Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.028126
Article

Tech Science Press

Artificial Potential Field Incorporated Deep-Q-Network Algorithm for Mobile Robot Path Prediction

A. Sivaranjani[1,*] and B. Vinod[2]

[1]Department of Robotics and Automation Engineering, PSG College of Technology, Coimbatore, 641004, India
[2]Department of Electrical and Electronics Engineering, PSG College of Technology, Coimbatore, 641004, India
*Corresponding Author: A. Sivaranjani. Email: asr.rae@psgtech.ac.in

Received: 03 February 2022; Accepted: 13 March 2022

Abstract: Autonomous navigation of mobile robots is a challenging task that requires them to travel from their initial position to their destination without collision in an environment. Reinforcement Learning methods enable a state action function in mobile robots suited to their environment. During trial-and-error interaction with its surroundings, it helps a robot to find an ideal behavior on its own. The Deep Q Network (DQN) algorithm is used in TurtleBot 3 (TB3) to achieve the goal by successfully avoiding the obstacles. But it requires a large number of training iterations. This research mainly focuses on a mobility robot's best path prediction utilizing DQN and the Artificial Potential Field (APF) algorithms. First, a TB3 Waffle Pi DQN is built and trained to reach the goal. Then the APF shortest path algorithm is incorporated into the DQN algorithm. The proposed planning approach is compared with the standard DQN method in a virtual environment based on the Robot Operation System (ROS). The results from the simulation show that the combination is effective for DQN and APF gives a better optimal path and takes less time when compared to the conventional DQN algorithm. The performance improvement rate of the proposed DQN + APF in comparison with DQN in terms of the number of successful targets is attained by 88%. The performance of the proposed DQN + APF in comparison with DQN in terms of average time is achieved by 0.331 s. The performance of the proposed DQN + APF in comparison with DQN average rewards in which the positive goal is attained by 85% and the negative goal is attained by −90%.

Keywords: Artificial potential field; deep reinforcement learning; mobile robot; turtle bot; deep Q network; path prediction

1 Introduction

The challenge of steering a mobile robot in an unknown area is known as motion planning. Navigating local paths is a common and practical problem in autonomous mobile robotics research. The autonomous robots use a tool path to choose the best path from point A to point B without colliding with any barriers [1]. The proposed approach for mobile robots is in the face of increasing scientific and technological breakthroughs is currently confronted with a complicated and dynamic world [2]. The traditional path planning algorithms lack certain salient merits such as least working cost and minimal processing time. To overcome these limitations, Reinforcement Learning (RL) has been proposed based on current developments. The Deep neural network algorithm is a part of RL. The Deep Q-Network (DQN) is utilized to train the TB3 in a Robot Operating System (ROS) simulation environment. TB3 is a robot is a term that is frequently used in robotic systems. In this research, TB3 Waffle Pi is selected over TB3 Burger due to its additional features such as enhanced sensing capabilities, better computing and higher power available to drive the wheels. This aids in handling heavier payloads.

ROS is an accessible robotics framework that comprises a variety of open-source tools and applications for building complex mobile robots [3]. It creates a transparent graph that allows robotics programming to be written and merged in a flexible and orderly manner, such as low-level firmware, control algorithms, sensory perceptions methodologies, navigation algorithms, and so on. Without the use of any programming, ROS may connect with Gazebo directly. modifications to execute in simulations rather than the actual world [4]. It enables the entire robotics computer system to work in real-time while being advanced to the simulation's desired pace. ROS Melodious is the ROS dispersion that was employed in this study. The necessary tools for training the DQN agents are provided by ROS and Gazebo.

DQN Agent's mission is to get the TB3 to its target without colliding. TB3 receives a positive reward when it moves closer to its objective, and a bad incentive when it moves further away. The episode terminates when the TB3 collides with an obstruction or after a set amount of time has passed. TB3 obtains a tremendous positive bonus when it arrives at its target and a massive poor reward when it smashes into an obstacle during the episode. The DQN approach was presented by Tai et al. for robot route optimization simulators. Exaggeration of action value and sluggish fast convergence were determined to be the DQN individual's flaws [5].

To overcome this, the shortest path algorithm is included with DQN to achieve better results and to primarily reduce the training time of the DQN algorithm. For effective path planning, the Artificial Potential Field (APF) method is used with DQN. In the coordinate space, the artificial field consists of a repellent vector field and an attractive prospective field. The resulting force is the consequence of the attracting and negative charges combining. The size and direction of the resultant force define the robot's mobility state.

This algorithm's Efficient quantitative equations and simplicity is commonly used for automatic guided mobile robot navigation [6–8]. This method is frequently used to solve the local minima dilemma, which arises when the total force applied to a robot is equal to zero. Alternative approaches for avoiding local minima have been documented in several research projects, such as altering the Gaussian function of potential and using a sample selection scheme. This could force the robot to take a longer route. This problem is not considered in this study because, both DQN and APFs are used to find the next step of the robot. The advantages of combined APF and DQN are minimum training time and that, the robot takes the shortest path to reach the goal.

Yang et al. had explained about the fully automated warehouse or stock-keeping units performing the "Goods to People" model majorly based on the material handling robot. With this robot, the industry can conserve manpower and increase the efficiency of the plant by eliminating manual material handling and transportation. This article had utilized the Deep-Q-Network algorithm with Q-learning methods are combined with an actual approach and neural network perceived loudness innovations in relevance feedback. As a result, the goal of this paper is to answer the problem of multi-robot trajectory tracking is fully automated warehouses or stock-keeping units [9].

Bae et al. had analyzed the effectiveness for accomplishing specific tasks, several practical methods to the challenge of multi-robot path planning. Moreover, all the robots involved in this group operate individually as well as cooperatively depending on the given scenarios, thus the area of search is improved. Reinforcement Learning in any path planning approach gives major focus to fixed space where each object has interactivity [10].

Yu et al. (2020) had explained that by observing the scene and performing extraction of features, the fitting obtained from the state action function was achieved using neural network models. The mapping of the current state to hierarchy relevance feedback was achieved with the help of the enhancement function, allowing the robot to become even more mobile. Route optimization efficiency in network should be improved robotic systems, these two methodologies were naturally merged. This research has yet to conduct experiments on dynamical environments and related scenarios. But emphasizes theoretical accuracy with the conducted research [11].

Wang et al. (2020) had described that the dynamic path planning algorithm is incapable of solving problems related to wheeled mobile robots with scenarios including slopes and dynamic obstacles constantly moving at their rate. The Tree-Double Deep Q Network technique for variable trajectory tracking in robotic systems is suggested in this research. The Q Network with a Double Deep is used in this approach. It rejects the incomplete and outlier-based detected paths by improving the performance. The DDQN approach is combined with the tree-based method because of the binary tree. This study took the best option available in the present state and performed fewer activities, resulting in a path that fits the restrictions. Eventually according to the obtained state-based results were repeated to the plurality of eligible paths. This research utilizes ROS simulations and practical experimentations to verify the hypothesis [12].

