symmetry

MDPI

Article

Improved Path Planning for Indoor Patrol Robot Based on Deep Reinforcement Learning

Jianfeng Zheng, Shuren Mao, Zhenyu Wu, Pengcheng Kong and Hao Qiang *

School of Mechanical Engineering and Rail Transit, Changzhou University, Changzhou 213164, China; zjf@cczu.edu.cn (J.Z.); 19085206180@smail.cczu.edu.cn (S.M.); 19085206557@smail.cczu.edu.cn (Z.W.); 19085206920@smail.cczu.edu.cn (P.K.)

* Correspondence: qhao@cczu.edu.cn

Abstract: To solve the problems of poor exploration ability and convergence speed of traditional deep reinforcement learning in the navigation task of the patrol robot under indoor specified routes, an improved deep reinforcement learning algorithm based on Pan/Tilt/Zoom(PTZ) image information was proposed in this paper. The obtained symmetric image information and target position information are taken as the input of the network, the speed of the robot is taken as the output of the next action, and the circular route with boundary is taken as the test. The improved reward and punishment function is designed to improve the convergence speed of the algorithm and optimize the path so that the robot can plan a safer path while avoiding obstacles first. Compared with Deep Q Network(DQN) algorithm, the convergence speed after improvement is shortened by about 40%, and the loss function is more stable.

Keywords: patrol robot; path planning; autonomous navigation; DQN; rewards and punishments function

Citation: Zheng, J.; Mao, S.; Wu, Z.; Kong, P.; Qiang, H. Improved Path Planning for Indoor Patrol Robot Based on Deep Reinforcement Learning. *Symmetry* **2022**, *14*, 132. https://doi.org/10.3390/sym14010132

Academic Editors: Chengxi Zhang, Jin Wu and Chong Li

Received: 10 December 2021
Accepted: 7 January 2022
Published: 11 January 2022

1. Introduction

Path planning is an essential direction of robot research [1]. It is the key to realizing autonomous navigation tasks, which means a robot can independently explore a smooth and collision-free path trajectory from the starting position to the target position [2]. Traditional path planning algorithms include the A-star algorithm [3], Artificial Potential Field Method [4], and Rapidly Exploring Random Tree [5], and so on, which are used to solve the path planning in a known environment and are easy to implement. Still, robots have poor exploration ability in path planning. Given, because of the problems in traditional algorithms, a deep reinforcement learning algorithm has been introduced to enable robots to make more accurate movement directions in environmental states, which is a combination of deep learning and reinforcement learning [6–8]. Deep learning obtains the observation information of the target state by perceiving the environment. In contrast, reinforcement learning, which uses the reward and punishment functions to guide whether the action is good or not, is a process in which the patrol robot and the environment interact trial and error constantly.

The first deep reinforcement learning model, namely DQN, was proposed by Mnih et al. [9], which combined neural network with Q-learning and used a neural network to replace the Q value table to solve the problem of dimension disaster in Q-learning. Still, the convergence speed was slow in network training. DQN was applied to path planning for model-free obstacle avoidance by Tai et al. [10], which was the over-estimation of state-action value, inevitably resulting in sparse rewards and not the optimal path for robots. The artificial potential field, which accelerated the convergence speed of the network, increased the action step length and adjusted the direction of the robot to improve the precision of the robot's route planning, was introduced in the initialization process of Q value [11], leading to having good effect in the local path planning of the robot but poor implementation in the global path planning.

Because of the fixed and symmetrical patrol route, it is difficult for the patrol robot to avoid obstacles. The A-star algorithm is characterized by large performance consumption when encountering target points with obstacles. Artificial Potential Field Method cannot find a path between similar obstacles. The path planned by the Artificial Potential Field Method may oscillate and swing in narrow channels when new environmental obstacles are detected. Rapidly Exploring Random Tree is challenging to find the path in an environment with limited channels. Therefore, to effectively solve the patrol robot exploration algorithm problems of slow convergence speed and poor ability, an improved path planning for indoor patrol robot based on deep reinforcement learning is put forward in this paper, which use PTZ to perceive the surrounding environment information, combining the patrol robot's position information with the target of a state-space as network input [12–15]. The position and speed of the patrol robot are taken as the output, and a reasonable reward and punishment function is designed to significantly improve the convergence speed of the algorithm and optimize the reward sparsity of the environmental state space in the paper [16–19].

2. Background

Reinforcement learning is a process in which different rewards are obtained by way of "trial and error" when the patrol robot and the environment interact, which means that the patrol robot will not be affected by the initial stage of the environment, which does not guide the patrol robot to move but only rate its interaction [20–22]. High score behavior and low score behavior need to be remembered by the patrol robot, which just needs to use the same behavior to get a high score and avoid a low score when it interacts next time [23–26]. The interaction process of reinforcement learning is shown in Figure 1. DQN, which uses a neural network for fusion, is based on Q-learning for overcoming the defect of "dimension disaster" caused by large memory consumption of Q-learning to store data [27–29].

