

Traffic Signal Control System Using Contour Approximation Deep Q-Learning^{*}

Dr. Ramya R S¹[0000–1111–2222–3333], K K Bharath², K Revanth Krishna³,
Kancham Jaswanth Reddy⁴, and M Sri Bhuvan⁵

Dayananda Sagar College of Engineering, Bangalore, India 560078

Abstract. A reliable transit system is essential and offers a lot of advantages. But traffic has always been an issue in major cities, and one of the main causes of congestion in these places is intersections. To reduce traffic, a reliable traffic control system must be in place. This research sheds light on how to consider dynamic traffic at intersections and minimise traffic congestion using an end-to-end deep reinforcement learning approach. The goal of the model is to reduce waiting times at these crossings by controlling traffic in various scenarios after receiving the necessary training.

Keywords: Deep Q-Learning · Artificial intelligence · Machine learning · Traffic signal control system.

1 Introduction

Over time, there has been a rise in the amount of people moving around, which has led to an increase in traffic. Numerous negative effects of increased traffic include longer travel times and higher pollution levels brought on by increased fuel consumption of cars. Currently, a traditional, three-phase, pre-timed signal system that is red, yellow, and green is employed. This signalling system works with a set duration based on the provided timings. This approach, however, does not account for the dynamic traffic that moves through each lane at intersections.

There is ongoing research being done to provide new approaches to the problem of traffic congestion. Numerous approaches are now being used to address this issue as a result of the technology sector's rapid growth. [1] [2] One such instance is the Internet of Things [3] [4] where traffic is efficiently controlled using cameras and other sensors. Wireless sensor networks (WSN) [5] and the use of embedded platforms [6] are two more effective techniques.

However the field of Artificial Intelligence and Machine Learning has given rise to many methods which are effective in controlling traffic. Reinforcement Learning (RL) is a practical method for controlling traffic signals [7]. Use of deep reinforcement learning [8], which introduces feature-based state representation and rewards idea, is one example among many. This technique makes the TSC scalable. The deep convolution neural network [9] used in this technique, which

^{*} Dayananda Sagar College of Engineering.

also uses deep reinforcement learning, collects parameters from the raw traffic data, such as the locations, speeds, and waiting times of the cars.

In this paper are present a traffic control system that takes images of traffic intersections with dynamically flowing vehicles as input and performs traffic signalling. The model's objective is to decrease the average waiting time at crossings while also identifying emergency vehicles and opening the lane in the which they are travelling and give them top priority. We run the model on different scenarios and compare with other proposed traffic signal control systems.

The contributions of this paper are as follows:

1. We propose an off-policy end-to-end deep reinforcement learning based traffic signal control system to control dynamic traffic efficiently at intersections with top priority for emergency vehicles.
2. The model proposed detects for emergency vehicles on any of the sides of the intersection and aims to reduce its waiting time by clearing the traffic in that lane.

2 Related Work

Several other machine learning models that have been used to propose a traffic signal control system are studied here.

A deep reinforcement learning model is created in [11] to control the traffic light cycle. By compiling the traffic patterns, the complex traffic situation in the United States has been assessed. Small grids are used to divide up the entire traffic intersection. In this model, states, rewards, and actions are all used.

Using a Gaussian convolutional neural network is an alternative strategy. A graph convolutional neural network is developed in [9]. Agents are established in a distributed manner to design a policy to run a traffic signal at an intersection using the recommended method as opposed to speaking to other agents directly. The suggested method can locate comparable policies using an FCNN twice as fast as the conventional RL-based method, and it is also better equipped to handle variations in traffic demand. In the context of six crossings, the outcomes of NFQI employing the GCNN, FCNN, and fixed-timing control were compared.

Chen et al. [12] utilised a five-layered convolution with a (2x2) filter size without a pooling layer. The unique method used by the authors (PCNN) involved simulating periodic traffic data using deep neural networks based on convolution. The study folded the time-series to create the input, which incorporated historical and real-time traffic data. They replicated the amount of congestion from the previous time slot in the matrix to illustrate the relationship between a new time slot and the recent past.

