

Robot Navigation using Reinforcement Learning

M. Kavitha^a, R. Srinivasan^b, Bahdramraju Vishnusai^c and Y. Sri Surya Ganesh^d

^a Faculty, Computer Science & Engineering, Vel Tech Rangarajan Dr.Sagunthala, R&D Institute of Science and Technology, India. kavitha@veltech.edu.in

^b Faculty, Computer Science & Engineering, Vel Tech Rangarajan Dr.Sagunthala, R&D Institute of Science and Technology, India. rsrinivasan@veltech.edu.in

^c Student, Computer Science & Engineering, Vel Tech Rangarajan Dr.Sagunthala, R&D Institute of Science and Technology, India. naiduvisnu1@gmail.com

^d Student, Computer Science & Engineering, Vel Tech Rangarajan Dr.Sagunthala, R&D Institute of Science and Technology, India. suryaganesh227@gmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Learning to navigate in an unknown environment is a crucial capability of mobile robot. Conventional method for robot navigation consists of three steps, involving localization, map building and path planning. However, most of the conventional navigation methods rely on obstacle map, and do not have the ability of autonomous learning. In contrast to the traditional approach, we propose an end-to end approach which uses deep reinforcement learning for the navigation of mobile robots in an unknown environment. The model is trained with deep reinforcement techniques using Q-Learning algorithm. Through Q-Learning algorithm, mobile robot can learn the environment gradually through its wonder and learn to navigate to the target destination. The experimental result shows that mobile robot can reach to the desired targets without colliding with any obstacles. In future the same can be enhanced to implement the object classifying methods to understand traffic signal by the robot in order to travel in roads. Further it can be extended to create a self-driving car. Cascade classifier is used for classifying the data.

Keywords: Robot, Navigation, Path Building, Target, Destination, Traffic, Classification, Q-learning, Localization.

1. Introduction

Nowadays Artificial Intelligence is emerging in various domains like agriculture, robotics, automobile etc. This technology leads us to the future. This can resolve various problems based on experience gained through continuous learning. The era of self-driving started in the 1960s and researches were begin in the 1970s [3]. Currently, the self-driving car was capable of detecting a simple line, which is the initial stage. Recently self-driving cars are developed using the Artificial Intelligence (AI), which is still under research. Deep learning techniques can be used to recognize the objects and helps the car to understand the new environment. However, it takes long time and cost to retrieve the real time data and information. Developing virtual environmental self driving cars are also under research. The experimental results of above said techniques help to choose the appropriate algorithm and method to implement in the real time environment. Image processing is playing a key role in autonomous cars to detect objects and also for decision making. Another aspect is the Robots which can be used in daily life like cleaning, rescuing, and mining. Robot navigation has to be done to learn the unknown environment. Reinforcement Learning shall be applied to make the robot autonomous. However, enabling robots to navigate autonomous in a real world is still a challenging task. Traditional navigation method consists of localization, map building, and path planning. Deep learning [7] shows a great performance in feature representations and in realizing a human level intelligence. This paper proposes a self navigation robot to identify its own path without human effort by using reinforcement learning. Applying image processing is to recognize the traffic signals and traffic signs. The scope of this paper is to reduce traffic deaths and eliminates the stop-and-go ways by 100% and also it reduces the travelling time.

2. Related Work

Juntae kim et al, explains how deep learning plays a key role in Artificial Intelligence [7]. The developers should spend a lot of time and money to learn how to drive with image data and steering wheel control data for self-driving cars. Collection of data is the tedious process for different kinds of vehicles like busses, trucks, vans and so on. The authors proposed a Novel method to apply a virtual game environment data as training data for the self-driving cars. In the first stage, the effectiveness is verified by applying the training data to control a vehicle on a driving the activities happening with in the nodes of a blockchain network game. In the next stage, data set is evaluated in the real world vehicle driving. [5] Deep Reinforcement Learning can be used to navigate the robot using sensor. Meera Gandhi.G et al, [2] concluded that a huge number of cars can learn collectively from the experience of one single car, the implementation of training algorithm is decreased even if the cars are more. The complexity of the procedure reduces from n to 1, where n can be as big as it can. So training each car is eliminated.