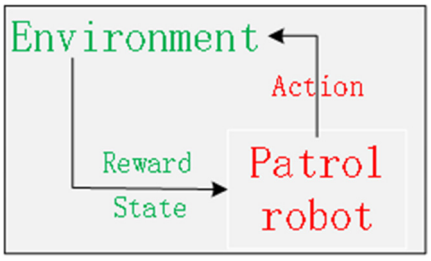


Figure 1. Reinforcement learning flow chart.

Deep reinforcement learning mainly realizes the learning interaction between the patrol robot and the environment, which is composed of deep learning and reinforcement learning. Deep learning uses the patrol robot's built-in sensors to receive and perceive the information of the surrounding environment and obtain the information of the current state of the patrol robot. While reinforcement learning is responsible for the patrol robot to explore and analyze the acquired environmental information, which helps the patrol robot to make correct decisions and makes the patrol robot can achieve navigation tasks come true [30–32].

The DQN algorithm combines a neural network, which needs to model Q table and uses RGB image as input to generate all Q values and Q-learning which uses Markov decision for modeling and uses the current state, action, reward, strategy, and next action in Markov decision for representation. The experience playback mechanism is introduced to improve the sample correlation of the robot and solve the efficiency utilization problem of the robot in DQN, which uses the uniqueness of the target Q value to enhance the smoothness of the action update. DQN includes three steps: establishing target function, target network, and introducing experience replay.

Target function. The target function of DQN is constructed by Q-learning, and the formula is as follows:

$$Q'(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma max_{a'}Q(s',a') - Q(s,a)\right] \quad (1)$$

where (s,a) indicates the current status and action. (s′,a′) indicates the next state and action. Q(s,a) represents the current state-action value. Q′(s,a) represents the updated state-action value. α represents the learning rate which means how many errors in the current state will be updated and learned. The value ranges from 0 to 1. γ represents the attenuation value of future rewards, ranging from 0 to 1. Because the Q value in the DQN algorithm is mostly randomly generated. For the convenience of calculation, we need to use the maximum value max.$max_{a'}Q(s',a')$ means the maximum value of Q at the next state-action value. Breaking down the expression for Q values yields the following expression:

$$(s1) = r2 + \gamma Q(s2) = r2 + \gamma[r3 + \gamma Q(s3)] = r2 + \gamma\{r3 + \gamma[r4 + \gamma Q(s4)]\} = \cdots \quad (2)$$

$$\text{Namely } Q(s1) = r2 + \gamma \cdot r3 + \gamma^2 \cdot r4 + \gamma^3 \cdot r5 + \gamma^4 \cdot r6 + \cdots \quad (3)$$

It is not difficult to conclude that the value of Q is correlated with the rewards at each subsequent step, but these associated rewards decay over time, and the further away from state s1, the more state decay.

The target state-action value function can be expressed by the Bellman equation as follows:

$$y' = r + \gamma maxQ(s',a',\theta) \quad (4)$$

where y′ is the target Q value. θ is the weight parameter trained in the neural network structure model. $maxQ(s',a',\theta)$ means the maximum value of Q ant the next state-action value and θ.

The loss function is the mean square error loss function, and the formula is as follows:

$$L(\theta) = E\left[\left(y' - Q(s,a,\theta)\right)^2\right] \quad (5)$$

Target network. The current state-action value function is evaluated by DQN through the target network and prediction network. The target network is based on a neural network to get the target Q value, using the target Q value to estimate the Q value of the next action. The prediction network uses the stochastic gradient descent method to update the weight of the network $\Delta\theta$, and the formula of the gradient descent algorithm is shown as follows:

$$\Delta\theta = E\left[y' - Q(s,a,\theta_1)\nabla_\theta Q(s,a,\theta_1)\right] \quad (6)$$

where ∇ is Hamiltonian.

Experience replay. The experiential replay mechanism improves the sample relevance of the patrol robot and solves the problem of efficient utilization of the patrol robot. The patrol robot can obtain the sample database during the information interaction between the patrol robot and the environment. It stores the sample database into the established experience pool and randomly selects a small part of data for training samples, and then sends the training samples into the neural network for training. The experience replay mechanism utilizes the repeatability of the sample itself to improve learning efficiency.

The structure of the DQN algorithm is shown in Figure 2. s (state) represents the current cycle step, r (reward) represents the reward value generated by the current cycle step, a(action) represents the behavior caused according to the current cycle step state, and s_(next State) represents the next cycle step. Target_net and Eval_net refer to Q_target and Q_eval, respectively. Parameters of Eval_net are deferred to parameters in Target_net. Q_target and Q_eval represent the two neural networks of the DQN algorithm, which follows Q-learning, the predecessor of the DQN algorithm. They are not much different, just different parameters. Q_eval contains behavioral parameters, and theoretical behavior

is bound to deviate from actual conduct (loss). Thus, in the traditional DQN algorithm, parameters s, loss, and symmetric Q_target and Q_eval will affect the Train value.