Convolution neural networks were suggested as a possible implementation of the traffic sign recognition method by Alexander et al. [13]. The paper also shows alternative CNN designs side-by-side for comparison. The TensorFlow library and CUDA's massively parallel architecture for multithreaded programming are used to train the neural network. On a mobile GPU, the full detection and

recognition procedure for traffic signs is carried out in real time. The author used TensorFlow, a deep learning framework, to tackle the identification of traffic signs issue. Training and testing were carried out on the GTSRB dataset.

Modern reinforcement learning (RL) techniques for online signal controller optimisation were put forth by Mousavi et al. [14]. In this way, the set of approaching vehicles (incoming lane, speed, waiting time, queue length), as well as the present signal assignment (enabled phase), frequently determine the status of the intersection. An RL agent must optimise a technique that links state mapping to signal (phase) assignment. Deep Q-learning techniques are frequently used for this type of learning with the goal of learning the projected future value from each action at a particular state. The controller is then told to take the best steps it can going forward.

Table 1 shows the comparison of different reinforcement learning methods.

3 Problem Definition

In this section, we define the problem statement for our traffic signal control system.

Consider a single crossroads where traffic flows in all four directions. Traffic signals are used to indicate whether or not vehicles in a lane are allowed to proceed or must halt. We consider STOP (indicated by a red light) and GO (indicated by a green light) as the two possible states for the signal in order to keep things simple. We adhere to left-hand traffic laws, but this has no impact on the model, which can be adjusted to operate under right-hand situation as well.

The transitions for the signal are indicated below: Following the same method mentioned in the above paragraph, four different states are possible.

1. Signal 1 is green and the rest are red
2. Signal 2 is green and the rest are red
3. Signal 3 is green and the rest are red
4. Signal 4 is green and the rest are red

The intersection can be modelled using the Markov decision process. It provides a mathematical framework for representing environments in reinforcement learning applications. [10] For our application, we define a 4-tuple MDP (Markov Decision Process) to represent the environment as follows:

$$\langle S_t, A_t, R_{t+1}, S_{t+1} \rangle \quad (1)$$

where S_t represents the current state, A_t represents the action taken by the agent, S_{t+1} is the state reached when action A_t is taken at state S_t , and R_{t+1} is the reward for taking action A_t at S_t and reaching S_{t+1} .

4 TRAFFIC SIGNAL CONTROL DESIGN

In this section, we introduce the different aspects of the proposed model such as the environment, reward, and the actions.

Table 1. Different reinforcement models

SI No.	YEAR	METHOD USED	WORKING	APPLICATIONS
1	2020	Model based Monte Carlo	Utilises sampling to estimate the Markov Decision Process dynamics. The model uses Monte Carlo to predict transitions directly from the data and samples a large number of (s, a, r, s') pairs.	Used in situations where there is a lot of data to process. There is a guarantee that the estimations will be quite accurate.
2	2018	Policy gradient based	The policy gradient-based method directly learns from the policy rather than modelling the value function.	Utilized in autonomous vehicle design where the state consists of convoluted situations provided to agents.
3	2018	Actor Critic	The actor critic technique estimates and updates both the policy and value functions.	There are many actor critic methods such as A2C that are used in autonomous driving.
4	2019	Q-Learning	It is a temporal difference method, an off-policy method where a Q-value is estimated to optimise the actions.	It is used extensively in training agents to play games and has been proven effective in being able to learn complex rules.
5	2017	Deep Reinforcement Learning	It employs neural networks that don't explicitly specify the characteristics; instead, they use the state as input directly.	It is utilized in tasks where the state is not easily representable, such as tasks where the screen itself acts as a state.
6	2019	Value function approximation	This approach learns by decreasing the loss between an actual Q value and an estimated Q value.	Useful in large or continuous state spaces. The ability to swiftly adapt to various sample sets is a major benefit.
7	2019	SARSA	Like Q-Learning, it is a temporal difference method that combines dynamic programming and monte carlo techniques.	Provides excellent results in games, robotics and mechatronics control.

