Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 1789–1794
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.27409.3590

Robot control interaction with cloud-assisted analysis control

Alaa Adeb Abdulraheem*, Aqeel Abdulazeez Mohammed

Department of Electronics and Communications, College of Engineering, University of Baghdad, Baghdad, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

Path planning with avoiding obstacles autonomously with a large of computing capabilities in an unknown dynamic environment is a difficult challenge for a mobile robot to solve. This research solves this challenge by combining deep Q-network (DQN) with cloud computing. To begin, a DQN is created and trained to predict the state-action value function of a mobile robot. The information collected from the original RGB image (pixels in the image) taken from the surrounding is fed into the DQN using a cloud computing platform, which reduces the algorithms high computation complexity; Finally, the action chosen policy picks the current optimal mobile robot action. To validate the DQN algorithm, we trained the robot in a dynamic environment with a simple and complex case. The simulation results show that, in a simple case of the environment, the DQN technique can converge to explore a path with fewer steps and higher average reward than in a complicated case and find a collision-free path with an accuracy rate of 89% in the simple case and when the environment becomes more complex, the accuracy rate is 70%.

Keywords: cloud services, deep Q-learning, Autonomous Navigation of the robot, Obstacle avoidance
2020 MSC: 68T07

1 Introduction

Intelligent robots are becoming more common in our daily lives. Intelligent robots can be used in several scenarios [17], such as space explorations, hazardous environments, military [8], surveillance, inspection, data gathering [6], agriculture], education, medicine [10], etc. The most basic goals of Intelligent robots are to avoid obstacles and find paths in unidentified environments [7]. Furthermore, these robotic applications necessitate high-performance computational skills to achieve an acceptable level of collision-free path planning [9], and, due to the limited computational capabilities of the robot [19], so cloud computing can be used to increase computational power and storage space for robots [23], which needed to achieve their allocated tasks [18]. the robot may be attempting to discover a collision-free path from an initial location to an end location with utilizing high computation power by use of a Deep Q-learning algorithm according to cloud computing services which analyzed data of sensor (e.g., Camera) [11], that input to the network which uses this information to learn the optimal action to do at each state via trial and error to maximize reward [20, 15]. It can develop a policy for working in a complex environment by assessing each state and action pair [16].

*Corresponding author
Email addresses: a.rahmeem1020@coeng.uobaghdad.edu.iq (Alaa Adeb Abdulraheem), akeel.a@coeng.uobaghdad.edu.iq (Aqeel Abdulazeez Mohammed)

Received: February 2022 *Accepted:* April 2022

2 Related works

Much algorithm-based navigation like DRL and others have successfully solved robot control problems by using raw observation information. Many works discussed these algorithms for finding the path in free space with avoided obstacles. In [3] uses a stereo camera to create a grid-edge-depth map that signifies the position and distance of obstacles in the environment, then the robot makes decisions to safe navigation. In [1] a control approach for a mobile robots navigation system based on the combination of type-2 fuzzy reasoning optimized by wavelet networks, with no obstacle in the surrounding. In [12] proposed A* with hybrid POS-ANN algorithm to achieve safe navigation for the robot in different environments. In [8] suggested a cloud-based deep Q-network path planning system that calculates and forwards the shortest path for the robot in a static environment. In [24] trained the robot for autonomous navigation in a dynamic environment using the DQN algorithm, which directly outputs the optimal actions using the current observation with a camera sensor. In [9], presented a DRL technique to drive a robot in an indoor area autonomously depending on the sensor evaluated by cloud services. In [18] propose DQN with the ultrasonic and RGB sensors to allow autonomous navigation for robots that communicated with each other in the factory. The paper [14] improve the control of quadrotor flying autonomously overcrowded simulation environments with a monocular camera by using the DRL method. In [4, 2] employ DRL and cloud services to train robots to build the desired safe path in a static environment. In [21] a deep reinforcement learning is used in a web-based instructional tool to teach mobile robot navigation in static surroundings depending on input data of 2D-LiDAR. In paper [25] figures a multi-robot path-planning model using the deep Q-network technique to avoid collisions with static and dynamic obstacles and reach the target point by traveling the shortest distance in the shortest time in the indoor environment. In [22] use DQN algorithm to solve the multi-robot cooperative problem, each robot with the information from the image can reach the target and avoidance of static and dynamic obstacles. In [13] steered robot to the fixed target in a simple environment, uses the DQN, which maps the camera images to output robot action decisions. In [5] presented a path planning method based on deep q-learning and the notion of rolling wave planning is used to move the robot safely in the unknown environment.