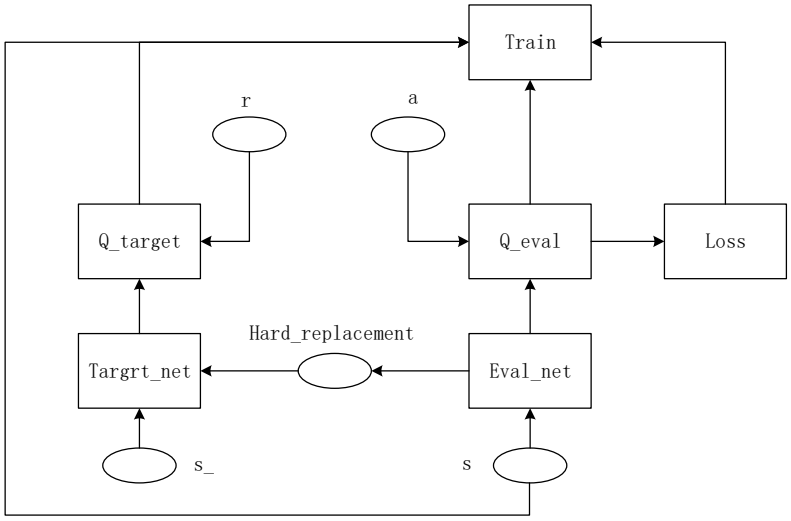


Figure 2. Structure diagram of DQN algorithm.

3. Improved Algorithm

The main goal of the patrol robot is to be able to reach the target point and return to the starting point autonomously in the indoor route environment. So a kind of improved deep reinforcement learning algorithm is put forward in this paper, which takes the acquired image information and target position information as the input of the network, the position and speed of the patrol robot as the output of the next action, and the specified circular route with a boundary as the test. Thus, the patrol robot can realize the process of its running towards the target point in the specified route with limited conditions and finally returning to the starting point to complete the automatic walking task. Since the robot needs to walk on the indoor prescribed route to realize the patrol function, the conventional DQN algorithm is easy to find the optimal path for the open route. Still, for the circular prescribed route, it is easy to fall into the local optimal and cannot complete the path. Therefore, the DQN algorithm needs to be improved.

3.1. Overview of Improved DQN Algorithm

The structure of the improved DQN algorithm is shown in Figure 3, which is transmitted to the evaluation function through the current iteration step s. The evaluation function generates four different data types and sends the data to the target function through four symmetric routes. In addition to the four transmission lines of the objective function, the next iteration step s_ also affects them. The evaluation function and Q function often have a loss value during operation, which, together with the evaluation function, Q function, and the current iterative step s, guides the patrol robot training.

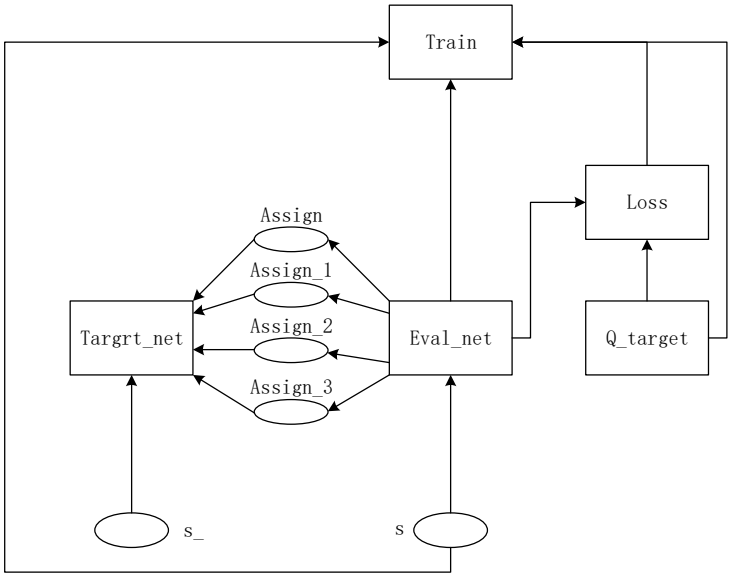


Figure 3. Structure diagram of improved DQN algorithm.

In the simulation environment, the patrol robot takes the directly collected image information as the training sample, then combines its environmental state characteristics and the target point to be reached as the input of the network, and takes the Q value of the current position as the output of the network model, and uses the ε-greedy strategy to select the action to reach the next state. When the next state is reached, a complete data tuple (s,a,r,s′) can be obtained by calculating the corresponding reward value r. Then, data of this series are stored in the experience replay pool D, and small-batch samples are extracted from the experience replay pool D and put into the neural network for training.

In the course of network training, if the patrol robot uses PTZ to identify the obstacle, the improved algorithm can make the robot avoid the block effectively. Otherwise, the patrol robot will continue to navigate until the target point is reached. The improved algorithm design is shown in Algorithm 1.

Algorithm 1. Improved Deep Q-learning Network Based on Patrol Robot.

