International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 3, June 2025, pp. 2748~2757
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i3.pp2748-2757

Indoor navigation for mobile robots based on deep reinforcement learning with convolutional neural network

Khoa Nguyen Dang[1], Van Tran Thi[2], Nguyen Van Thang[3]
[1]Faculty of Engineering and Technology, International School, Vietnam National University, Hanoi, Vietnam
[2]Faculty of General Education, University of Labour and Social Affairs, Hanoi, Vietnam
[3]University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

Article Info

Article history:

Received Jul 15, 2024
Revised Dec 23, 2024
Accepted Jan 16, 2025

Keywords:

Convolutional neural networks
Deep Q-network
Gazebo
Line tracking
Mobile robot

ABSTRACT

The mobile robot is an intelligent device that can achieve many tasks in life. For autonomous, navigation based on the line on the ground is often used because it helps the robot to move along a predefined path, simplifies the path planning, and reduces the computational load. This paper presents a method for navigating the four-wheel mobile robot to track a line based on a deep Q-network as a control algorithm to desire the action of the mobile robot and a camera as a feedback sensor to detect the line. The control algorithm uses a convolution neural network (CNN) to generate the mobile robot action, defined as an agent of deep Q-network. CNN uses images from the camera to define the state of the deep Q network. The simulations are performed based on Gazebo software which includes a 3D environment, mobile robot model, line, and Python programming. The results demonstrate the high-performance tracking of mobile robots with complex line trajectories, achieving errors of less than 100 px, which is compared with the traditional vision method (VNS), the MSE of the proposal method is 0.0264 lower than VNS with 0.0406. Showcases proved convincingly that effectiveness suggested a control approach.

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

Corresponding Author:

Khoa Nguyen Dang
Faculty of Engineering and Technology, International School, Vietnam National University
144 Xuan Thuy Road, Cau Giay, Ha Noi 100000, Vietnam
Email: khoand@vnuis.edu.vn

1. INTRODUCTION

Mobile robots (MR) have been rapidly developing in many fields in life such as industry [1], military [2], and medicine [3] with the purpose of transportation [4], assistance [5], rescue [6], cave exploration [7], and operation in polluted environments [8] to check the pollution level of the environment. Because they are very flexible, perform many tasks, are autonomous, and can free up human hands. For each mission, MR is designed with different types as two-wheel mobile robots (TWMR), and four-wheel mobile robots (FWMR), [9], [10]. Each type has advantages and disadvantages in the specific applications. For example, TWMR is often small in size, and small payload [11]. While FWMR has more rigidity, larger payload, better drivability, and more stability in cornering [12]. In general, the FWMR is often used to demonstrate control algorithms as well as apply them to real life. Additions, the FWMR is similar to a car vehicle, and autonomous vehicle systems are of interest both indoors and outdoors nowadays. With outdoor, the vehicle could use the global positioning system for navigating. However, the indoor is lacks this, which has to use the landmark or road markings to compute its position and tracking.

By using a global sensor like a camera [13], the broadcasting signal is sent to all mobile robots, and the position of MR is detected based on the broadcast control. This system could support the position for

many MRs in this network but the configuration complex and the low signal transfer affect to performance of each MR. To localized control MR, energy consumption comparison [14] relating to tracking accuracy is used to strict criterion. This work shows the high performance of MR in tracking position field, but maintaining energy and its comparison are difficult work. It needed high accuracy to decide the controller for MR. If this performance is reduced for some reason, the trajectory of MR is affected and does not follow the purpose. Other research related to the control of each actuator with dead-zone input [15] is to maintain the trajectory of MR, which could prove the stability of the Lyapunov function but it is difficult to apply them to real devices and environments. Besides, an adaptive control algorithm based on sliding mode control [16] was developed for the lateral stability of mobile robots in both simulation and experimental. Herein, the MR could well track the S path trajectory. However, the vibrated appears in the yaw rate and slip angle in the fast update. Normally, the control algorithms based on landmarks are complex and need more sensors to detect the environment. Therefore, using line markings such as color lines and magnetism lines to guide MR moves to the target is often considered in many research and applications. Herein, the cost of the magnetism line and its sensor for detecting are more expensive than the color line. In this paper, we select the black line on the ground as the marks for the MR.

The designing automatic navigation algorithms for MR based on the black line have been implemented by using an infrared sensor (IR) [17] and a camera sensor [18]. Both sensors deliver high effectiveness for each application. However, IR is often affected by light intensity and the condition of the environment, while the camera has more useful information by the larger view of the point. A combination of artificial intelligence (AI) trends is developing in all fields of life. Then AI helps MR to navigate lines is focused in some research [19], [20] which uses an artificial neural network (ANN) to compute the speed of each actuator of MR. With this structure, the dataset for training the weight of ANN needs a larger size and is defined in detail. To overcome this limitation, reinforcement learning (RL) is the most used in automatic navigation because of its self-learning, exploration, and discovery of the environment [21]. However, it has limitations related to the memory in the continuous environment due to the self-learning process which could be solved by a combination of RL and deep learning (DL) called deep reinforcement learning (DRL) which uses the loop between agent and environment to finish the control tasks. Herein, environment provides the State and reward-like feedback control for the agent. After having this information, the agent will generate the action corresponding to send back to the environment like the control system. A table in agent to map between state input and action output. However, the memory space needs to be large and quickly accessible. For this reason, a neural network (NN) is used to replace the mapping table which is called a deep Q network (DQN) [22].