Ali et al. (2019) had explained about a Laser Range Finder's LS and sensor fusion (LRF) The sensor to reduce noise and ambiguity from sensor information and provide optimal path planning strategies, a fusion method is used. The results from the experiment show that the methods can safely drive the robot in road driving and circle situations. During navigation, motion control employing twin feedback mechanisms based on the RAC-AFC controller is employed to track the WMR's planning path while rejecting the disruption. For a novel WMR platform that would navigate in a road roundabout environment, localization and control algorithms are developed. A Resolution Accelerated Control (RAC) combined with active control (AFC) for rejecting disturbances is utilized to control the kinematics characteristics of WMR [13].

Sun et al. (2019) had analyzed the number of samples necessary for random exploration grows increases with the number of steps required to achieve a reward in the reward curricular training approach. Displays outstanding capacity to migrate to different unfamiliar places and good planning skills in a Mapless world. The mechanism can also withstand current disturbances. The paper presents a reward-based training strategy for curricular learning. Using sensor data as input and surging force and yaw moment as output, the system achieves motion planning through the policy. This strategy addresses the issue of sparse rewards in complicated contexts while avoiding the poor training effects of intermediary rewards [14].

Lin et al. (2013) had explained the TSK-type neural fuzzy networks are designed using a novel Adaptive Group Organization Cooperative Evolutionary Algorithm (AGOCEA). To autonomously create neural fuzzy networks, the suggested AGOCEA employs a group-based cooperative iterative method and a self-organizing technique. It can dynamically estimate the parameters of adaptive neuro-fuzzy inference networks, eliminating the need to assign some critical parameters ahead of time. Our model has a shorter rescue time than a standard model that uses a static judgement method, according to simulation data [15].

Xin et al. (2017) had explained to simulate the mobile robot state-action value function, It is developed and trained a deep Q-network (DQN). Without any finger features or feature detection, the DQN receives the original Rgb values (image pixels) from the surrounding. Finally, while traveling, the mobile robot reaches the objective point. Our hard-to-learn robot path planning method is a fantastic end-to-end mobile robot trip planning solution method, according to the experimental data. This research proposes a revolutionary end-to-end path planning approach for mobile robots utilizing a Deep Learning approach [16].

Liu et al. (2015) had analyzed topological recognition quality criteria, notably for semi-structured environments in 2D. Then we show how to develop topological separation for semi-structured settings using an incremental technique. Our unique method is based on spectral clustering of discrete-time metric maps decomposed using a modified Voronoi breakdown incrementally The robustness and quality of the topology providing an overview using the proposed approach are demonstrated in real-world experiments [17].

Mnih et al. (2015) had explained Agents must construct efficient models of the surroundings based on increased sensory information, and use them to transfer previous experience to a whole new situations condition to use learning algorithm effectively in settings nearing real-world complexities. We show that using only the pixels and the game score as inputs, the deep Q-network agent was able to outperform all previous methods and achieve a level similar to that of a professional human game. The gap between top sensory information and actions is bridged in this research [18].

An end-to-end Tree branch embedding network has been established by Sun et al. (2021) [19] for the precise feature extraction of the vehicles. The local and global features of the vehicles are utilized for the re-identification of the vehicle through the cameras. Mainly the region features of the vehicles are emphasized for better prediction using colour image analysis. The training of the dataset results in identifying the similarities and differences between the target and other vehicles. Further enhancement in the proposed has to be included by adding an adaptive technique for further efficiency in the prediction.

A deep learning-based object detection strategy for the traffic information collection using Unmanned aerial vehicle has been proposed by Sun et al. (2021) [20]. A real time small object detection algorithm incorporated YOLOv3 has been utilized to detect the objects in traffic monitoring and provides specific information. The local and global features are fused and are emphasized by the feature Pyramid network. The work has to be enhanced with further optimization for enhanced prediction.

This paper contributes a Reinforcement Learning approach that enables efficient path prediction for robot navigation in the environment based on ROS system using Applied Potential Field induced Deep-Q-Network.

2 Materials and Methods

Within certain limitations, the purpose of path planning is to determine an optimal or near-optimal accident path from a preliminary step to a destination point. Time, path length, closeness to barriers, and kinematic are all factors to consider. are all possible restrictions. These approaches necessitate an environmental map or operating parameters. A*, Artificial Potential Field, Probabilistic Roadmap, Rapidly Exploring Random Tree (RRT), and Message passing RRT are the most extensively utilized route planning techniques. Because of its efficiency and simplicity, the Artificial Potential Field algorithm is widely employed in independent mobile robot path plan research. So, this algorithm is further chosen in this research and it is combined with the DQN algorithm to get better results.

Deep Q Network is a method of RL that combines neural networks and Q learning. RL algorithms It doesn't have close oversight or a complete model of the world; instead, it learns by engaging with it repeatedly, with a reward signal as feedback for the model's effectiveness [21]. The purpose is to increase the performance over time with the given constraints. The training time and the path length is reduced and the number of successful targets is increased while using the proposed modified algorithm. Fig. 1 indicates the flow diagram of the proposed approach.

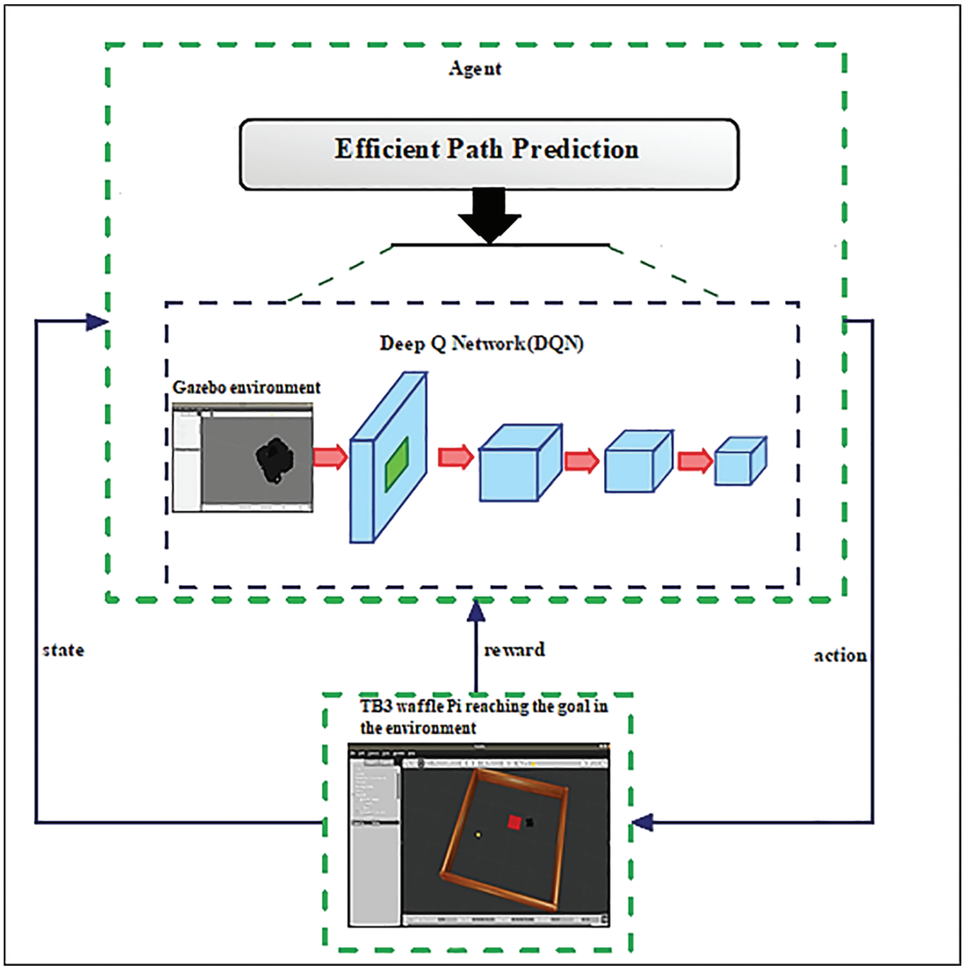

Agent
Efficient Path Prediction
Deep Q Network(DQN)
Gazebo environment
state
reward
action
TB3 waffle Pi reaching the goal in the environment