Initialize replay memory $\boldsymbol{D}$ to capacity $\boldsymbol{N}$
Initialize action-value function $\boldsymbol{Q}$ with random weights
for episode = 1, $\boldsymbol{M}$ do
Initialize sequence s_t from replay memory $\boldsymbol{D}$ and preprocess sequenced $\boldsymbol{\theta}$, $\boldsymbol{\theta} = \boldsymbol{\theta}_1$
for t = 1, $\boldsymbol{T}$ do
With probability ε select a random action a_t
Otherwise select optimal action $a_t = \max_a Q'((s_t), a; \theta)$
Execute action a_t in emulator, observe new reward function r_t and next image s_{t+1}
Set transition (s_t, a_t, r_t, s_{t+1}) and store it in $\boldsymbol{D}$
Preprocess $\theta_{t+1} = \theta_{(s_{t+1})}$
Sample random mini-batch of transitions $\left(s_j, a_j, r_j, s_{j+1}\right)$ from $\boldsymbol{D}$
Set $y_j(a|s) = \begin{cases} r_j & for\ terminal_{j+1} \\ r_j + \gamma max_{a'} Q\left(s_{j+1}, a'; \theta\right) & for\ non-terminal_{j+1} \end{cases}$
Perform a gradient descent step on $L(\theta)$

$$L(\theta) = E\left(y_j - Q(s, a, \theta)\right)^2$$

Update policy gradient
end for
end for

3.2. Improved Target Point Function

The target point is the coordinate of the position that the patrol robot needs to achieve in the motion state and represents the final position of the patrol robot. It needs to drive on a fixed route for the patrol robot when performing tasks and return to the initial position after completing the patrol. If the target point is placed in the starting position, the DQN algorithm will skip to the end without iteration, thus skipping the patrol step. Even if the design function makes the algorithm stop iteration when it reaches the initial position for the second time, due to the characteristics of the DQN algorithm, it will also stop iteration quickly, and the final walking path may only be a small lattice.

In the circular path, the patrol robot has a long distance to walk as the start point of the path is just the endpoint. When encountering obstacles, the conventional DQN algorithm will begin from the starting point again, resulting in a long calculation cycle of the whole circle.

Therefore, the paper improves the target point function, which the circular route used in this paper is abstracted. In a grid of 30*30, the track boundary is set in black to represent obstacle points. The red point represents the starting point which is the initial position of the patrol robot. The yellow point represents the target point. The paper segments the whole ring line, the yellow spot under an initial state for the first target of the patrol robot. While the patrol robot achieves the first target point, the second target point will be set in the forward line of the patrol robot, at the same time, the first target point will be the new starting point, and turn the white grid point closest to the first target point on the previous route into an obstacle point to make it become the barriers for which the patrol robot can accelerate the efficiency of iteration in the subsequent iterations of the patrol robot, and it also prevents the patrol robot from going "backward" in each iteration. The target point of the last stage is set as the starting point of the initial step to ensure that the patrol robot can complete the whole loop route.

3.3. Improved Reward and Punishment Function

The calculation is performed only according to the improved target point function modified in Section 3.2. As the robot has a relatively large space for walking and the degree of freedom is greatly improved, the algorithm calculation time is too long, which violates the original intention of using the DQN algorithm. Therefore, this paper starts from the reward and punishment function r, and the reward and punishment function r of the design change is shown as follows.

$$r_{(i,j)} = \begin{cases} -1,\ (i,j)\ is\ an\ obstacle \\ 1,\ (i,j)\ is\ a\ target\ point \\ \mu,\ r_{(m,n)} \neq -1, m \in [i-1, i+1], n \in [j-1, j+1] \\ 0,\ others \end{cases} \tag{7}$$

In the original algorithm, the reward value of all points except the obstacle point and target point is 0. However, the reward and punishment function used in this paper adds a new reward and punishment value μ, which ranges from 0 to 1. The point to which this value is assigned must have the following characteristics: the nine points centered on this point, including the eight surrounding points, are not obstacle points. This point is better than other blank points, and the patrol robot should choose it first.

As shown in Figure 4, the blue point is the new punishment and reward point, but the green point is not because the point to the lower left of the green point is the obstacle point.

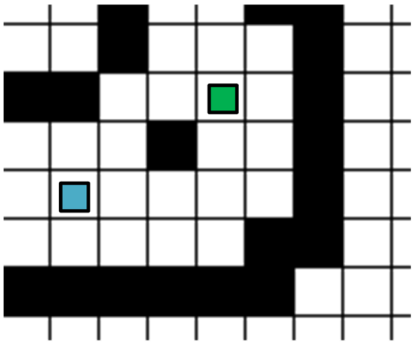


Figure 4. An illustration of the new reward and punishment value.

4. Experimental Analysis and Results

4.1. Experimental Environment and Parameter Configuration

To realize the running experiment of patrol robots and verify the validity of the algorithm in this paper under the specified route, the following experiments comparing with DQN are carried out in this paper. The experimental environment is composed of NVIDIA GeForce GTX 1660 SUPER GPU server, ROS operating system of the robot and Pycharm. The patrol robot training process is trained in the simulation environment built by Pycharm and then transplanted into the patrol robot named Raspblock with PTZ.

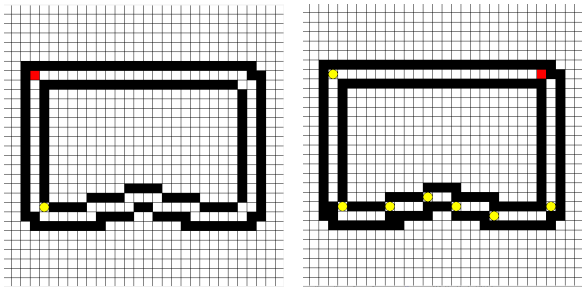
The patrol robot takes the state Q value as the input and the action Q value as the output, thus forming a state-action pair. If the patrol robot bumps into obstacles while running, it will get a negative reward. If the patrol robot reaches the target point, it will get a positive reward; the patrol robot also receives fewer positive rewards if it walks on a road point with no obstacles around it. The patrol robot can avoid obstacles in the learning process and keep approaching the target to complete the path planning process through the method of reward and punishment mechanism. Parameter Settings of the improved deep reinforcement learning algorithm are shown in Table 1.