To present NN in DQN for controlling the mobile robot, the ANN [23], single shot multibox detector [22] and convolutional neural network [24] have been considered using the red, green, blue-depth (RGB-D) and lidar sensors to detect the marks in the environments. So, the DQN in the navigation field for a mobile robot based on line tracking is a new point in this paper. In particular, CNN is an architecture in deep learning that could reach high accuracy in classification and segmentation based on images from a camera [25]. In this paper, the camera is used to capture the image from the environment and analyze it to determine the control signal for MR based on DRL. And CNN is selected for the DQN structure in DRL. Furthermore, the mobile robot must be simulated before applying it to real experiments to avoid trouble-related control algorithms, parameters, and environmental conditions. The simulation step could help reduce the cost of the development product as well as test the control algorithm. This research uses the image from the camera for training the CNN structure to make it more intelligent. For simulation, the Gazebo simulation software is suggested to present the environment, sensor model (camera), and mobile robot model based on physics engines [26].

In this paper, we propose to use the classical DQN structure in DRL to generate action for navigating the mobile robot with one RGB camera to follow the line. The DQN is developed based on the CNN structure. And simulation is built based on Gazebo software to present the 3D environments, modeling mobile robots, and sensors (camera). The simulation results show the effectiveness of the proposal method that MB could track the black line with small errors (less than 100px) and complex desired trajectory.

The rest of the paper is presented as follows: section 2 is mobile robot modeling while the overall RL and DQN algorithm are presented in section 3. Next, the proposed control algorithm for mobile robots using DQN agent by the convolution neural network is developed in section 4. Section 5 presents results obtained in the Gazebo simulation environment to verify the control performance and section 6 is the conclusion.

2. METHOD

2.1. Mobile robot modeling

A four-wheeled mobile robot (FWMR) consists of four wheels with a radius of *r*. Herein, two wheels in each front and rear must be linked to ensure they have the same speed to prevent the mobile robot

from slipping. The mobile robot model and its geometry are presented in Figure 1 [1], which could be detected by the link between a global frame (X, Y) and a local frame (x, y); and the orientation α.

The linear velocity of each wheel is computed as $v_{ix} = r_i\omega_i$ [1]. Where v_{ix}, r_i, and ω_i are linear velocity, radius, and angular of the i^{th} wheel (i=1..4) along the x-axis, respectively. Let's define the linear velocities of the left (v_L) and right side (v_R) of the mobile robot as $v_L = v_{1x} = v_{4x}$ and $v_R = v_{2x} = v_{3x}$.

The linear velocity (v_x, v_y) and angular velocity (w) of the vehicle along the *x-axis* could be represented as $v_x = \frac{\omega_L+\omega_R}{2}$, $\omega = \frac{-\omega_L+\omega_R}{c}$, and $v_y = 0$. Where $2c$ is the distance between two wheels W_1 and W_2 in Figure 1 and $\begin{bmatrix}\omega_L \\ \omega_R\end{bmatrix} = \frac{1}{r}\begin{bmatrix}v_L \\ v_R\end{bmatrix}$. The target of control is to move the mobile robot to a target defined as X, Y, and orientation. They are presented by the kinematic equation of the FWMR as in (1).

$$\begin{bmatrix}\dot{X} \\ \dot{Y} \\ \dot{\alpha}\end{bmatrix} = \begin{bmatrix}cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}v_x \\ v_y \\ \omega\end{bmatrix} \quad (1)$$

In this paper, the FWMR model is used to demonstrate the tracking of the line algorithm based on deep reinforcement learning with DQN type. Herein, DQL uses the CNN structure with image input which is captured from the camera. DRL is to generate the Action as the control signal to the mobile robot.

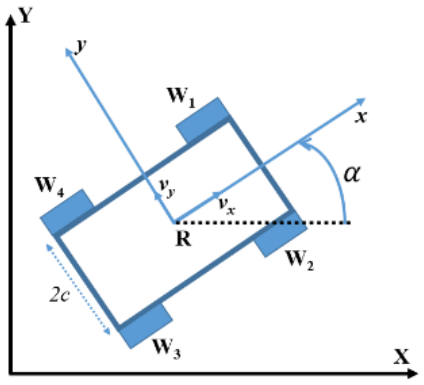


Figure 1. Four-wheeled mobile robot modeling

2.2. Reinforcement learning and deep Q-network

In reinforcement learning (RL), the relationship between the agent and environment is shown in Figure 2. Herein, the agent receives states (s_t) and reward (r_t) from the environment, which will select the action (a_t) based on s_t and policy π as in (2) [27]. Term ε is the epsilon coefficient which plays a critical role in managing the trade-off between exploration (agent trying out new actions to discover their effects and rewards) and exploitation (agent using its current knowledge to maximize reward), enabling the agent to learn effectively from its environment, a_t is the selected action, $arg\max(.)$ is the index of the maximum value and $Q(s_t)$ is the value at s_t. The a_i is obtained by finding the max of $Q(s_t, a_i)$ of s_t in Table. 1.