Figure 1: Flow diagram of the proposed approach

2.1 DQN Algorithm

The Markov characteristic is assumed to apply to the majority of RL problems. It asserts that subsequent states are determined only by the present state, not by previous ones. Making the Markov hypothesis permits reinforced learning issues to be formally stated as Markov Decision-Making (MDPs), which are defined by their states, actions, and simulated results between states.

Answering an MDP entails determining the best policy, with a policy determining what action should be taken in every given state. In Eq. (1), it is described as the projected sum of discounted future benefits moving from that condition under the policy. The sum of reduced incentives is denoted by Gt, where is the discount factor. A sale price near to one indicates that long-term incentives are weighted similarly to short-term rewards, but a reduced interest factor indicates that the individual is myopic and only cares about prizes that are due this month in Eq. (2).

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}; \quad 0 \leq \gamma \leq 1 \tag{1}$$

$$v_\pi(s) = E_\pi[G_t| \ S_t \ = s] \tag{2}$$

Eq. (3) can be used to explain the government action relationship. If the activity ‘a’ is performed from state’s and policy is implemented, this is the sum of reduced incentives. The Q-value of that state-action pair is defined as the activity of countries and action ‘a’.

$$q_\pi(s, \ a) = E_\pi \ [G_t| \ S_t \ = s, \ A_t \ = a] \tag{3}$$

The optimum action to perform in every given state is described by the optimal policy of an RL issue. If action is conducted in the initial state, the predicted value of the compounded return of future benefits will be maximized. Finding the optimum solution is easy if the optimum action functional form q (s, a) is known. The robot is just doing the activity with the highest q-value in each phase it finds itself through. Making this decision in each phase would effectively maximize the agent’s payoff [22,23]. To produce greater results, DQN employs several critical advances The key innovation of DQN was to utilize the deep convolutional neural networks [24]. In addition, DQN employs an experience replay buffer. The N past transition experiences is sent to a database D_t = e_1, …, $e_{t, \text{ during}}$ learning. Each phase includes batch updates for training the Q network, which is evenly sampled from D, (s, a, r, s0) U. (D). Smoothing out training, lowering having to learn variance, preventing oscillation, and preventing bias are all benefits of memory replay. To avoid fluctuations in the data, the final invention was to employ a delayed and fixed version of the Q-network for target estimate. DQN uses the implied error of the Q-value predictions to compute Q changes. For updating step, this is described in the following loss function in Eq. (4).

$$L_i(\theta_i) = E_{(s,a,r,s')\sim U(D)}[(Y_i^Q - Q(s, \ a; \ \theta_i))^2 \tag{4}$$

$$Y_i^Q = r + \gamma \underset{a'}{max} \ Q(s', \ a'; \ \theta_i^-) \tag{5}$$

The gradient required to execute transfer learning to update Q (s, a; I) is the derivatives with appropriate weights of this loss function i. To send out tailored reminders. The primary network. Q (s, a; θ_i) is used by DQN in a delayed and fixed version. The target network characteristics are changed every cycle to match the main network characteristics, as shown in Eq. (5). When a goal network is not used, the same network is used to generate the target.

Another DQN algorithms improvement is providing a Q-value for each action in the multilayer perceptron. This is important because it allows all Q-values to be estimated using only one forward transit across the system. A Q-network, on the other hand, accepts a condition and a probable action as input and produces a single Q-value for that action as output. The procedure can be made substantially faster by estimating all Q-values with a single forward pass. Fig. 2 is an example of Deep Learning.

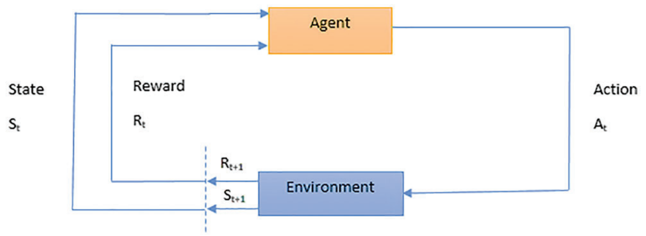


Figure 2: Illustration of deep learning

2.2 Artificial Potential Field

APF's prospective force has both attractive and repulsive forces. The mobile robot advances it toward the objective location since it generates an attractive force, while the barriers repel it [25,26]. Concerning the potential field and the map, the robots can locate its location. The mobile robot can determine its next essential action depending on the field. If new impediments surface while the robot moves, the prospective field could be updated to incorporate this new knowledge. The associated synthetic force F (q), acting at the location q = (x, y), is found using the discrete potential field component U(q), as shown in Eq. (6).

$$F(q) = -\nabla U(q); \quad \nabla \ U = \begin{pmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{pmatrix} \tag{6}$$

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{7}$$

$$F(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) \tag{8}$$

$\nabla \ U(q)$ U at location q is denoted by U(q). Eq. (7) demonstrates the possible field operating on the robot as the sum of the attractive field of the objective and the repellent field of the impediments. In the same way, the pressures can be split into an attractive and repelling half, as illustrated in Eq. (8). A parabolic distribution, such as in Eq. (9), can be used to define a value-added.

$$U_{att}(q) = \frac{1}{2} k_{att} \ \rho^2_{goal} \ (q) \tag{9}$$

$\rho^2_{goal} \ (q) \ F_{att}$ denotes the Distance measure, and k_{att} is a constant scale parameter. It is distinguishable, as indicated in Eq. (10), resulting in the attractive force F att. The repellent perspective is capable of producing a force that repels all known impediments. Whenever the robot is close to the thing, this ought to be quite strong, but when the robot is far away from the object, it should have little effect on its motion.

$$F_{att}(q) = -k_{att} \ (q - \ q_{goal}) \tag{10}$$

$$U_{rep}(q) = \begin{cases} \frac{1}{2} \quad k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_o} \right)^2, & if \ \rho(q) \le \rho_o \\ \quad 0, \quad if \ \rho(q) \ \ge \rho_o \end{cases} \tag{11}$$

$$F_{rep}(q) = \begin{cases} k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_o} \right) \ \frac{1}{\rho^2(q)} \frac{q - q_{obstacle}}{\rho(q)}, & if \ \rho(q) \le \rho_o \\ 0, & if \ \rho(q) \ \ge \rho_o \end{cases} \tag{12}$$

where k rep is a scale parameter, _o is the individual's impact separation, and q rep is the shortest distance between q and the object. Eq. (11) depicts the repelling virtual infrastructure $\rho(q)$, which can be positive or negative and approaches infinity as q approaching the object [27]. (q) is recognizable everywhere in the free coordinates space if the object border is convex and piece-wise distinct. The repelling pressure F rep is the result of this, as shown in Eq. (12). The actions that cause $(q) = F_{att}(q) + F_{rep}(q)$ to be applied to a point robot subjected to attracted and repellent forces cause the TB3 to move away from the barriers towards the target. $k_{rep} \ U_{rep} \ F_{att}(q) + F_{rep}(q)$.