These past efforts have produced highly effective results for the scenarios proposed, with the majority of works relying on the raw image observations to learn agent decision-making policies without using cloud services for image processing approaches, and all of these studies working in environments with a static target. In this paper, image processing approaches are coupled with cloud services to navigate autonomously in a dynamic environment using Deep Q-learning with image data as input.

3 Proposed method and technology

3.1 System Architecture

Figure 1. shows a simple diagram of the system. The mobile robot after taking images, transfers them to the cloud computing platform to make image processing through it and then feedbacks parameters that are input to the network. in this network we use the DQN algorithm, which improves the agent's efficiency based on its knowledge of the surroundings, allowing it to navigate autonomously to it is target place and choose a collision-free path.

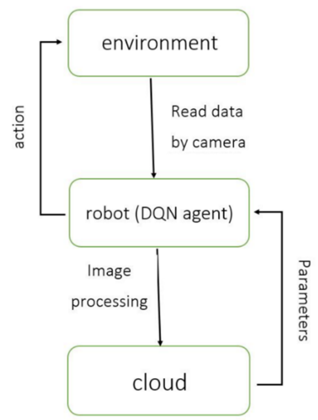


Figure 1: Block diagram for the system

3.2 Algorithm for deep Q-learning

In the case of the classic Q-learning, in a series of activities the agent interacts with its surrounding. each time instantaneous t action a_t, the agent takes note of the current state (frame of camera) s_t. The agent is then rewarded for its action an given the next state (camera frames) s_{t+1}. To maximize the cumulative reward r_t ,agent learn in his surroundings. The Q-value is a term that refers to the action-value of state s and action a, to update the Q-values is used Bellman equation according to eq. (3.1) [15].

$$Q(s_t, a_t) = r_t + \gamma \max Q(s_{(t+1)}, a_{t+1}) \qquad \gamma \in [0, 1] \tag{3.1}$$

The states are the training data that are used as input and the Q values are used as the target output. When learned efficiently of the planning, it ensures that in a particular state s_t, choosing an action is $a_t = \max Q(s_t, a)$ as a result, the future discounted reward R_t will be maximizing,where

$R_t = \sum_{i=t}^{T} \gamma^{i-t} r_i \qquad , i = \{t+1, t+2 \ldots\}$ t is denoted a time step

Using this function of reward, the agent will rapidly and effectively develop an optimum policy.

In Deep Q-learning (DQN), learning a neural network is used to map states to values $Q(s, a)$ and therefore needs many training iterations before it can meet. So, there's the agent captures and stores experience data that are produced during its movement in a form as (s_t, a_t, r_t, s_{t+1}) in replay memory D. This parameter guarantees the number of training samples. When replay memory reaches to 1000 experiences, the robot selects a mini-batch of experience data at random and it uses for the neural network's training, in addition the collision avoidance learning from photo is difficult and necessitates the use of more depth neural networks. To train these deep neural networks, they require a high number of computations for data and large memory for storage, so we use cloud application services to solve this problem. Fig. 2. illustrates the flow chat for DQN algorithm used in our test.