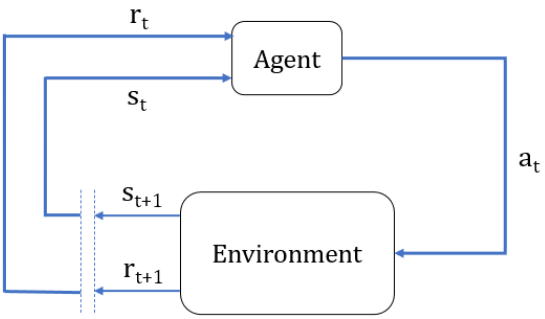


Figure 2. Reinforcement learning structure

$$a_t = \begin{cases} a_i \; at \; arg\max(Q(s_t)) & if \; random\,[0,1] < \varepsilon \\ a_r \; at \; int(random[1,m]) & if \; random\,[0,1] \geq \varepsilon \end{cases} with \; i,r \; \in [1,..,m] \quad (2)$$

$$Q(s_t, a_t) = r_{t+1} + \gamma \max\,(Q(s_{t+1}, a_{t+1})) \quad (3)$$

In Table 1, $[s_1, a_1, \ldots, s_n, a_m]$ are state-action pairs with n states and m actions, respectively. $Q(s_t, a_i)$ is updated based on the Bellman equation [3] in (3), where γ is the discount factor with a value defined from 0 to 1. Bellman equation provides a recursive decomposition of the value function, which is central to solving decision-making problems where an agent seeks to maximize cumulative rewards over time. However, the Q-table is limited to being stored in memory and structure organization, which could not save large rows and columns. To overcome this point, the neural network is a replacement method with the input layer (states) and output ($Q(s_t, a_t)$) which is called deep Q-network (DQN) as in Figure 3.

This paper uses the camera to detect the line definition on the ground based on the neural network typed convolution neural network (CNN). It is applied to find the policy of RL as in Figure 3, which is well known as a class of deep neural networks, most commonly applied to analyzing visual imagery. In the navigation of mobile robots, CNN could be used to generate the stage to select actors of DRL. The details are presented in the next section.

Table 1. Relationship between action and state

	s_1	s_2	…	s_n
a_1	$Q(s_1, a_1)$	$Q(s_2, a_1)$	…	$Q(s_n, a_1)$
a_2	$Q(s_1, a_2)$	$Q(s_2, a_2)$	…	$Q(s_n, a_2)$
…	…	…	…	…
a_m	$Q(s_1, a_m)$	$Q(s_2, a_m)$	…	$Q(s_n, a_m)$

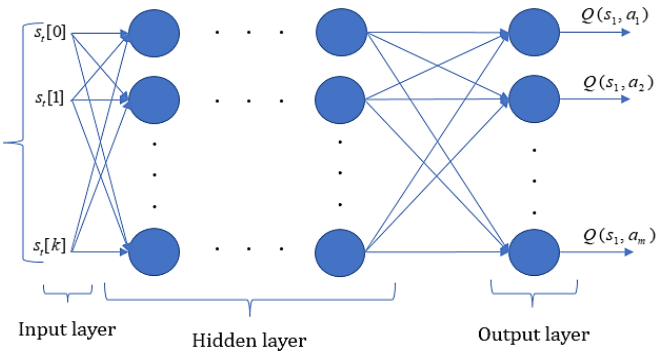


Figure 3. Neural network replacing for Q-Table in RL

2.3. Developing deep Q-network for mobile robot

In this section, we represent the architecture of DQN which contains 3 parts: preprocessor, CNN layer, and action set as shown in Figure 4. State output from the environment is the image description with three parameters the width, the height, and the color. In this case, the width and height are 480 pixels (px), and 360 pixels, respectively. The color is presented with three units by red-green-blue (RGB). To simplify the network architecture of CNN, instead of using an RGB image with full color (3 channels), the image will be converted to the gray image (defined with the single channel) where each pixel is presented by a single intensity value (from 0 to 255).

Therefore, the image could be presented as (480×360×1) which is resized to (84×84) to reduce the number of pixels from the image and present with (84×84×1) as input for CNN. This stage is to reduce the computation complexity and time training of neural networks. The architecture of CNN consisting an input layer of size 84×84×1 as in Figure 5. Then three convolution layers are 32 filters of the kernel matrix 8×8 with 4 strides, 64 filters of the kernel matrix 4×4 with 2 strides, and 64 filters of kernel matrix 3×3 with one stride are applied to filter the information and produce a feature map and then they are converted to the single dimensional vector [28]. The data is sent to the hidden layer with 128 neurons.

Both convolution layers and hidden layers use rectified linear units (ReLU) for the activation function. The output layer with three outputs presents three actions turn left (L0.5 is 0.5 rad/s), forward (F0.2 is 0.2 m/s), and right (R0.5 is 0.5 rad/s) as in (4). The linear activation function is applied to align output in range.

$$action = \begin{cases} L0.5 & if\ arg\ max(Q_{CNN}(s_t)) = 0 \\ F0.2 & if\ arg\ max(Q_{CNN}(s_t)) = 1 \\ R0.5 & if\ arg\ max(Q_{CNN}(s_t)) = 2 \end{cases} \quad (4)$$