Fig. 3 illustrates the overall structure U_{rep} of the proposed method for robot path prediction utilizing deep RL and the shortest technique.

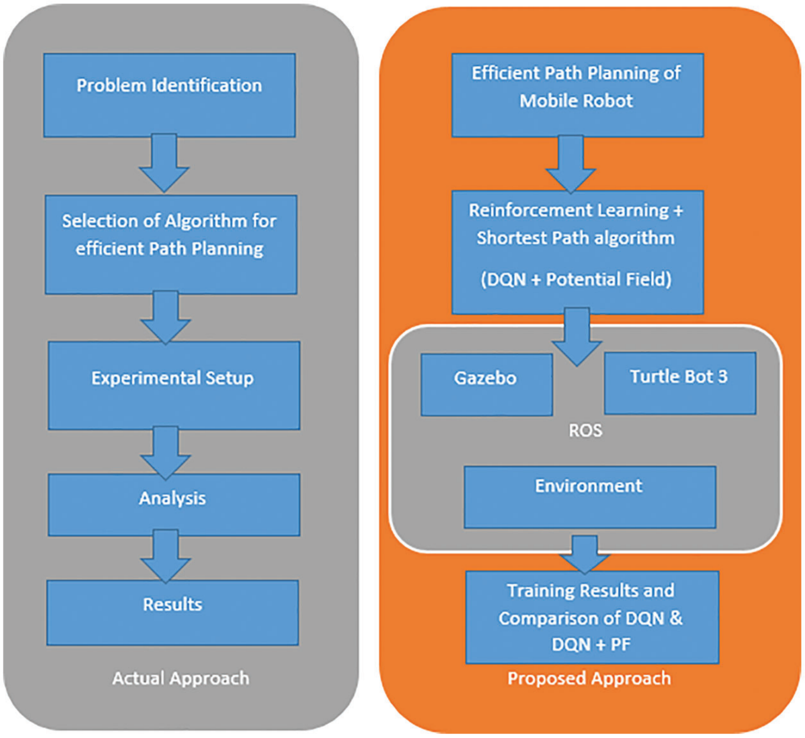

Problem Identification
Selection of Algorithm for efficient Path Planning
Experimental Setup
Analysis
Results
Actual Approach
Efficient Path Planning of Mobile Robot
Reinforcement Learning + Shortest Path algorithm
(DQN + Potential Field)
Gazebo
Turtle Bot 3
ROS
Environment
Training Results and Comparison of DQN & DQN + PF
Proposed Approach

Figure 3: Illustration of proposed approach of the research

2.3 Proposed Algorithm

The mobile robot should navigate from one place to another place without colliding with any obstacles. The processing time and the path length should be minimum while it travels from start to goal position. Sampling-based algorithms were used in the path planning of a mobile robot. The state-action value function of a mobile robot is approximated by a deep Q-network (DQN). The well-trained DQN then determines the Q value for each possible mobile robot action (radius left, radius right, the direction forward).

The TurtleBot3 (TB3) is trained in a Robot Boot Loader (ROS)–Gazebo simulation environment using the Fully Convolutional (DQN), which is a blend of RL and Deep Neural Networks. ROS is an open-source morpho platform that features technology separation and delivers services from a Linux kernel. TB3 is a ROS-based robot that is small, configurable, and utilized in education and research. It's a popular tool in robotic systems. The goal of TB3 is to drastically reduce the product's size and weight while sacrificing its function or performance. There are three different versions of TB3. The TB3 Waffle Pi was chosen since it offers more benefits than the other two models. A reward is given to TB3 when it acts in a state. A positive or negative reward might be given. When TB3 achieves its goal, it is rewarded handsomely. When it comes across a barrier, it receives serious negative compensation.

TB3 Waffle Pi is trained by DQN to reach the goal. The TB3 decides what action to perform at each time step based on the present situation to optimize future benefits. The state must a sufficient volume of knowledge from the surroundings for the agents to make the best decision possible. A series of the positional versatility and three prior depths photos were taken by a depth sensor was defined as the state. In an empty and obstacle-free area, the agent is educated. The agent's instruction is divided into chapters, each of which begins with the robot's start and objective locations being chosen from a list of options. The training average time, average rewards, and the number of successful targets is identified for the DQN algorithm. It took more time for training and the number of successful targets is also less.

To reduce the training time and maximize the rewards, the shortest path algorithm, APF is incorporated in the DQN algorithm. One of the algorithms mostly used by the researchers is Artificial Potential Field. APF gives minimum processing time and path length. But, when the algorithm combines with Reinforcement Learning, it gives a greater number of successful targets and takes minimum average time to reach the goal autonomously with a reduced lifetime.

A succession of depth photos collected from a depth camera, combined with an angle denoting the heading toward the intended target, was chosen as the state. The effectiveness of the suggested strategic planning is evaluated in a simulated world based on the Robot Operation System (ROS) and contrasted to the conventional method DQN algorithm. The number of successful targets, average training time, and average rewards has been taken to validate the results. The proposed hybrid algorithm gives a greater number of successful targets. Also, it took less training time, and maximize rewards.

While navigating from start to goal position; at first two points are generated, one is by DQN, and another is by APF. Always it chooses the next point from starting point only with the DQN algorithm, but it verifies with APF, whether the chosen point is optimal or not. The robot heading and goal are shown in Fig. 4.

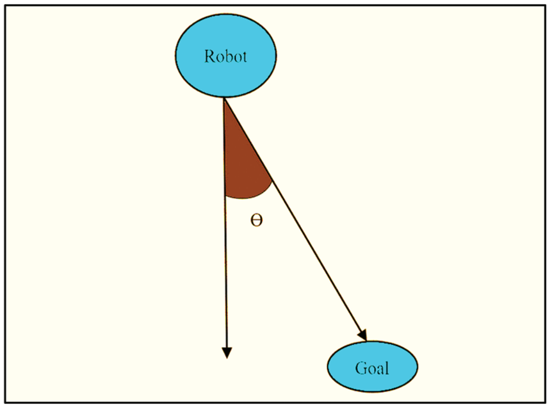


Figure 4: Robot heading and goal

The agents should be rewarded for navigating the robots going away from the goal and colliding with barriers and being fined for traveling away from the goal and hitting with hurdles in mobile robot navigation. To do so, the reward function, r, was created as shown in Eq. (13).

$$r = \cos\theta \tag{13}$$

If the value of the reward function i.e., cosine angle between the two points is positive then the robot will move on to the next location since the location is ideal which is selected by DQN. Else the robot collides in the environment.