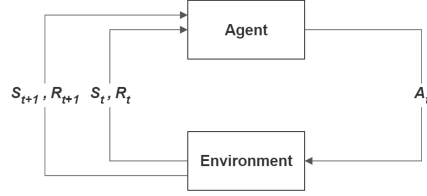


Fig. 1. Interaction between agent and environment

4.1 Environment

Fig.1 represents the environment where we conduct our experimentation. A surveillance camera placed at the crossroads allows us to record the status as seen in this context. Since this data is essential for instructing the RL agent, the camera should be set up so that it can adequately watch traffic from all 4 lanes.

4.2 Reward

Since the main goal of this experiment is to minimize the average waiting time at junctions, this parameter can act as a good reward. The average waiting time is calculated by summing up the waiting time (time between stopping at red light and starting at green light) of each vehicle in every lane and then dividing by the total number of vehicles.

Once we have got the average waiting time, we need to consider the negative of this number as reward. This is because the reward acts as a positive signal for our RL agent and it is trying to maximize the reward. While this approach is simple, it can also increase the complexity of computation as the average waiting time can become a very large number in case of heavy traffic.

4.3 Actions

The agent can take 4 given actions at any given time i.e make any one of the four signals green and the rest stay red. This is a simplified version of the actual way signals work in reality, as yellow lights are used to ask vehicles to slow down. But since the duration of yellow lights is anyway going to be constant, it is not something worthy enough to be learned by our agent. However, we can train our agent on these 4 actions and it will still be able to work in environments which use yellow lights because that depends on the design of the environment and does not affect the learning curve of the agent.

5 Algorithm

The algorithm of the Q-network is presented in algorithm 1. The algorithm works by first getting the state at any time t . It can then take one of two paths - one

path chooses to "explore" i.e choose a random action and the other path chooses "exploitation" i.e it chooses an action based on the Q-value output of the Q-network. Once the action is sent to the environment, the rewards and the next state are received. These values are then used to fit the Q-network. This process takes place infinitely and the Q-network is eventually able to converge on the policy.

Algorithm 1 Deep Q-Learning algorithm

```

1: while  $t \leftarrow 1$  to  $N$  do
2:    $s = \text{getState}()$ 
3:    $a = 0$ 
4:    $\epsilon = \epsilon_0 - t$ 
5:   if  $\text{rand}(0, \phi) < \epsilon$  then
6:      $a = \text{rand}(0, 4)$ 
7:   else
8:      $\text{probs} = \text{CA\_DQN}(\text{state})$ 
9:      $a = \max(\text{probs})$ 
10:  end if
11:   $\text{send\_action\_to\_env}()$ 
12:   $r = \text{receive\_reward}()$ 
13:   $s' = \text{getState}()$ 
14:   $\text{target} = \text{copy}(\text{Q-network})$ 
15:  calculate Q-value
16:   $\text{targetNetwork.Q-value} = \text{calculated Q-value}$ 
17:   $\text{loss} = \text{MSELoss}()$ 
18:   $\text{optimizer} = \text{Adam}()$ 
19:  add tuple into replay buffer
20:  if  $t \% \Omega = 0$  then
21:    sample  $B_{RB}$  tuples from replay buffer
22:     $\text{target} = \text{copy}(\text{Q-network})$ 
23:    for  $B_{RB}$  tuples sampled do
24:      calculate Q-value
25:       $\text{targetNetwork.Q-value} = \text{calculated Q-value}$ 
26:       $\text{loss} = \text{MSELoss}()$ 
27:       $\text{optimizer} = \text{Adam}()$ 
28:    end for
29:  end if
30: end while

```

6 Handling Emergency Vehicles

Once we have trained an effective object detection algorithm, the idea is to narrow down on the lane that has the maximum number of emergency vehicles at that timestep and free up that lane.

A disadvantage of this approach is that we will not be taking the magnitude of emergency of the vehicles into account, but the only way of knowing this would be if the vehicles themselves transmit information to our algorithm, which cannot be expected of all vehicles.

Our algorithm will, however ensure fairness by selecting a random lane among the lanes with maximum number of emergency vehicles, if there is a deadlock.

For the purpose of addressing emergency vehicles, we use YOLOv7. YOLOv7 is currently the latest architecture [15] of the popular You Only Look Once object detection algorithm that is capable of predicting better bounding boxes at a high speed.

