[14] C. G. Li, Y.-F. Hong, P.-K. Hsu, and T. Maneewarn, "Real-time topological localization using structured-view ConvNet with expectation rules and training renewal," *Robot. Auto. Syst.*, vol. 131, https://doi.org/10.1016/j.robot.2020.103578, 2020.

[15] W.-C. Chen, C.-H. G. Li, "Task-oriented automatic steering for AMR utilizing depth vision deep learning," in *Int. Conf. Adv. Robot. Intell. Syst. (ARIS 2021)*, 2021, #1060.

[16] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2016, pp. 770-778.

Table II A Comparison of RGB-D, 2D-LiDAR, and 3D-LiDAR

Component	**RGB-D**	**2D-LiDAR**	**3D-LiDAR**
Cost (USD)	150~350	1000~7000	5000~12000
Space capability	3D	2D	3D
Color capability	high	no	no
Light interference	yes	no	no
Update rate	12~50fps	10~20fps	12.5~50fps
Field of view	small	large	large