To solve the inherent instability problem of using compliance in RL, DQN used two strategies: experience replaying and targeted networks. Experiencing replay memory, which store transitions of the form (st, at, st + 1, rt + 1) in a cyclic buffer, allow the Agent to sample from and learn on previously witnessed data. This not only reduces the number of interactions with the environment but also allows for the sampling of information batches, minimizing the variability of learning upgrades. Furthermore, the temporal correlations that can harm RL systems are eliminated by sampling evenly from a vast memory. From a practical standpoint, current equipment can easily manage batches of data simultaneously, improving capacity. While the initial DQN system employed uniform sampling, subsequent research has revealed that prioritizing samples based on TD errors is more advantageous for learning purposes. Reinforcement learning functions are shown in Fig. 5.

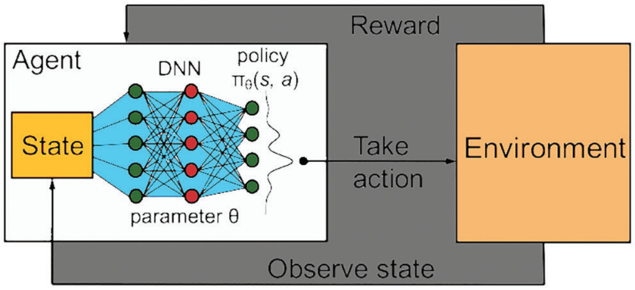


Figure 5: Deep reinforcement learning value functions

2.3.1 Reward System

The external environment and subjective objects define reward function. The advantages of the incentive function design have a significant impact on the speed and quality of teaching and learning. It is vital to precisely define the robot's activities to train the network to develop a viable control plan. Unlike simple directives like "continue" or "turn left or right," the activities in our network are defined in discrete forms to manage connection speed and rotational velocity. Allowing your robot to run as fast as possible while punishing it with simple *in situ* spins is the bonus function. The total plot reward is the sum of all instantaneous payouts for each plot phase. This event is triggered if a collision is detected. Allowing the robot to run as fast as possible while punishing it with simple *in situ* spins is the bonus function. The total plot reward is the sum of all instant payouts for each plot phase.

2.3.2 The DQN and Function Approximation

Fig. 6 shows **t**he DQN, it is fed four grey-scale gaming images that have been concatenated over time as inputs, which are then evaluated by several convolutional layers to extract spatiotemporal properties such as ball handling in Pong or Break. The final feature map from the convolutional layers is analyzed by several fully connected layers, which more naturally encode the effects of actions. Typical controllers, on the other hand, use a set that was before steps and thus cannot change their status processing in response to the learning signal. Or, in this context, the value of all acts in a distinct set of actions, the joystick's different orientations and the fire button. This not only allows the network to choose the best action (s, a) after a single backward pass, but it also makes it easier for the system to encode initiative information in the lower convolutional layers. With the sole objective of improving its video gaming record, the DQN learns to extract significant visual data while simultaneously encoding objects, their movements, and, most importantly, their relationships.

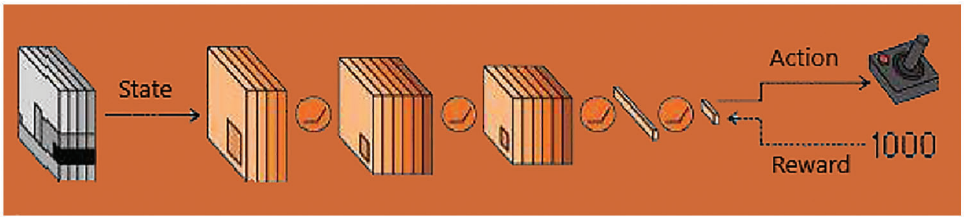


Figure 6: The DQN

That if such a database could be created, it would be poorly occupied, and knowledge received from one state-action pair could not be spread to others. The DQN's strength rests in its ability to use deep neural networks to cohesively encode both strong data and the Q-function. It would be impossible to tackle the discrete Atari realm from raw visual inputs without such a capability.

The policy system utilizes the fixed targeted system instead of calculating the TD error based on its own rapidly fluctuating estimates of the Q-values. The parameters of the network device are adjusted to match the policy network after a set number of steps during learning. Both the encounters replay as well as the target systems have since been employed in other DRL experiments.

The positioning refers to the process of the state–a stack of grey-scale frames from the video game—with convolution and fully connected layers, as well as nonlinear effects between every level. The network outputs a discrete action at the last layer, which correlates to one of the game's potential control inputs. The game generates a fresh score based on the current state and action taken. The DQN learns from its choice by using the reward—the difference between the new and prior score. The incentive is used to refresh the channel's estimation of Q, and the difference between both the old and new estimates is backpropagated through the system. The design of a Dueling Deep Q-network changes a standard DQN into a Reinforcement Learning architecture more suited to model-free reinforcement learning, with the goal of lowering loss as much as feasible. A typical DQN design is made up of a stream of completely linked layers, but the Dueling DQN divides the stream into two parts: one for the value function and the other for the advantage function. Double DQN and Dueling DQN are combined in Dueling Double DQN. As a result, the estimating problem is solved, and efficiency is better. The training sequence s is viewed as a Markov Decision Process as the traditional reinforcement learning approach (MDP). Engaging with the environment allows the robot to make choices. Moving ahead, turning half-left, turning left, turning left-right, a rightmost lane is some of the activities.