By enabling thousands of cars to interact and share their experience they can learn faster and easier. Saving time, money and resources is the main target of this method. Human efforts are also reduced. The system talks about the integration of AI and Block chain technology. Blockchain technology gives a concept of the public ledger where all are recorded in the form of log entries. akafumi Okuyama et al, [9] proposed an algorithm named Deep reinforcement learning. They successfully learned the policies from the images and utilized Deep q networks (DQN). This algorithm estimates the problems and improves the performance of the robot by interacting with the surrounding environment in order to make decision. At each step, the robot chooses an action according to the current observation. They evaluated the model in two environments, one in simple and another in complex. In the beginning, the robot crashed the obstacles frequently and could not reach the destination. After many attempts finally it acquires the surrounding knowledge by interacting with the environment. Eventually the robot could navigate to the destination quickly and autonomously in both simple and complex environment without any collisions. They tested this in the normal environment; in future the work can be extended in a dynamic environment and transferred to the real world. [11] Data mining algorithms shall be used to mine the path of the robot. [6] J. F. Khan et al, proposed automatic traffic signal and sign detection which is an important part of the autonomous driving system. They proposed two stages, detection using a novel application of maximum stable extremal regions (MSERs) and recognition with a histogram of oriented gradient (HOG) features which are classified using a linear support vector machine(SVM). MSERs are used to detect the traffic signals and signs. A sober filter is used to find the horizontal and vertical derivations, traffic signals are recognized by using a HOG application[4]. The input image is read and initially it identifies Initially input image is read and white MESRs are found, then color MSERS are found. Finally the shape and sign is classified and merged with the previous result. LSRB and RSLB algorithm is used to create robot[12]. The purpose of the robot is to find the shortest path but due to the size the robot was not dynamic and fast maze solver. AI based line following robot uses right hand rule to follow the line based on the destination. Robot can also be controlled using LiDAR and Deep Nerval Network [10]. The experimental result showed 70% to 80% accuracy. Dijkstra algorithm can be used to find the shortest path but it took long time for execution. [9] Deep Q learning can be used to develop self autonomous driving car.

3. Robot Navigation

Proposed Model

The real-time hardware model is developed to find the destination path in a real time environment. Reinforcement Learning is used to learn the new environment and Image processing is used to recognize the signals and images. Raspberry pi is used to control the sensors and wheels of the robot. Fig. [1] describes that obstacles in environment path will be detected by ultrasonic sensors, Traffic and road signals will be detected by pi camera. The data collected will be sent to raspberry pi which analyzes and identifies the path makes decision for path detection. Q learning algorithm is used to analyze data and cascading classifier makes the robot to follow the road signals. This model can be used as a solution for an autonomous car.

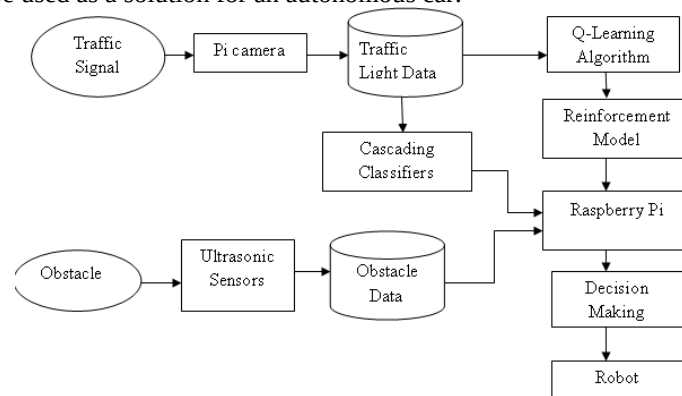


Fig. 1. System design

Obstacle Detection

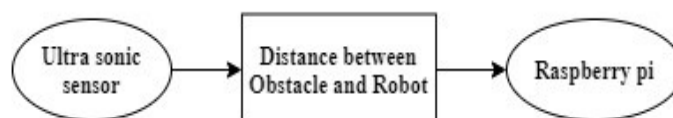


Fig. 2. Obstacle detection

Fig. [2] shows that Ultrasonic sensor senses the distance between the obstacle and robot. It calculates the distance and takes decision in which direction the robot has to move.

Traffic Sign Detection

**Fig. 3.** Traffic sign detection

Fig. [3], shows that Pi camera captures the video of the path and transmits to system. Cascading classifier is using to process the image and to detect the type of traffic sign, which helps the robot to follow the traffic sign.

Path Creation

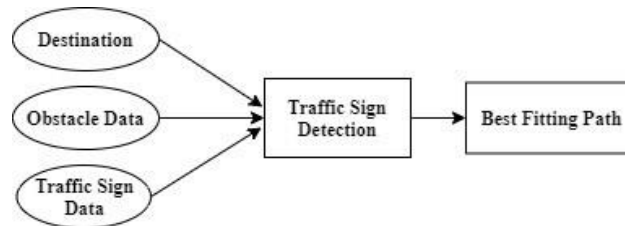
**Fig. 4.** Path creation

Fig. [4] shows the identification of best fitting path. Q-Learning Algorithm is used to analyze the data and to create the best fitting path to reach the destination.