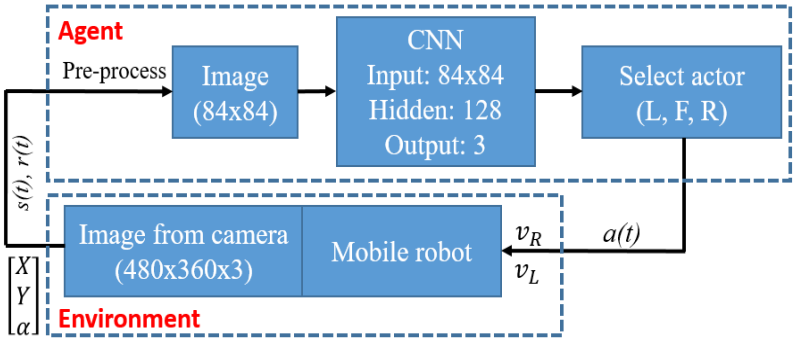


Figure 4. DQN Architecture for mobile robot

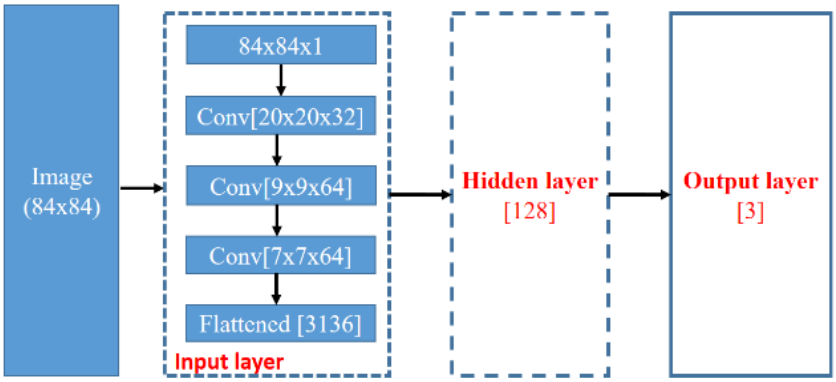


Figure 5. Convolution neural network structure

To define the reward for DQN, an error of robot situations is computed by the absolute position of setpoint and feedback based on the X axis. From the image of the camera, two points (feedback point and set point) could be determined as in Figure 6. Herein, Figures 6(a) and 6(b) are samples presented to the corner with the right side and left side, respectively. Figure 6(c) presents a straight situation without any corners. The feedback point is defined as the center of the camera view with the middle cross of width=480 and height=360, and the set point is the center of the rectangle covering the line trajectory. The robot is following the line when $X_{feedback}$=$X_{setpoint}$, the error could be defined as (5).

$$error = |X_{setpoint} - X_{feedback}| \quad (5)$$

The error is presented by the distance in pixels. Then, rewards will be defined based on the error value as shown in Table 2, which is divided into five levels -2, -1, -0.5, 0.5, and 1. These rewards are counted based on previous actions in (4) and the current state (error evaluations). Specifically, the error is less than equal to 80 pixels and the mobile robot is in the state forward (straight), this action (straight) is encouraged to maintain. Therefore, the reward is a positive number added to 1. Otherwise, the MB is in left/right control, and these actions are not encouraged in the next action. The reward is set to a negative number of -0.5. Similarly, if the error is in the range of 80 pixels to 120 pixels and all actions in the previous configured. This case is generally encouraged, so the reward is the normal value added to 0.5. Finally, the mobile robot should maintain the smallest error, and to avoid the worst case which sets the reward to *-2* if the *error* > 220. At the initial time, the DQN is empty, and it needs the training process for studying and updating the weight to become more intelligent. Next part, the training process will be considered in detail.

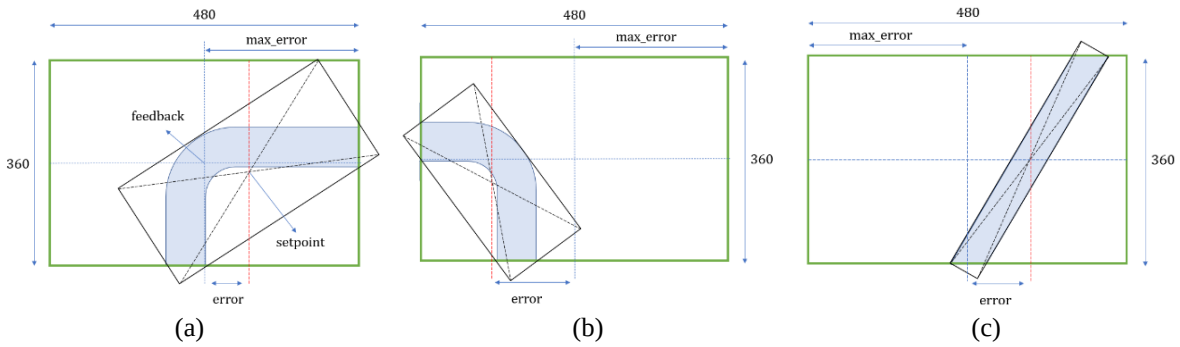


Figure 6. Computing error method: (a) view of corner with the right side, (b) view of corner with the left side, and (c) view of no corner with the straight situation

Table 2. Reward definitions

Actions	Conditions Errors (pixels)	Reward value	Descriptions
F0.2	$error \leq 80$	1	Encourage go straight
L0.5, R0.5		-0.5	Not encourage
Any actions	$80 < error \leq 120$	0.5	General encourage
F0.2	$120 < error \leq 220$	-1	Not encourage
L0.5, R0.5		0.5	General encouragement (turn left/right)
Any actions	$error > 220$	-2	Worst case