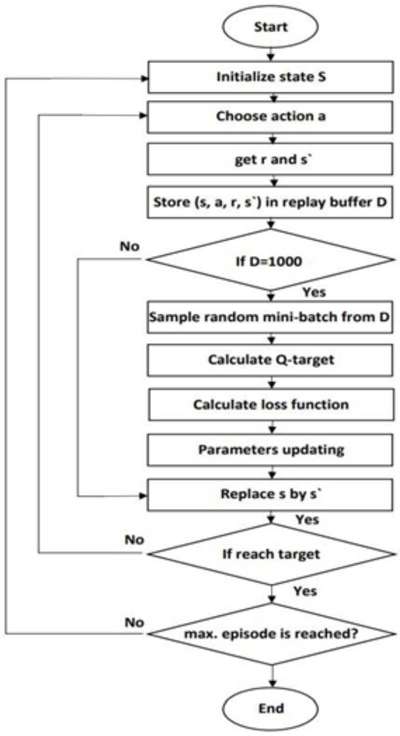


Figure 2: flow chat for Deep Q-learning algorithm

4 Experiment and results

To demonstrate the robot's ability of the autonomous navigation, we used TensorFlow deep learning technologies and the Python programming language to run the simulation experiment and train the robot in a dynamic environment with four different cases, as seen in Figure 3, which is 10 x 10cm in size. The mobile robot contains a camera sensor that can collect RGB images simultaneously, which are analyzed using cloud computing. The robot then receives the information from the cloud and feeds it to a neural network with one input layer, one hidden RELU layer (256 neurons), and one dense linear output layer (9 neurons), then chooses the best action for each iteration with a discount factor equal to 0.99, experience replay memory 50,000, and minibatch equal to 64. For a total of 20000 iterations, this process is repeated. until the robot discovers the most efficient path to the destination without colliding with any obstacles.

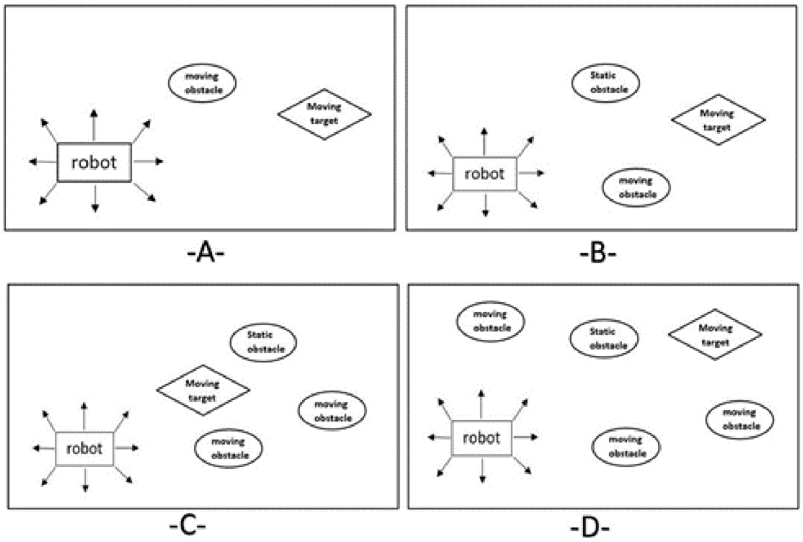


Figure 3: Simulation environment are shown for different four cases: (A) one dynamic obstacle (B) one static obstacle and one dynamic (C) one static obstacle and tow dynamic (D) one static obstacle and three dynamic.

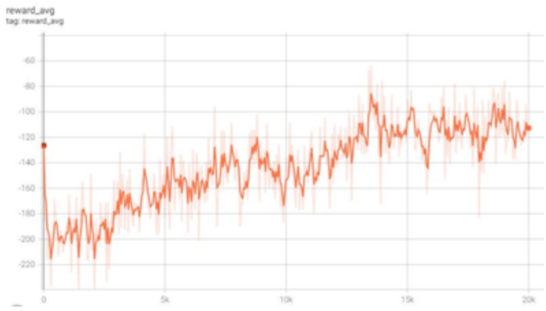
Through the agent training process, the agent gradually learns to execute an action that will raise the reward. Figure 4 shows the curves of cumulative reward change in the training process for four cases. It can be seen that the average cumulative reward are increasing clearly in the first 5,000 episodes during the training process, After 15,000 episodes of training, the reward value is small increasing and average cumulative reward is became -112(in case a),and it is equal to -78, -60, -180 for case b, c, d respectively, that means the agent can navigate around obstacles in changing environment and quickly arrive at the desired location to receive a continuing reward until the current training episodes are completed, but the variability and the complexity of the dynamic environment increases the difficulty of path planning and the robot will prevent from gaining higher rewards.

The accuracy rate of the suggested algorithm is calculated in addition to the average reward. The accuracy is a measure how well model predict corrects right behaviors.

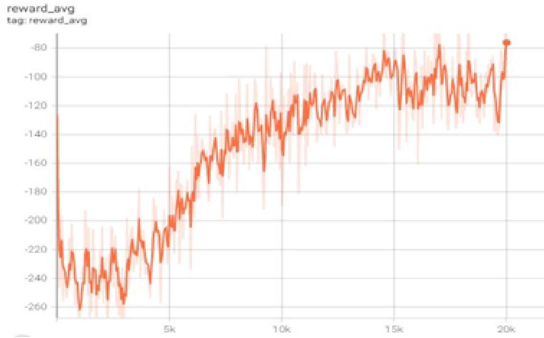
Results of comparison the accuracy rate which achieved by the DQN model for four cases are described in fig 5, In the first case, the accuracy rate is about 89%, in this case the proposed algorithm achieved better accuracy rate than other, while when the number of obstacles is increase to 2, 3 and 4, the accuracy rate is decrease to 78%, 73%, and 70% respectively. With result for this study, concluding we can provide the robot great capabilities of autonomous motion in complex environments.