Q-learning Algorithm

Q-learning algorithm is a model-free and value based reinforcement learning algorithm. The Q in the Q-learning stands for quality. $Q^*(s, a)$ is the expected value (cumulative discounted reward) in state s and then following the optimal policy. The (1) Q-function uses the Bellman equation and takes two inputs: state (s) and action (a).

$$Q^\pi(s_t, a_t) = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s_t, a_t] \quad (1)$$

$Q^\pi(s_t, a_t)$ stands Q-values for a particular state. $E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots]$ stands for expected discounted cumulative reward and s_t, a_t denotes the state and action. Q learning algorithm works in five steps that mentioned below.

Step 1: Initialize the Q-table

Initially Q-table has to be built. There are n columns, where n = number of actions and m rows, where m = number of states. In this example n =Go Left, Go Right, Go Up and Go Down and m = Start, Idle, Correct Path, Wrong Path and End. Q-Table is initialized with the values of 0 as shown in Table [1].

Table 1. Initial Q-table

Action	Left	Right	Up	Down
Start	0	0	0	0
Idle	0	0	0	0
Correct Path	0	0	0	0
Wrong Path	0	0	0	0
End	0	0	0	0

Table 2. Updated Q-table

Action	Left	Right	Up	Down
Start	0	1	0	0
Idle	0	0	0	0
Correct Path	0	0	0	0
Wrong Path	0	0	0	0
End	0	0	0	0

Step 2: Identify and Action

Step 3: Do the Process and action

The steps 2 and 3 is executed for an undefined amount of time until the training is stopped. Based on the Q table, an action 'a' in the state 's'; is chosen. All the Q values will be '0', when all episodes starts initially. Bellman equation is used to update the Q-values to move right. Epsilon greedy strategy is also used, initially epsilon rate is higher. Random action is chosen by exploring the environment because the agent doesnot know about the

environment at the initial stage, later on epsilon rate decreases. Due that agent becomes more confident in finding the Q-values. Updation is done using Bellman equation as shown in the Table [2].

Step 4: Measure Reward

Reward will assign based on the performance of the object.

Step 5: Evaluate

Till the end of the learning process, the above steps are repeated. Thus the Q-table is updated and the value of Q is increased. The Q(state, action) returns the future reward which is expected at that state.

$$NewQ(s, a) = Q(s, a) + \alpha[R(s, a) + \gamma \max_{a'} Q'(s', a') - Q(s, a)] \quad (2)$$

2) $NewQ(s, a)$ defines new Q-value for the state and action, $Q(s, a)$ stands current Q-values, α stands learning rate, $R(s, a)$ Reward for taking an action in state, γ stands Discount rate, $\max_{a'} Q'(s', a')$ maximum expected future reward

Q-Table is updated by exploring the agent's environment. The agent starts exploiting the environment once the Q-Table is ready. The final Q-table is shown in Table [3]

Table 3. Final Q-table

Action	Up	Down	Right	Left
Start	0	0	1	0
Idle	0	0	0	0
Correct Path	0	50	22	0
Wrong Path	15	0	18	0
End	0	0	1	0

4. Simulation And Results

Robot's Q learning algorithm is tested using various learning rate. The values are : 1, 0.75, 0.5 and 0.25. The results show that robot with the value 0.25 cannot learn to avoid obstacles. Sometimes robot with 0.5 learning rate succeed to learn. Robot with 0.75 and 1 learning rate can learn obstacle avoidance task well every time.

Result of Q-learning algorithm

The proposed system shows that it makes the robot to travel in an unknown environment to the desired destination with crossing obstacles. Fig. [5] describes a graphical representation of steps taken by the object in every stage. An object has taken 1000+ stages to finds the best fitting path. The X-axis indicates number of stages by the object. Y axis indicates that number of steps took by the object to reach its destination. Fig. [6] shows the graphical representation of the cost taken by the object in every stage. In this algorithm, cost represents the number of rewards taken by the object. If the object moves towards the correct direction then algorithm will reward with +1. So that the system trains each and every episode.