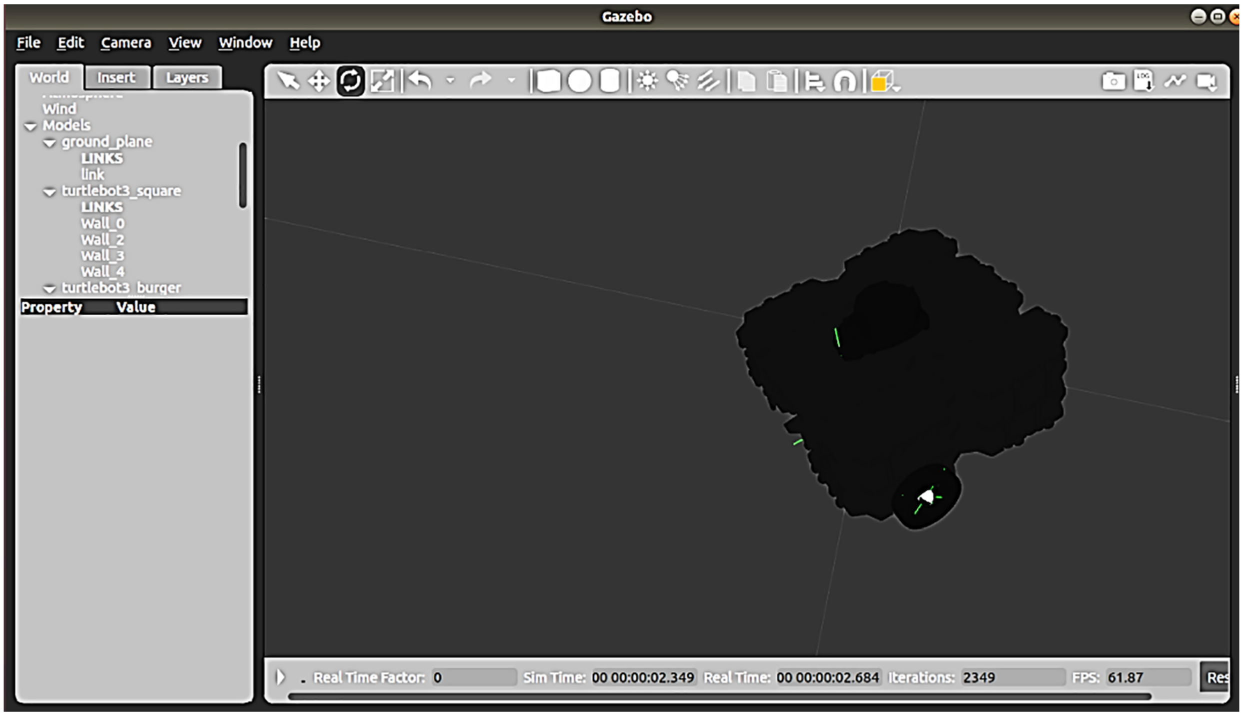
2.3.3 DQN Based on Deep Reinforcement Learning

The suggested navigation technique can successfully achieve autonomous and collision-free motion of the robot to the location of the target object without constructing the environmental map in advance, according to simulation navigation experimental data. It is a successful autonomous navigation technique that demonstrates the viability of Deep Reinforcement Learning in mobile robot automated driving. This method's use is confined to discontinuous action space agents. However, some early works have used the DQN methodology to learn optimal actions from visual input. Because the convolutional neural network can automatically extract complicated features, it is the best choice when dealing with high-dimensional and continuous state data. The network is improved by Dueling DQN. It takes advantage of the model structure to express the value function more thoroughly, allowing the model to perform much better and reducing the overestimation of the DQN Q value.

3 Experimental Setup, Results and Discussions

The RL's education to fully imitate robot dynamics, the agent proposed in this work will require a simulation that can replicate robot motions in real-time, replicate depth cameras to provide depth images for training, support a wide range of obstacles, and be a complete physics simulator [28,29]. The gazebo is the free software simulation that best meets the parameters for training the navigation algorithm [20–31]. It offers physical and sensor simulations that are quicker than genuine. The capacity to create worlds to replicate models of experimental robots is provided by the Robot Operating System, an open-source robotics software framework (ROS).

The simulations are performed on ROS Melodic distribution and Gazebo that operate on Ubuntu 18.04. TB is an industrial automation robot that is frequently utilized [32,33]. The latest version TB3 Waffle Pi is selected here because of more advantages than previous versions. The TurtleBot3 series by Robotics offers a range of open-source robotics platforms that are programmable on ROS, and highly powerful and performing despite their small size and low price. These mobile robots are equipped with a full autonomous navigation system, 2 servos, a motion controller, and a microcontroller. A Turtlebot3 Burger is a lightweight, compact, cost-effective, adaptable, and lightweight mobile robotics platform. It includes a high SBC with Intel CPU in terms of computations, a True Sense Application Of deep learning for object detection and 3D SLAM, and an increased power actuator that ensures a maximal linear speed of 0.26 m/s and an angular speed of 1.8 rad/s thus the iteration can be reduced using Pi. Fig. 7 shows the TB3 Waffle Pi in a ROS-Gazebo context.

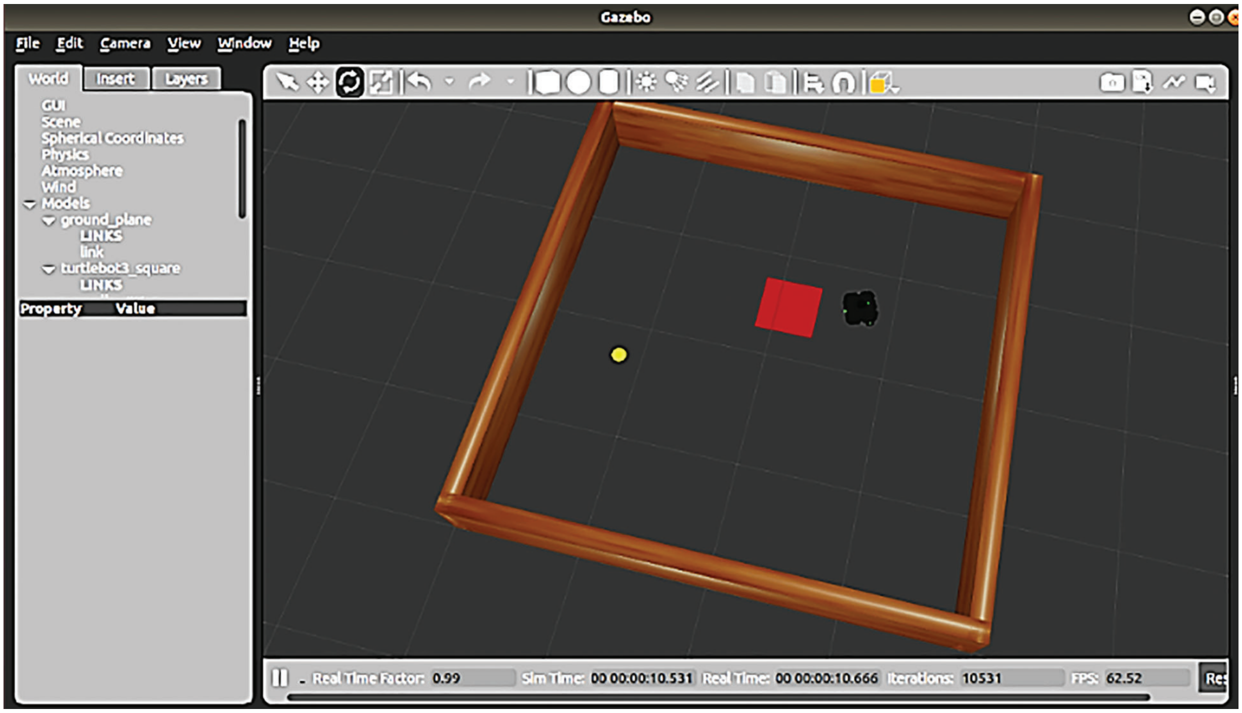


Real Time Factor: 0 Sim Time: 00 00:00:02.349 Real Time: 00 00:00:02.684 Iterations: 2349 FPS: 61.87

Figure 7: TB3 waffle Pi in ROS-gazebo environment

The 4 × 4 map with no obstacles is taken as the environment while the four walls are considered as obstacles. The TB3 should reach the goal without colliding the walls. The main aim of the research is to test the efficiency of the conventional and proposed algorithms. The environment which is selected in this research is shown in Fig. 8.