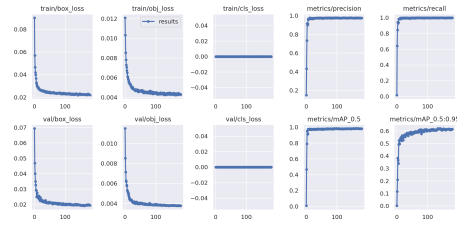


Fig. 2. Training results of the YOLOv7 model

7 Results

Fig2. shows a graph of the training results which represents the average waiting time of vehicles for our model.

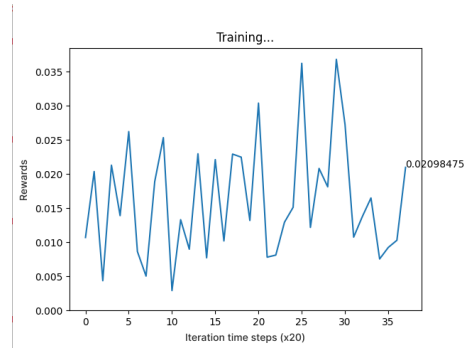


Fig. 3. Training results of CA-DQN model

References

1. Myungeun Eom and Byung-In Kim, "The traffic signal control problem for intersections: a review" in proceedings of European Transport Research Review, September 2020.
2. Syed Shah Sultan Mohiuddin Qadri, Mahmut Ali Gökçe Erdinç Öner, "State-of-art review of traffic signal control methods: challenges and opportunities" in proceedings of European Transport Research Review, 2020.
3. Javaid, Sabeen Sufian, Ali Pervaiz, Saima Tanveer, Mehak. "Smart traffic management system using Internet of Things." in proceedings of 20th International Conference on Advanced Communication Technology pp. 393-398, 2018.
4. Sonali P Kshirsagar, Priyanka H Mantala, Gayatri D Parjane and Kalyani G Teke. Intelligent Traffic Management based on IoT in proceedings of International Journal of Computer Applications vol. 157, no. 2, pp. 26-28, January 2017.
5. Bhuvaneswari, P. T. V. et al. "Adaptive Traffic Signal Flow Control Using Wireless Sensor Networks." in proceedings of Fourth International Conference on Computational Intelligence and Communication Networks, pp. 85-89, 2012.
6. A. D. Bharade and S. S. Gaopande, "Robust and adaptive traffic surveillance system for urban intersections on embedded platform," in proceedings of Annual IEEE India Conference (INDICON), pp. 1-5, 2014.
7. Erwin Walraven, Matthijs T.J. Spaan, Bram Bakker, "Traffic flow optimization: A reinforcement learning approach" in Engineering Applications of Artificial Intelligence Volume 52, pp. 203-212, June 2016.
8. Kovari Balint, Tettamanti Tamas, Becsi Tamas. "Deep Reinforcement Learning based approach for Traffic Signal Control," pp. 279-285, September 2021.
9. T. Nishi, K. Otaki, K. Hayakawa and T. Yoshimura, "Traffic Signal Control Based on Reinforcement Learning with Graph Convolutional Neural Nets," in proceedings of 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 877-883, 2018.
10. Sara El Hamdani, Salaheddine Loudari, et al., "A Markov Decision Process Model for a Reinforcement Learning-based Autonomous Pedestrian Crossing Protocol," in proceedings of 2021 3rd IEEE Middle East and North Africa COMMUNICATIONS Conference, pp. 147-151, December 2021.
11. Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," July 2022.
12. M. Chen, X. Yu, and Y. Liu, "PCNN: deep convolutional networks for short-term traffic congestion prediction," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 11, pp. 3550–3559, 2018.
13. Alexander Shustanov, Pavel Yakimov, CNN Design for Real-Time Traffic Sign Recognition, Procedia Engineering, Volume 201, pp. 718-725, 2017.
14. Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. IET Intelligent Transport Systems, 11(7):417–423, 2017.
15. Hard Parikh, RS Ramya, KR Venugopal, "Car Type and License Plate Detection Based on YOLOv4 with Darknet Framework (CTLPD)" in IOT with Smart Systems: Proceedings of ICTIS 2022, Volume 2, pp. 603-612, 2022.