The training process of the DQN algorithm is shown in Figure 7. Herein, the predict and target model (PM, TM) are presented as the CNN type, which is used to predict the action values and calculate the value for the next state in Bellman (7), respectively. Initially, the agent receives the parameters, $s_{t,}$ and r_t, from the environment to choose the action based on (7) where $Q(s_t)$ is the output of the PM network. After that, the environment will export the new r_{t+1}, $s_{t+1,}$ and continue a new loop. A buffer memory M_t is used to store all data $M_t = [s_t, a_t, r_{t+1}, s_{t+1}]$ for the training of the CNN model. When M_t size comes to a maximum value definition, the lost function is executed and is used to update the weight of PM After that, the M_t is refreshed for new step storing. This step ensures the efficiency of the training process with the enough large dataset. The size of M_t could be defined as too large so it could affect to compute process and take longer time. Therefore, a batch_size is used to extract the number of random data from M_t for the training process.

$$MSE = \frac{1}{2}\left[Q_{eq}(s_t, a_t) - Q_{PM}(s_t, a_t)\right]^2 \text{ where } Q_{eq}(s_t, a_t) = r_t + \gamma \max\left(Q_{TM}(s_{t+1})\right) \quad (6)$$

The weight updating of PM based on the loss function could be computed by the mean squared error following in (6). Where $Q_{TM}(s_{t+1})$ is the output values of the TM network at s_{t+1} and $Q_{PM}(s_t, a_t)$ is the output of PM network at a_t and s_t. The PM weights are updated during n iteration and they are copied to update to the weights of TM. Based on MSE, the training process uses the Adam optimization with a learning rate selection [29].

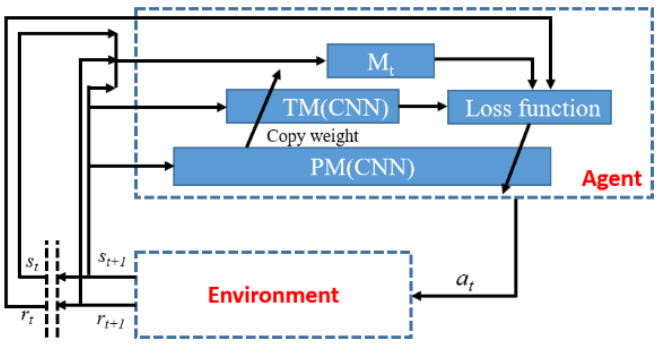


Figure 7. Training process for CNN model in agent

3. RESULTS AND DISCUSSION

In this paper, we use a Gazebo simulation environment [30] to simulate the four-wheel mobile robot tracking a line using the vision (camera plugin in Gazebo) as shown in Figure 8, and Python programming. The parameters of the mobile robot [31] and DQN algorithm [32] are presented in Table 3. For training processing, the number of trials is selected 40.000 times called episode_step by experimental. Each episode_step will have a start time (mobile robot go) and reset (until worse case) defined as *n_step* time. As a result in Figure 9(a), the FWMR moves with a short time (small *n_step*, small reward) and a short distance in the first episode_step because all parameters and weight are initial. They need more time for training and updating and then the n_step and reward could be better in episode_step number 57 as shown in Figure 9(b), which also presents the average reward value.

After the training process, the testing process to performed on a mobile robot tracking the line which builds in Gazebo simulation. The result in Figure 10 shows the performances of FWMR, in which the errors are computed with no absolute in (9) and shown in Figures 10(a) and 10(b). The MR tracking performance is shown in Figure 10(c). Overall, the error is in the range of -200px to 200px, and most of the focus is on -100 to 100px. This means that the error defined the deviation between the vision view (robot view) and the line, which is small (100 px is around 2.6 cm) and suitable for mapping in each direction, such as forward, left, and right. The error is less than 240, and that mobile robot is in the handle, not in the worst case which has the reward set by -2. With the case errors to around ±220 px, the mobile robot goes around the corner in the map. But it may not often because it depends on the previous direction and state of MR. The control tendency is to the error reduced to zero. So the errors exist with ±220px only maintained during a short time. Besides, mean squared error (MSE) is computed and compared between the proposal method and the traditional vision method [33] (using image moment) as shown in Table 4. Based on the error control in pixels, the convert to the meter is used and computed MSE (1 meter = 3779.52 pixels). The MSE in the DRL method is lower in VSN. It proves that the DRL method has higher performance than the VSN method.