In this environment, the TB3 experiments with the conventional DQN algorithm and the proposed DQN algorithm. The results are compared in the terms of the number of successful targets, average time (seconds), and average rewards. In the training period, the proposed algorithm gives more successful targets compared to DQN. The comparison of results is tested with 1000 episodes where, an episode is the ratio of a single batch of datasets per the respective tasks. An episode usually means one single dataset. In the 50th episode, the DQN algorithm gives only 6 successful targets but the proposed algorithm gives 10 successful targets. As the episode increases the number of successful targets also increases. In the 500th episode, the proposed algorithm gives 349 successful targets while the conventional algorithm gives only 295 successful targets. So, when compared with the conventional algorithm the proposed algorithm gives more successful targets as shown in Fig. 9. The performance improvement rate of the proposed DQN + APF in comparison with DQN in terms of the number of successful targets is attained by 88%.



Real Time Factor: 0.99 Sim Time: 00 00:00:10.531 Real Time: 00 00:00:10.666 Iterations: 10531 FPS: 62.52

Figure 8: TB3 waffle Pi reaching the goal in the environment

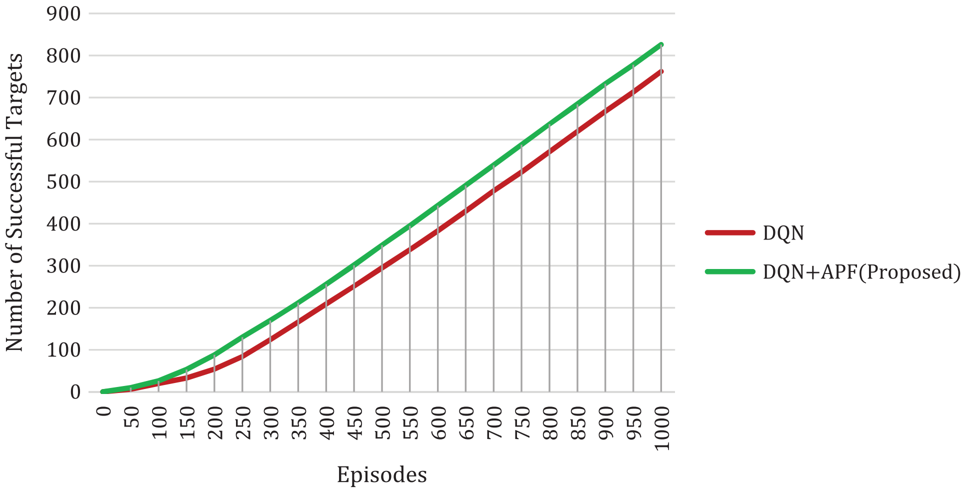


Figure 9: Comparison of algorithms concerning the number of successful targets and episodes

The average time taken to train the TB3 Waffle Pi is less in the proposed algorithm when compared to the conventional DQN. With the reward function, a direct reaction is the final notches the robot can achieve, with the trained method for selecting the knowledge with the highest Q-value at this point in the series, we've reached the 50th segment. The average time taken for the conventional algorithm is 108 s but the proposed algorithm takes only 81 s to reach the target. In the initial training period that is from the 50th episode to the 400th episode the average time taken is more for the DQN algorithm. But the proposed algorithm took very little time as shown in Fig. 10. The performance improvement rate of the proposed DQN + APF in comparison with DQN in terms of average time is attained by 0.331 s.

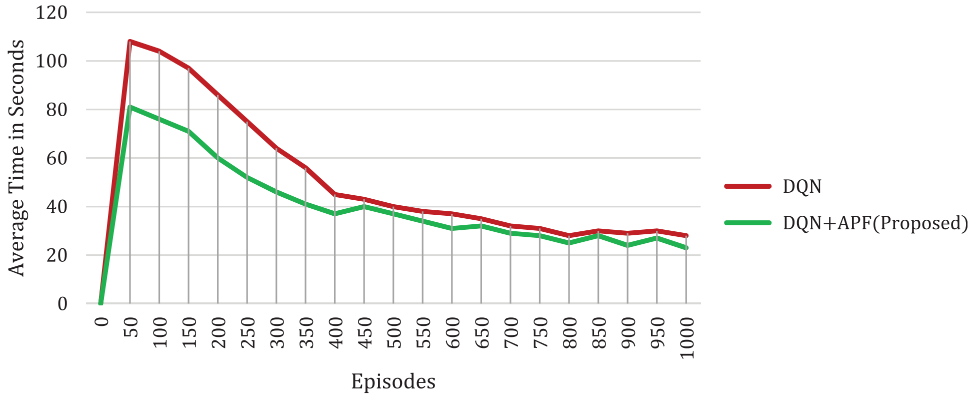


Figure 10: Comparison of algorithms concerning the time taken in seconds and episodes

When TB3 takes an action in a state, it receives a reward [10]. The reward design is very important for learning. A reward can be positive or negative. When TB3 gets to the goal, it gets a big positive reward. When TB3 collides with the wall it gets a negative reward. In the initial training from the 50th episode to the 250th episode, the conventional DQN algorithm gets a negative reward only. So, the number of successful targets will be very less as shown in Fig. 8. But in the proposed algorithm the 250th episode itself gets a positive reward. Between the 300th episode to the 550th episode, the proposed algorithm gets a better average reward function than the DQN algorithm as shown in Fig. 11. The performance of the proposed DQN + APF in comparison with DQN average rewards in which the positive goal is attained by 85% and negative goal is attained by −90%

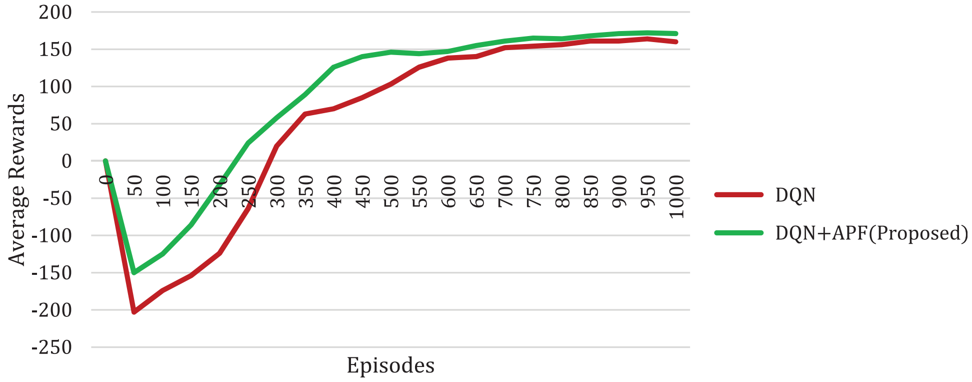


Figure 11: Comparison of algorithms concerning average rewards and episodes

4 Conclusion

In this paper, the Artificial Potential Field is to increase the average reward and sample effectiveness, integrate it with the DQN algorithms. In addition, the suggested method minimizes path planning time, increases the number of effective objectives throughout training, reduces convergence time and improves the robotic arm TB3's seamless and effective mobility characteristics. In a ROS-Gazebo simulator, this study shows that the proposed algorithm can navigate a TB3 Waffle Pi to specified places. In this study, the proposed approach is limited to a static setting. The suggested methodology has yet to investigate dynamic environmental changes. By comparing the proposed DQN + APF with DQN in terms of a number of successful targets, the performance improvement rate is 88%. In terms of average time when compared to DQN, the proposed DQN + APF attains an average time of 0.331 s. The performance of the proposed DQN + APF in comparison with DQN average rewards in which the positive goal is attained by 85% and the negative goal is attained by −90%

Acknowledgement: The authors with a deep sense of gratitude would thank the supervisor for his guidance and constant support rendered during this research.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Muthukumaran and R. Sivaramakrishnan, "Optimization of mobile robot navigation using hybrid dragonfly-cuckoo search algorithm," *Tierärztliche Praxis*, vol. 40, pp. 1324–1332, 2020.