5 Conclusion

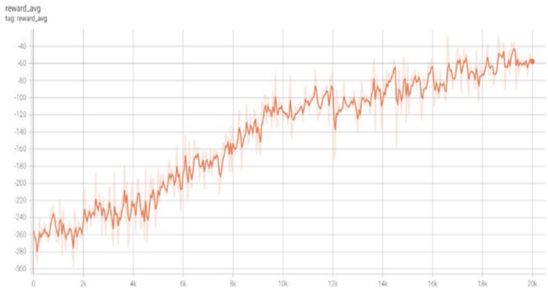
We present a deep reinforcement learning technique to automatically navigate a robot in the changing and uncertain environment, and avoid collisions with obstacles meanwhile. that happen by using observations from images as input only, which extraction from images by depending on the cloud application service. This algorithm has been trained in four cases of simulation environments that contain different numbers of static and dynamic obstacles. The DQN algorithm's effectiveness was assessed by measuring rewards over time in four different environments. According to the findings, the robot learned a policy to maximize rewards over time. We also evaluated the network's predict accuracy rate using this technique. The network successfully predicted the right action, and the robot learned the proposed algorithms policy.in addition, the findings have proven the usefulness of the proposed method. In the future, we'll



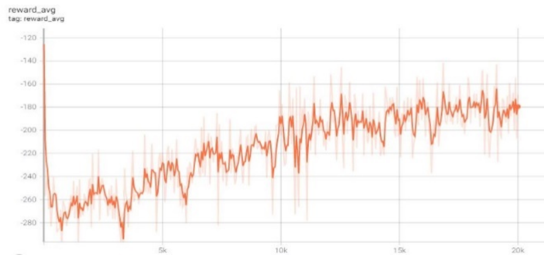
a-average reward for an agent with avoiding one obstacle(dynamic)



b-average reward for an agent with avoiding two obstacles(one dynamic and one static)



c-average reward for an agent with avoiding three obstacles (one static and two dynamics)



d-average reward for agent with avoiding four obstacles (one static and three dynamic)

Figure 4: average cumulative reward curves.

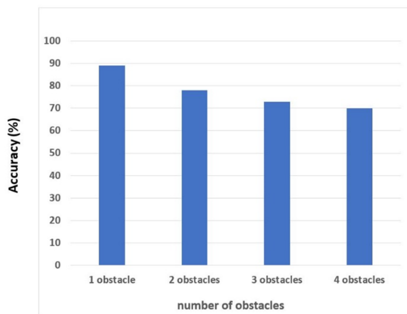


Figure 5: show the comparison results of accuracy rate

extend the experiment with neural network designs that are more complicated which can make the experiment results even better, also we'll test our technique on a robot in a real -world setting.

References

[1] F. Affane, K. Meguenni, and A. Omari, *Type-2 fuzzy logic controller optimized by wavelet networks for mobile robot navigation*, Indones. J. Electr. Eng. Comput. Sci. **18** (2020), no. 1, 326–334.

[2] A. Anwar and A. Raychowdhury, *Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning*, IEEE Access **8** (2020), 26549–26560.

[3] S. Canbaz and G. Erdemir, *Performance analysis of real-time and general-purpose operating systems for path planning of the multi-robot systems*, Int. J. Electr. Comput. Eng. **12** (2022), no. 1, 285.

[4] C. Chen, J. Jiang, N. Lv, and S. Li, *An intelligent path planning scheme of autonomous vehicles platoon using deep reinforcement learning on network edge*, IEEE Access **8** (2020), 99059–99069.

[5] M. Etemad, N. Zare, M. Sarvmaili, A. Soares, B. Brandoli Machado, and S. Matwin, *Using deep reinforcement*

learning methods for autonomous vessels in 2d environments, Canad. Conf. Artif. Intell., Springer, 2020, pp. 220–231.