Table 1. Parameter Settings.

Parameter	Value
Batch	32
Episode	10,000
Learning rate α	0.01
Reward decay γ	0.9
ε − greedy	0.9

4.2. Experimental Modeling Procedure

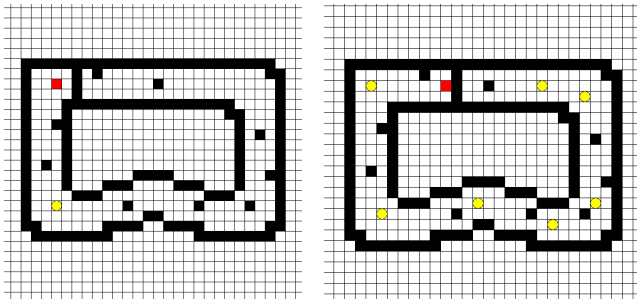
The paper is modeling as described in Section 3.2 and the characteristics of map symmetry are shown in Figure 5a. The red point represents the starting point of the robot, which can change as needed. The yellow point represents the target point, and the next target point will generate after the robot reaches it until it runs a full lap. The black points represent obstacles or track boundaries. The state of the final phase is shown in Figure 5b. The last target point is in the same position as the start of the first stage.



(**a**) (**b**)

Figure 5. Schematic diagram of the initial model. (**a**) The first stage; (**b**) The final stage.

The annular region described above conforms to the general condition. However, when obstacles suddenly appear in the specified path, the patrol robot must prioritize obstacle avoidance and then patrol according to the loop. In this paper, the corridor shape is retained, and the white area in the middle of the two black circles is enlarged to deal with obstacles in the line. The new model that preserves the map symmetry is shown in Figure 6. As the black block points in the corridor are randomly set in Figure 6a, the randomness of barriers and the rationality of the algorithm are ensured. The black spots representing obstacles in the model include the original track boundary and the newly added obstacle spots. The new reward and punishment value μ at the part of Section 3.3 is introduced. The state of the last stage under the new model is shown in Figure 6b.



(**a**) (**b**)

Figure 6. Schematic diagram of the new model. (**a**) The first stage; (**b**) The final stage.

4.3. Analysis of Experimental Results

In this paper, the DQN algorithm and the improved algorithm using training times and loss function values are analyzed, and the experimental results are compared. With the same parameters, the convergence speed of the improved DQN algorithm is about 30% faster than that of the DQN algorithm, and the average loss function value of the improved DQN algorithm is about 25% smaller than that of The DQN algorithm under the same training times. As shown in Figure 7, Figure 7a represents the change curve of training times-loss function obtained by the operation of the DQN algorithm, and Figure 7b represents the same by the operation of improved DQN algorithm with a value of 0.5 for the particular reward and punishment value μ. The number of training times is less than 500, which belongs to the initial training stage. The patrol robot is in the stage of exploration and learning and fails to make correct judgments on obstacles, resulting in a significant loss. When the number of training reaches 500 times, the patrol robot is still exploring the learning obstacle avoidance stage, indicating that it has begun to identify obstacles and correctly avoid some obstacles. However, due to the lack of training, it is still learning and interacting with the environment to adjust further actions to avoid more obstacles to reduce losses. The improved DQN algorithm in the current state is more stable. When the training times are between 500 and 1500, the patrol robot is unstable under the DQN algorithm and the improved DQN algorithm. When the training times reach about 1800, the loss function

of the improved DQN algorithm tends to be balanced. When the training times reach about 2600, the loss function of the DQN algorithm is regionally balanced. In the testing stage, the training result model is used to test the network in the same environment to verify its effectiveness further. The objective function and reward value function of test and training are consistent. Therefore, the improved algorithm can shorten the network training time and make the patrol robot plan a shorter path.