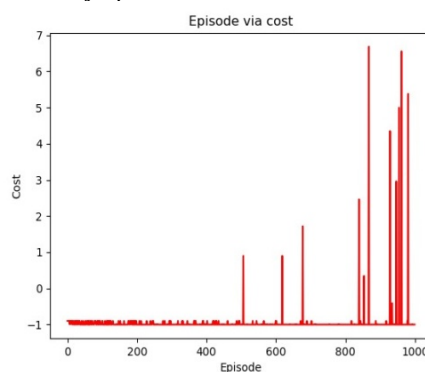


Fig. 5. Episode via cost

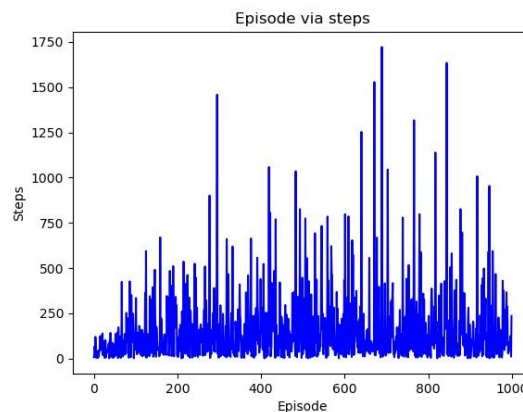


Fig. 6. Episode via steps

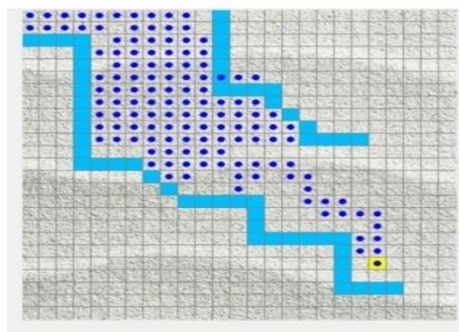


Fig. 7. Simulation result

Fig. [7] shows the simulation result, a virtual environment created using tkinter. Tkinter is a python module used for create user interface. Blue boxes indicates obstacles and Blue color circle shows the Travel history and yellow object indicates destination

5. Conclusion

The physical robot using q-learning algorithm can perform well for autonomous navigation task. Q learning mechanism can learn and understand the obstacle avoidance and successfully collects the positive rewards continually. Learning rate of Q learning mechanism affects the robot's learning performance. The experimental results show that if the learning rate is high then the learning phase is fast. In the future, the same shall be used to develop an autonomous car that can take the traveller to the destination in a real-time environment.

References

1. G. Oliveira, R. Silva, T. Lira, L.P. Reis, "Environment mapping using the lego mindstorms nxt and lejos nxj", 14th Portuguese Conference on Artificial Intelligence, EPIA, pp. 267-278, 2009.
2. G.M. Gandhi, "Artificial Intelligence Integrated Blockchain For Training Autonomous Cars", Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Vol. 1, pp. 157-161, 2019.
3. H. Wicaksono, "Q learning behavior on autonomous navigation of physical robot", 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 50-54, 2011.
4. J. Greenhalgh, M. Mirmehdi, "Real-time detection and recognition of road traffic signs", IEEE transactions on intelligent transportation systems, Vol. 13, No. 4, pp. 1498-1506, 2012.
5. S.H. Han, H.J. Choi, P. Benz, J. Loaiciga, "Sensor-Based Mobile Robot Navigation via Deep Reinforcement Learning", IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 147-154, 2018.
6. J.F. Khan, S.M. Bhuiyan, R.R. Adhami, "Image segmentation and shape analysis for road-sign detection", IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 1, pp. 83-96, 2010.
7. J. Kim, G. Lim, Y. Kim, B. Kim, C. Bae, "Deep learning algorithm using virtual environment data for self-driving car", International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), pp. 444-448, 2019.

8. L. Khriji, F. Touati, K. Benhmed, A. Al-Yahmedi, "Q-learning based mobile robot behaviors coordination", *International Renewable Energy Congress*, pp. 293-299, 2010.
9. T. Okuyama, T. Gonsalves, J. Upadhay, "Autonomous driving system based on deep q learning", *International Conference on Intelligent Autonomous Systems (ICoIAS)*, pp. 201-205, 2018.
10. J.K. Wang, X.Q. Ding, H. Xia, Y. Wang, L. Tang, R. Xiong, "A LiDAR based end to end controller for robot navigation using deep neural network", *IEEE International Conference on Unmanned Systems (ICUS)*, pp. 614-619, 2017.
11. Kavitha.M, Sugnathy.M, Srinivasan.R, Intelligent learning system using data mining-ILSDM, *International Journal of Innovative Technology and Exploring Engineering*, Volume 8, Issue 9, July 2019, Pages 505-508
12. Akib Islam , Farogh Ahmad , P.Sathya, Shortest Distance Maze Solving RoboT, *International Journal of Research in Engineering and Technology*, Volume: 05 Issue: 07, Jul-2016.