[2] P. Wang, X. Li, C. Song and S. Zhai, "Research on dynamic path planning of wheeled robot based on deep reinforcement learning on the sloping ground," *Journal of Robotics*, vol. 2020, pp. 1–20, 2020.

[3] M. Quigley, B. Gerkey and W. D. Smart, "Programming Robots with ROS," in *A Practical Introduction to the Robot Operating System*, 1st ed., Sebastopol, CA: O'Reilly Media, Inc., pp. 1–417, 2015.

[4] S. Dilip Raje, "Evaluation of ROS and gazebo simulation environment using turtlebot3 robot," in *Proc. 11th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, Setúbal, Portugal, pp. 1–5, 2020.

[5] L. Tai and M. Liu, "Towards cognitive exploration through deep reinforcement learning for mobile robots," arXiv preprint arXiv:1610.01733, 2016.

[6] J. Xin, H. Zhao, D. Liu and M. Li, "Application of deep reinforcement learning in mobile robot path planning," in *Proc. 2017 Chinese Automation Congress (CAC)*, Jinan, China, pp. 7112–7116, 2017.

[7] Z. Peng, J. Lin, D. Cui, Q. Li and J. He, "A multiobjective trade-off framework for cloud resource scheduling based on the deep Q-network algorithm," *Cluster Computing*, vol. 23, no. 4, pp. 2753–2767, 2020.

[8] M. M. Rahman, S. H. Rashid and M. M. Hossain, "Implementation of Q learning and deep Q network for controlling a self balancing robot model," *Robotics and Biomimetics*, vol. 5, no. 1, pp. 1–6, 2018.

[9] Y. Yang, L. Juntao and P. Lingling, "Multi-robot path planning based on a deep reinforcement learning DQN algorithm," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 177–183, 2020.

[10] H. Bae, G. Kim, J. Kim, D. Qian and S. Lee, "Multi-robot path planning method using reinforcement learning," *Applied Sciences*, vol. 9, no. 15, pp. 3057, 2019.

[11] 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, no. 63, pp. 1–12, 2020.

[12] M. A. Ali and M. Mailah, "Path planning and control of mobile robot in road environments using sensor fusion and active force control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2176–2195, 2019.

[13] Y. Sun, J. Cheng, G. Zhang and H. Xu, "Mapless motion planning system for an autonomous underwater vehicle using policy gradient-based deep reinforcement learning," *Journal of Intelligent & Robotic Systems*, vol. 96, no. 3–4, pp. 591–601, 2019.

[14] S. F. Lin and J. W. Chang, "Adaptive group organization cooperative evolutionary algorithm for TSK-type neural fuzzy networks design," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 3, pp. 1–9, 2013.

[15] J. Xin, H. Zhao, D. Liu and M. Li, "Application of deep reinforcement learning in mobile robot path planning," in *Proc. 2017 Chinese Automation Congress (CAC)*, Jinan, China, pp. 7112–7116, 2017.

[16] M. Liu, F. Colas, L. Oth and R. Siegwart, "Incremental topological segmentation for semi-structured environments using discretized GVG," *Autonomous Robots*, vol. 38, no. 2, pp. 143–160, 2015.

[17] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002.

[18] N. Sariff and N. Buniyamin, “An overview of autonomous mobile robot path planning algorithms,” in *Proc. 2006 4th Student Conf. on Research and Development*, Shah Alam, Malaysia, pp. 183–188, 2006.

[19] W. Sun, G. Dai, X. Zhang, X. He and X. Chen, “TBE-net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–13, 2021.

[20] Z. Jiangzhou, Z. Yingying, W. Shuai and L. Zhenxiao, “Research on real-time object detection algorithm in traffic monitoring scene,” in *Proc. 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, Shenyang, China, pp. 513–519, 2021.

[21] L. Tai and M. Liu, “A robot exploration strategy based on q-learning network,” in *Proc. 2016 IEEE Int. Conf. on Real-Time Computing and Robotics (RCAR)*, Angkor Wat, Cambodia, pp. 57–62, 2016.

[22] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare and J. Pineau, “An introduction to deep reinforcement learning,” *Foundations and Trends in Machine Learning*, vol. 11, no. 3–4, pp. 219–354, 2018.

[23] K. M. Jung and K. B. Sim, “Path planning for autonomous mobile robot using potential field,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 9, no. 4, pp. 315–320, 2009.

[24] J. Sfeir, M. Saad and H. Saliah-Hassane, “An improved artificial potential field approach to real-time mobile robot path planning in an unknown environment,” in *Proc. 2011 IEEE Int. Symp. on Robotic and Sensors Environments (ROSE)*, Montreal, QC, Canada, pp. 208–213, 2011.

[25] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Proc. Autonomous Robot Vehicles*, New York, NY, pp. 396–404, 1986.

[26] H. E. Romeijn and R. L. Smith, “Simulated annealing for constrained global optimization,” *Journal of Global Optimization*, vol. 5, no. 2, pp. 101–126, 1994.

[27] R. Dhaya, R. Kanthavel and A. Ahilan, “Developing an energy-efficient ubiquitous agriculture mobile sensor network-based threshold built-in MAC routing protocol, ” *Soft Computing*, vol. 25, no. 18, pp. 12333–12342, 2021.

[28] A. Ahilan, G. Manogaran, C. Raja, S. Kadry, S. N. Kumar *et al.,* “Segmentation by fractional order darwinian particle swarm optimization based multilevel thresholding and improved lossless prediction-based compression algorithm for medical images,” *IEEE Access*, vol. 7, pp. 89570–89580, 2019.

[29] B. Sivasankari, A. Ahilan, R. Jothin and A. J. G. Malar, “Reliable N sleep shuffled phase damping design for ground bouncing noise mitigation,” *Microelectronics Reliability*, vol. 88, pp. 1316–1321, 2018.

[30] M. Liu, F. Colas, F. Pomerleau and R. Siegwart, “A markov semi-supervised clustering approach and its application in topological map extraction,” in *Proc. 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, pp. 4743–4748, 2012.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness *et al.,* “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[32] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long *et al.,* “Caffe: Convolutional architecture for fast feature embedding,” in *Proc. 22nd ACM Int. Conf. on Multimedia*, Lisboa, Portugal, pp. 675–678, 2014.

[33] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre *et al.,* “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.