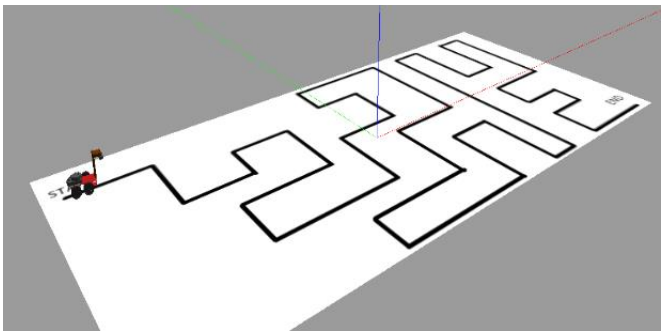
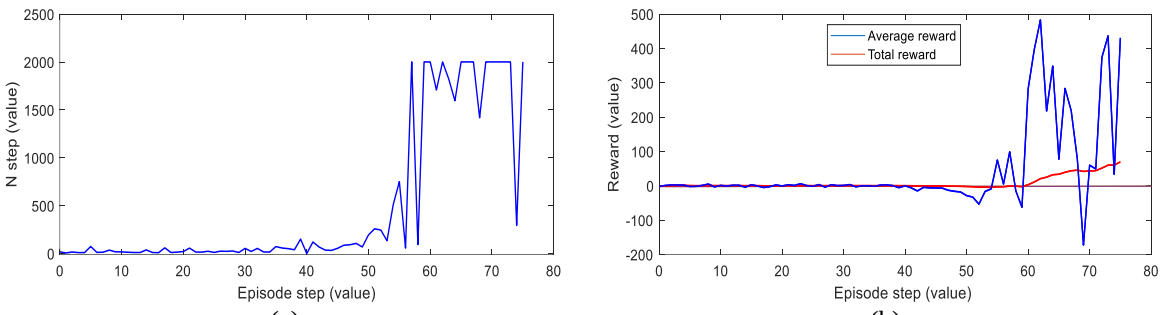


Figure 8. Four-wheel mobile robot follows line using camera scenario

Table 3. Parameter of mobile robot and DQN

Name	γ	ε	Size_camera	r	c	Batch_size	Episode_step	Learning rate
Value	0.99	1.0	480×360×3	*2 (cm)*	*5 (cm)*	32	40000	0.0001



(a) (b)

Figure 9. Training process results (a) n_step in one episodes_step and (b) reward and average reward in one episodes_step

Table 4. MSE comparison between proposal method and traditional vision method

Method	MSE * 100 (in meter)
Proposal method (DRL)	0.0264
Traditional vision method (VSN)	0.0406

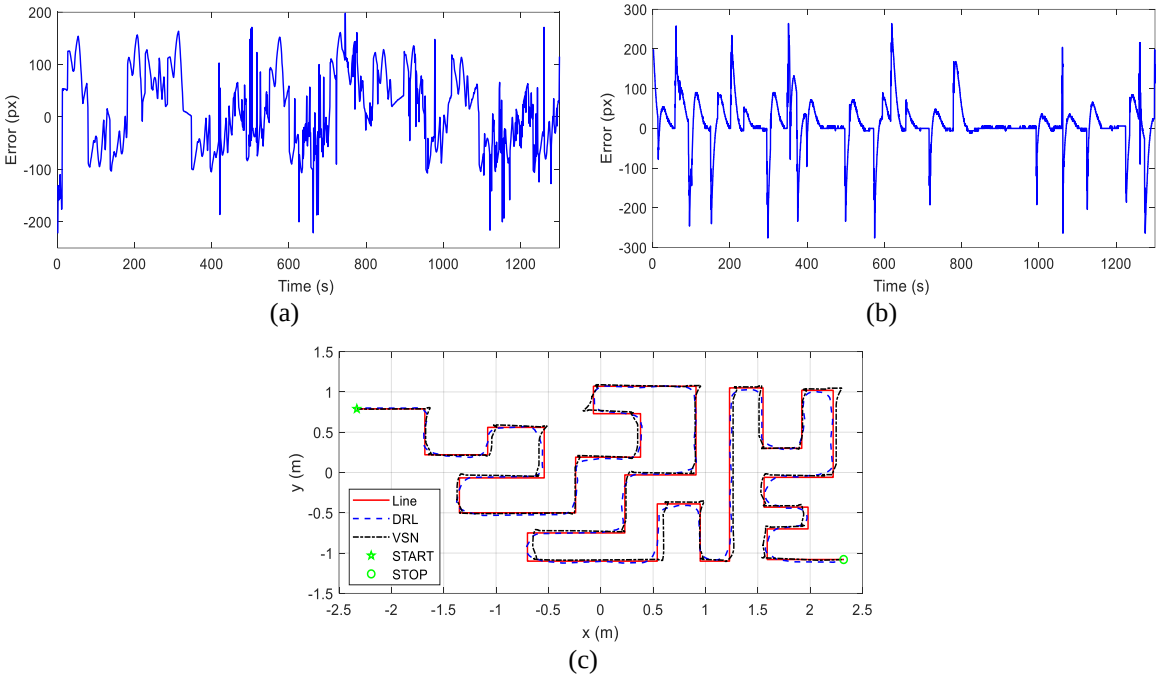


Figure 10. Performance of FWMR (a) error of point in image viewing from the robot in the proposal method, (b) error of point in image viewing from the robot in the traditional vision method, and (c) tracking line performance of FWMR

The performance of mobile robot tracking is shown in the details of Figure 10(c) that FWMR could complete a trajectory from the start point to the stop point in the map. Easily seems that the MR has a bigger error in each corner. But it is still better than the VSN method. However, in the straight line, MR could track the trajectory with higher performance (overlap line, closed distance with line). Thus, the MR uses the CNN based on the image from the camera can generate the action for each state given by the environment in the DRL structure. The target of indoor navigation of MR is established success in complex mapping and maintaining it in a real-time system. This proves our proposal is useful and brings high effectiveness for the navigation and tracking line of a mobile robot.