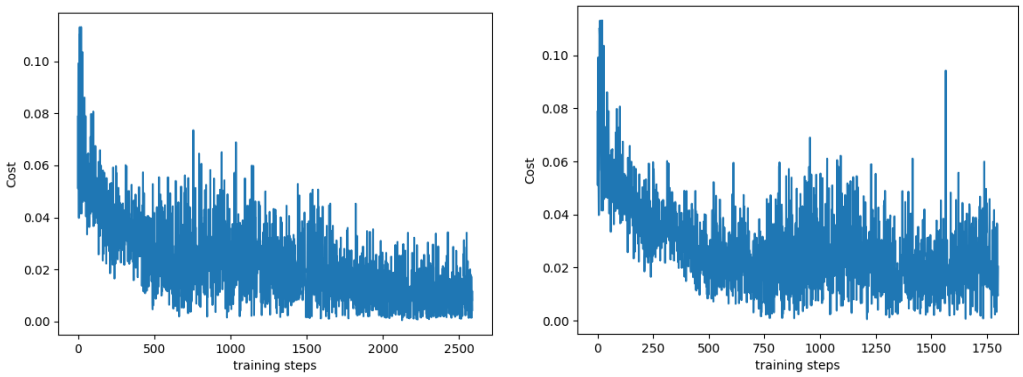
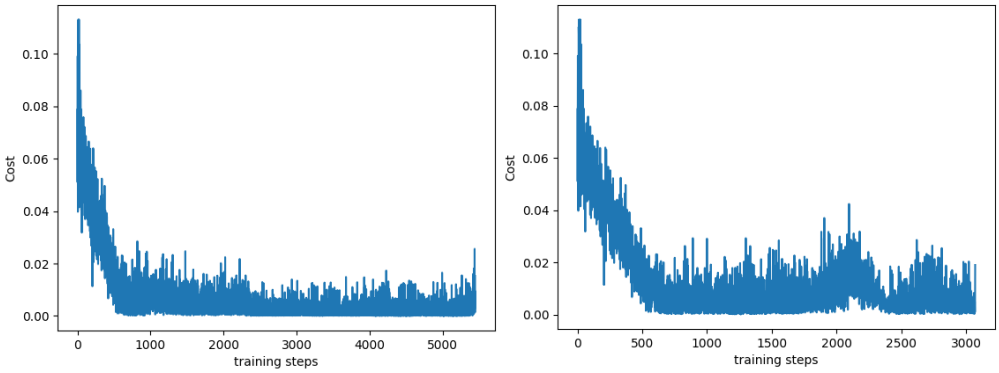


Figure 7. A comparison of the two DQN algorithms. (**a**) Conventional DQN algorithm; (**b**) Improved DQN algorithm ($\mu = 0.5$).

Although the improved objective function and reward value function reduces the convergence speed of the algorithm, the corresponding loss function slightly increases. The analysis results show that because the particular reward and punishment value of 0.5 set in the above experiment is too large, the accumulation speed of the reward and punishment value of the patrol robot is too fast, which makes the patrol robot have serious interference in the judgment of the later stage. Therefore, the following modifications are made in this paper, which is the special reward and punishment value 0.5 is changed to 10^{-6}, and it only affects the initial learning, adjusted to 0 after training. The training-loss function generated by the modified algorithm is shown in Figure 8. Figure 8a represents the change curve of the training-loss function obtained when the value of μ is 10^{-6} forever. Figure 8b illustrates the corresponding change curve obtained when the value of μ is 10^{-6} at the initial stage of the experiment and 0 at the later stage of the investigation. Compared with the improved algorithm before, the training times of the two algorithms are increased, which is caused by the sharp decrease of the particular reward and punishment value μ. However, by comparison, the convergence speed of the optimized algorithm with changing μ value is about 40% less than that of the algorithm with unchanged μ value. The number of training times is less than 500, which belongs to the initial training stage. The patrol robot is in the stage of exploration and learning and fails to make a correct judgment on obstacles. The loss value is still large, but both algorithms have a downward trend, which the loss function of the algorithm with μ value changing declines earlier. When the number of training reaches 500 times, both algorithms tend to balance, and the algorithm with changed μ value has fewer training times, which means that the improved DQN algorithm with changed μ value can complete the training faster, and the patrol robot can avoid obstacles and reach the target point more quickly. The feasibility of the improved algorithm is further verified.



(**a**) (**b**)

Figure 8. Influence of μ value on improved DQN algorithm. (**a**) $\mu = 10^{-6}$; (**b**) Changeable μ.

After several rounds of learning according to the improved algorithm, the patrol robot patrols according to the route shown in Figure 9. If the improved algorithm is not used, the patrol robot is prone to an infinite loop at the yellow target point in the lower-left corner of Figure 9, thus unable to complete the walking task. Because this algorithm is cyclic, no matter which small segment of the target point to take a screenshot will contain part of the obstacle point. Figure 9 is the final part of the target point of the algorithm in the whole circle. The corresponding obstacle point of the last part will be displayed, and other screenshots will also contain the related obstacle point. To distinguish, the transformation of the obstacle points is especially changed to blue.

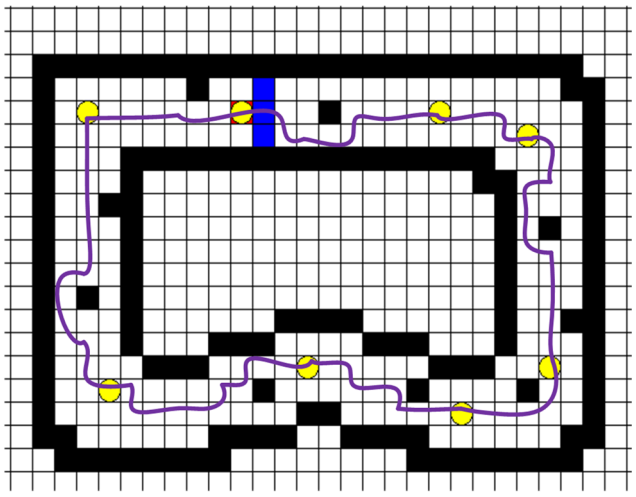


Figure 9. The approximate route of the improved algorithm.