[6] H. Guo, X. Wu, and N. Li, *Action extraction in continuous unconstrained video for cloud-based intelligent service robot*, IEEE Access **6** (2018), 33460–33471.

[7] L. Jiang, H. Huang, and Z. Ding, *Path planning for intelligent robots based on deep q-learning with experience replay and heuristic knowledge*, IEEE/CAA J. Autom. Sin. **7** (2019), no. 4, 1179–1189.

[8] K. Kamali, I. Bonev, and C. Desrosiers, *Real-time motion planning for robotic teleoperation using dynamic-goal deep reinforcement learning*, 17th Conf. Comput. Robot Vision (CRV), 2020, pp. 182–189.

[9] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, *A survey of research on cloud robotics and automation*, IEEE Trans. Automa. Sci. Eng. **12** (2015), no. 2, 398–409.

[10] J. Lee, A. Ab Ghafar, N. Nordin, F. Saparudin, and N. Katiran, *Autonomous multi-function floor cleaning robot with zig zag algorithm*, Indonesian J. Electr. Eng. Comput. Sci. **15** (2019), no. 2, 1653–1663.

[11] M. Lemos, A. de Souza, R. de Lira, C. de Freitas, and V. da Silva, *Robot navigation through the deep q-learning algorithm*, Int. J. Adv. Eng. Res. Sci. **8** (2021), 2.

[12] A. Mohammed, I. Aris, M. Hassan, and N. Kamsani, *New algorithm for autonomous dynamic path planning in real-time intelligent robot car*, J. Comput. Theor. Nanosci. **14** (2017), no. 11, 5499–5507.

[13] S. Nilwong and G. Capi, *Outdoor robot navigation system using game-based dqn and augmented reality*, 2020 17th International Conference on Ubiquitous Robots (UR), 2020, pp. 74–80.

[14] J. Ou, X. Guo, M. Zhu, and W. Lou, *Autonomous quadrotor obstacle avoidance based on dueling double deep recurrent q-learning with monocular vision*, Neurocomput. **441** (2021), 300–310.

[15] N. Pandey and M. Diwakar, *A review on cloud based image processing services*, 7th Int. Conf. Comput. Sustain. Glob. Dev. (INDIACom), IEEE, 2020, pp. 108–112.

[16] R. Patel, *Human robot interaction with cloud assisted voice control and vision system*, Ph.D. thesis, The University of Texas at Arlington, 2018.

[17] M. Penmetcha and B.-C. Min, *A deep reinforcement learning-based dynamic computational offloading method for cloud robotics*, IEEE Access **9** (2021), 60265–60279.

[18] M. Penmetcha, S. S. Kannan, and B.-C. Min, *Smart cloud: Scalable cloud robotic architecture for web-powered multi-robot applications*, IEEE Int. Conf. Syst. Man, Cybernet.(SMC), 2020, pp. 2397–2402.

[19] M. Saravanan, P. Kumar, and A. Sharma, *Iot enabled indoor autonomous mobile robot using cnn and q-learning*, IEEE Int. Conf. Ind. 4.0, Artif. Intell. Commun. Technol.(IAICT), IEEE, 2019, pp. 7–13.

[20] K. Sharma and R. Doriya, *Path planning for robots: An elucidating draft*, Int. J. Intell. Robot. Appl. **4** (2020), no. 3, 294–307.

[21] R. Suenaga and K. Morioka, *Development of a web-based education system for deep reinforcement learning-based autonomous mobile robot navigation in real world*, 2020 IEEE/SICE International Symposium on System Integration (SII), 2020, pp. 1040–1045.

[22] D. Wang, H. Deng, and Z. Pan, *Mrcdrl: Multi-robot coordination with deep reinforcement learning*, Neurocomput. **406** (2020), 68–76.

[23] Z. Wang, S. Yang, X. Xiang, A. Vasilijević, N. Mišković, and b. Na\dj, *Cloud-based mission control of usv fleet: Architecture, implementation and experiments*, Control Eng. Pract. **106** (2021), 104657.

[24] K. Wu, H. Wang, M. Esfahani, and S. Yuan, *Learn to navigate autonomously through deep reinforcement learning*, IEEE Trans. Ind. Electron. **69** (2022), no. 5, 5342–5352.

[25] Y. Yang, L. Juntao, and P. Lingling, *Multi-robot path planning based on a deep reinforcement learning dqn algorithm*, CAAI Trans. Intell. Technol. **5** (2020), no. 3, 177–183.