4. CONCLUSION

This paper presents the DQN in DRL for the navigation of a mobile robot following the line on the ground based on a convolution neural network for processing images from a camera. All steps are discussed in detail to clear performance and apply to the specific object. The results improved the effectiveness of the control algorithm so that the mobile robot could follow the line with high accuracy presented by the error of less than 100 pixels with different types of trajectories. This proposal could be applied to mobile robot navigation as well as in robotics fields in the future.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Khoa Nguyen Dang	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Van Tran Thi	✓	✓			✓			✓	✓	✓	✓	✓		
Nguyen Van Thang			✓	✓			✓			✓	✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, KND, upon reasonable request.

REFERENCES

[1] A. Markis, M. Papa, D. Kaselautzke, M. Rathmair, V. Sattinger, and M. Brandstötter, "Safety of mobile robot systems in industrial applications," *ARW & OAGM Workshop 2019*, pp. 26–31, 2019.

[2] A. Loganathan and N. S. Ahmad, "A systematic review on recent advances in autonomous mobile robot navigation," *Engineering Science and Technology, an International Journal*, vol. 40, p. 101343, Apr. 2023, doi: 10.1016/j.jestch.2023.101343.

[3] C. H. A. H. B. Baskoro, H. M. Saputra, M. Mirdanies, V. Susanti, M. F. Radzi, and R. I. A. Aziz, "An autonomous mobile robot platform for medical purpose," in *Proceeding - 2020 International Conference on Sustainable Energy Engineering and Application: Sustainable Energy and Transportation: Towards All-Renewable Future, ICSEEA 2020*, Nov. 2020, pp. 41–44. doi: 10.1109/ICSEEA50711.2020.9306161.

[4] Z. Gong, Z. Nie, Q. Liu, and X. J. Liu, "Design and control of a multi-mobile-robot cooperative transport system based on a novel six degree-of-freedom connector," *ISA Transactions*, vol. 139, pp. 606–620, Aug. 2023, doi: 10.1016/j.isatra.2023.04.006.

[5] B. S. O. Pallares, T. A. M. Rozo, E. C. Camacho, J. G. Guarnizo, and J. M. Calderon, "Design and construction of a cost-oriented mobile robot for domestic assistance," *IFAC-PapersOnLine*, vol. 54, no. 13, pp. 293–298, 2021, doi: 10.1016/j.ifacol.2021.10.462.

[6] S. Habibian *et al.*, "Design and implementation of a maxi-sized mobile robot (Karo) for rescue missions," *ROBOMECH Journal*, vol. 8, no. 1, Jan. 2021, doi: 10.1186/s40648-020-00188-9.

[7] H. Azpúrua *et al.*, "A survey on the autonomous exploration of confined subterranean spaces: perspectives from real-world and industrial robotic deployments," *Robotics and Autonomous Systems*, vol. 160, p. 104304, Feb. 2023, doi: 10.1016/j.robot.2022.104304.

[8] V. Bulut, "Path planning of mobile robots in dynamic environment based on analytic geometry and cubic Bézier curve with three shape parameters," *Expert Systems with Applications*, vol. 233, p. 120942, Dec. 2023, doi: 10.1016/j.eswa.2023.120942.

[9] P. S. Yadav, V. Agrawal, J. C. Mohanta, and M. D. Faiyaz Ahmed, "A robust sliding mode control of mecanum wheel-chair for trajectory tracking," *Materials Today: Proceedings*, vol. 56, pp. 623–630, 2022, doi: 10.1016/j.matpr.2021.12.398.

[10] D. U. Rijalusalam and I. Iswanto, "Implementation kinematics modeling and odometry of four omni wheel mobile robot on the trajectory planning and motion control based microcontroller," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 448–455, 2021, doi: 10.18196/jrc.25121.

[11] Z. Han, J. Long, W. Wang, and L. Wang, "Adaptive tracking control of two-wheeled mobile robots under Denial-of-Service attacks," *ISA Transactions*, vol. 141, pp. 365–376, Oct. 2023, doi: 10.1016/j.isatra.2023.06.022.

[12] J. S. Ling Leong, K. T. Kin Teo, and H. P. Yoong, "Four wheeled mobile robots: a review," in *4th IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAIET 2022*, Sep. 2022, pp. 1–6. doi: 10.1109/IICAIET55139.2022.9936855.

[13] M. H. Mohamad Nor, Z. H. Ismail, and M. A. Ahmad, "Broadcast control of multi-robot systems with norm-limited update vector," *International Journal of Advanced Robotic Systems*, vol. 17, no. 4, Jul. 2020, doi: 10.1177/1729881420945958.

[14] A. Stefek, T. van Pham, V. Krivanek, and K. L. Pham, "Energy comparison of controllers used for a differential drive wheeled mobile robot," *IEEE Access*, vol. 8, pp. 170915–170927, 2020, doi: 10.1109/ACCESS.2020.3023345.

[15] X. Luo, D. Mu, Z. Wang, P. Ning, and C. Hua, "Adaptive full-state constrained tracking control for mobile robotic system with unknown dead-zone input," *Neurocomputing*, vol. 524, pp. 31–42, Mar. 2023, doi: 10.1016/j.neucom.2022.12.025.