The calculation time of the four algorithms are analyzed, including the convergence steps of their curves to be smooth, the total number of training steps, and the loss function tabulation of convergence. The data is shown in Table 2. Particular reward values are used to distinguish the four algorithms, $\mu = 0$ for Figure 7a, $\mu = 0.5$ for Figure 7b, $\mu = 10^{-6}$ for Figure 8a, and changeable μ for Figure 8b.

Table 2. Some data for four algorithms.

	μ = 0	**μ = 0.5**	**μ = 10^{-6}**	**Changeable μ**
Operation time	300 s	80 s	63 s	35 s
Convergence steps	2000	1500	750	600
Total training steps	2600	1800	5500	3100
Loss function	0.030	0.040	0.010	0.015

5. Conclusions

To solve the problem that the patrol robot can complete the loop route, an improved DQN algorithm based on deep image information is proposed. The depth image information of the obstacle is obtained by using PTZ, and then the information is directly input into the network, which improves the convergence speed of network training. By enhancing the reward and punishment functions and adding new reward value points, the reward value of the robot is improved, the problem of sparse reward in the environment state space is solved by optimizing the state-action space, and the robot's action selection is more accurate to complete the patrol task. Simulation and experimental results show that the training times and loss function values of the DQN algorithm and the improved DQN algorithm are analyzed through comparative experiments, and the effective implementation of the improved algorithm is further verified in the testing stage. The improved algorithm not only enhances the robot's exploration ability and obstacle avoidance ability but also makes the planned path length safer, which verifies the improved DQN algorithm's feasibility in path planning.

In addition to the black ring boundary, the target point, the starting point, and the intermediate obstacle point in this paper are randomly set, so the improved algorithm has strong universality. According to the previous experiments, the DQN algorithm cannot achieve the best stability and the fastest calculation speed. The optimal result of this paper is to select the optimal operating speed based on ensuring stability. In the future, more restrictive conditions can be added to verify the correctness and reliability of the improved DQN algorithm.

Author Contributions: Conceptualization, J.Z. and H.Q.; methodology, S.M.; software, S.M.; validation, S.M., Z.W. and P.K.; formal analysis, S.M.; investigation, Z.W.; resources, P.K. and Z.W.; data curation, J.Z.; writing—original draft preparation, S.M.; writing—review and editing, S.M.; supervision, H.Q.; project administration, J.Z.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under the grant number [SJCX20_0932].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing does not apply to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, Y.; Wang, J.; Duan, X. Research on Path Planning Algorithm of Indoor Mobile Robot. In Proceedings of the 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Shenyang, China, 20–22 December 2013.
2. Wang, C.; Zhu, D.; Li, T.; Meng, M.Q.H.; Silva, C.D. SRM: An Efficient Framework for Autonomous Robotic Exploration in Indoor Environments. *arXiv* **2018**, arXiv:1812.09852.
3. Candra, A.; Budiman, M.A.; Pohan, R.I. Application of A-Star Algorithm on Pathfinding Game. *J. Phys. Conf. Ser.* **2021**, *1898*, 012047. [CrossRef]
4. Rostami, S.M.H.; Sangaiah, A.K.; Wang, J.; Liu, X. Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 70. [CrossRef]
5. Zhang, Z.; Qiao, B.; Zhao, W.; Chen, X. A Predictive Path Planning Algorithm for Mobile Robot in Dynamic Environments Based on Rapidly Exploring Random Tree. *Arab. J. Sci. Eng.* **2021**, *46*, 8223–8232. [CrossRef]
6. Lynnerup, N.A.; Nolling, L.; Hasle, R.; Hallam, J. A Survey on Reproducibility by Evaluating Deep Reinforcement Learning Algorithms on Real-World Robots. In Proceedings of the Conference on Robot Learning: CoRL 2019, Osaka, Japan, 30 October–1 November 2019; Volume 100, pp. 466–489.
7. Zhang, C.; Ma, L.; Schmitz, A. A sample efficient model-based deep reinforcement learning algorithm with experience replay for robot manipulation. *Int. J. Intell. Robot. Appl.* **2020**, *4*, 217–228. [CrossRef]
8. Chen, Y.; Leixin, X. Deep Reinforcement Learning Algorithms for Multiple Arc-Welding Robots. *Front. Control Eng.* **2021**, *2*, 1.

9. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
10. Tai, L.; Li, S.; Liu, M. A Deep-Network Solution towards Model-Less Obstacle Avoidance. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016.
11. Yu, X.; Wang, P.; Zhang, Z. Learning-Based End-to-End Path Planning for Lunar Rovers with Safety Constraints. *Sensors* **2021**, *21*, 796. [CrossRef]
12. Miao, K.; Ma, J.; Li, Z.; Zhao, Y.; Zhu, W. Research on multi feature fusion perception technology of mine fire based on inspection robot. *J. Phys. Conf. Ser.* **2021**, *1955*, 012064. [CrossRef]
13. Shi, X.; Lu, J.; Liu, F.; Zhou, J. Patrol Robot Navigation Control Based on Memory Algorithm. In Proceedings of the 2014 4th IEEE International Conference on Information Science and Technology, Shenzhen, China, 26–28 April 2014; pp. 189–192.
14. Xu, H.; Chen, T.; Zhang, Q.; Lu, J.; Yang, Z. A Deep Learning and Depth Image based Obstacle Detection and Distance Measurement Method for Substation Patrol Robot. *IOP Conf. Ser. Earth Environ. Sci.* **2020**, *582*, 012002.
15. Dong, L.; Lv, J. Research on Indoor Patrol Robot Location based on BP Neural Network. *IOP Conf. Ser. Earth Environ. Sci.* **2020**, *546*, 052035. [CrossRef]
16. Van Nguyen, T.T.; Phung, M.D.; Pham, D.T.; Tran, Q.V. Development of a Fuzzy-based Patrol Robot Using in Building Automation System. *arXiv* **2020**, arXiv:2006.02216.
17. Ji, J.; Xing, F.; Li, Y. Research on Navigation System of Patrol Robot Based on Multi-Sensor Fusion. In Proceedings of the 2019 8th International Conference on Advanced Materials and Computer Science(ICAMCS 2019), Chongqing, China, 6–7 December 2019; pp. 224–227. [CrossRef]
18. Xia, L.; Meng, Q.; Chi, D.; Meng, B.; Yang, H. An Optimized Tightly-Coupled VIO Design on the Basis of the Fused Point and Line Features for Patrol Robot Navigation. *Sensors* **2019**, *19*, 2004. [CrossRef]
19. Zhao, F.; Yang, Z.; Li, X.; Guo, D.; Li, H. Extract Executable Action Sequences from Natural Language Instructions Based on DQN for Medical Service Robots. *Int. J. Comput. Commun. Control* **2021**, *16*, 1–12. [CrossRef]
20. Seok, P.K.; Man, P.J.; Kyu, Y.W.; Jo, Y.S. DQN Reinforcement Learning: The Robot's Optimum Path Navigation in Dynamic Environments for Smart Factory. *J. Korean Inst. Commun. Inf. Sci.* **2019**, *44*, 2269–2279.
21. Sasaki, H.; Horiuchi, T.; Kato, S. Experimental Study on Behavior Acquisition of Mobile Robot by Deep Q-Network. *J. Adv. Comput. Intell. Intell. Inform.* **2017**, *21*, 840–848. [CrossRef]
22. Han, B.; Zhao, Y.; Luo, Q. Walking Stability Control Method for Biped Robot on Uneven Ground Based on Deep Q-Network. *J. Beijing Inst. Technol.* **2019**, *28*, 220–227.
23. Rahman, M.M.; Rashid, S.; Hossain, M.M. Implementation of Q learning and deep Q network for controlling a self balancing robot model. *Robot. Biomim.* **2018**, *5*, 8. [CrossRef]
24. da Silva, I.J.; Perico, D.H.; Homem TP, D.; da Costa Bianchi, R.A. Deep Reinforcement Learning for a Humanoid Robot Soccer Player. *J. Intell. Robot. Syst.* **2021**, *102*, 69. [CrossRef]
25. Peng, X.; Chen, R.; Zhang, J.; Chen, B.; Tseng, H.W.; Wu, T.L.; Meen, T.H. Enhanced Autonomous Navigation of Robots by Deep Reinforcement Learning Algorithm with Multistep Method. *Sens. Mater.* **2021**, *33*, 825. [CrossRef]
26. Tallamraju, R.; Saini, N.; Bonetto, E.; Pabst, M.; Liu, Y.T.; Black, M.J.; Ahmad, A. AirCapRL: Autonomous Aerial Human Motion Capture using Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6678–6685. [CrossRef]
27. Abanay, A.; Masmoudi, L.; Elharif, A.; Gharbi, M.; Bououlid, B. Design and Development of a Mobile Platform for an Agricultural Robot Prototype. In Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems, Larache, Morocco, 14–16 November 2017; pp. 1–5.
28. Budiharto, W.; Santoso, A.; Purwanto, D.; Jazidie, A. A method for path planning strategy and navigation of service robot. *Paladyn* **2011**, *2*, 100–108. [CrossRef]
29. Arvin, F.; Samsudin, K.; Nasseri, M.A. Design of a Differential-Drive Wheeled Robot Controller with Pulse-Width Modulation. In Proceedings of the 2009 Innovative Technologies in Intelligent Systems and Industrial Applications, Kuala Lumpur, Malaysia, 25–26 July 2009; pp. 143–147.
30. Bethencourt, J.V.M.; Ling, Q.; Fernández, A.V. Controller Design and Implementation for a Differential Drive Wheeled Mobile Robot. In Proceedings of the 2011 Chinese Control and Decision Conference (CCDC), Mianyang, China, 23–25 May 2011; pp. 4038–4043.
31. Zeng, D.; Xu, G.; Zhong, J.; Li, L. Development of a Mobile Platform for Security Robot. In Proceedings of the 2007 IEEE International Conference on Automation and Logistics, Jinan, China, 18–21 August 2007; pp. 1262–1267.
32. Sharma, M.; Sharma, R.; Ahuja, K.; Jha, S. Design of an Intelligent Security Robot for Collision Free Navigation Applications. In Proceedings of the 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), Faridabad, India, 6–8 February 2014; pp. 255–257.