[16] J. Li, J. Wang, H. Peng, Y. Hu, and H. Su, "Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2491–2500, Apr. 2022, doi: 10.1109/TSMC.2021.3050616.

[17] Y. Liu, S. Wang, Y. Xie, T. Xiong, and M. Wu, "A review of sensing technologies for indoor autonomous mobile robots," *Sensors*, vol. 24, no. 4, 2024, doi: 10.3390/s24041222.

[18] O. Kedziora *et al.*, "Active speed and cruise control of the mobile robot based on the analysis of the position of the preceding vehicle," in *2022 26th International Conference on Methods and Models in Automation and Robotics, MMAR 2022 - Proceedings*, Aug. 2022, pp. 181–186. doi: 10.1109/MMAR55195.2022.9874298.

[19] R. Farkh, K. Al Jaloud, S. Alhuwaimel, M. T. Quasim, and M. Ksouri, "A deep learning approach for the mobile-robot motion control system," *Intelligent Automation and Soft Computing*, vol. 29, no. 2, pp. 423–435, 2021, doi: 10.32604/iasc.2021.016219.

[20] D. Trujillo, L. A. Morales, D. Chávez, and D. F. Pozo, "Trajectory tracking control of a mobile robot using neural networks," *Emerging Science Journal*, vol. 7, no. 6, pp. 1843–1862, Dec. 2023, doi: 10.28991/ESJ-2023-07-06-01.

[21] H. Li, Q. Zhang, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2064–2076, Jun. 2020, doi: 10.1109/TNNLS.2019.2927869.
[22] M. F. R. Lee and S. H. Yusuf, "Mobile robot navigation using deep reinforcement learning," *Processes*, vol. 10, no. 12, p. 2748, Dec. 2022, doi: 10.3390/pr10122748.
[23] J. Zhao, M. Kong, Q. Li, and X. Sun, "Contract-based computing resource management via deep reinforcement learning in vehicular fog computing," *IEEE Access*, vol. 8, pp. 3319–3329, 2020, doi: 10.1109/ACCESS.2019.2963051.
[24] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, Oct. 2021, doi: 10.26599/TST.2021.9010012.
[25] R. Patel and S. Patel, "A comprehensive study of applying convolutional neural network for computer vision," *International Journal of Advanced Science and Technology*, vol. 29, no. 6 Special Issue, pp. 2161–2174, 2020.
[26] J. Platt and K. Ricks, "Comparative analysis of ROS-Unity3D and ROS-Gazebo for mobile ground robot simulation," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 106, no. 4, Dec. 2022, doi: 10.1007/s10846-022-01766-2.
[27] J. Yu, Y. Su, and Y. Liao, "The path planning of mobile robot by neural networks and hierarchical reinforcement learning," *Frontiers in Neurorobotics*, vol. 14, Oct. 2020, doi: 10.3389/fnbot.2020.00063.
[28] F. Foroughi, Z. Chen, and J. Wang, "A CNN-based system for mobile robot navigation in indoor environments via visual localization with a small dataset," *World Electric Vehicle Journal*, vol. 12, no. 3, p. 134, Aug. 2021, doi: 10.3390/wevj12030134.
[29] J. L. Ba and D. P. Kingma, "Adam: a method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.
[30] A. Farley, J. Wang, and J. A. Marshall, "How to pick a mobile robot simulator: a quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion," *Simulation Modelling Practice and Theory*, vol. 120, p. 102629, Nov. 2022, doi: 10.1016/j.simpat.2022.102629.
[31] W. Jo, J. Kim, R. Wang, J. Pan, R. K. Senthilkumaran, and B.-C. Min, "SMARTmBOT: a ROS2-based low-cost and open-source mobile robot platform," *arXiv:2203.08903*, 2022.
[32] M. K. Chegeni, A. Rashno, and S. Fadaei, "Convolution-layer parameters optimization in convolutional neural networks," *Knowledge-Based Systems*, vol. 261, p. 110210, Feb. 2023, doi: 10.1016/j.knosys.2022.110210.
[33] G. Yao, R. Saltus, and A. Dani, "Image moment-based extended object tracking for complex motions," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6560–6572, Jun. 2020, doi: 10.1109/JSEN.2020.2976540.

BIOGRAPHIES OF AUTHORS

Khoa Nguyen Dang received an M.Sc. degree in information technology from Hanoi University of Science and Technology, Vietnam, in 2012 and a Ph.D. degree in aerospace engineering from Ulsan University, Korea. He is a lecturer at the Faculty of Engineering and Technology, International School, Vietnam National University. His research interests include UAV, UGV, control systems, intelligent control, IoT, and artificial intelligence. He can be contacted at email: khoand@vnuis.edu.vn.

Van Tran Thi received an MSc. degree in information technology from Hanoi University of Science and Technology, Vietnam in 2012. Currently, she is a lecturer at the University of Labour and Social Affairs. Her research interests include UGV, control systems, IoT, and artificial intelligence. She can be contacted at email: tranvantk4@gmail.com.

Nguyen Van Thang received an M.Sc. degree in electrical engineering from Le Quy Don University, Vietnam in 2007, and his Ph.D. degree in the same major from the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam. He is currently a lecturer in the Faculty of Electronics and Telecommunications, UET. His interesting include IoT, DSP/sensor applications, and medical equipment. He can be contacted at email: nvthangdtvt@vnu.edu.vn.