International Conference on Advances in Materials, Computing and Communication Technologies
In Association with International Journal of Scientific Research in Science and Technology
Volume 9 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

Real Time Intelligent Traffic Signal System Using Deep Reinforcement Learning Technique

Sarvesh Srriram B[1], Sivaprasath K[1], Rajesh S.[2]

[1]UG Student, Department of Information Technology, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

[2]Associate Professor, Department of Information Technology, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

ABSTRACT

Traffic is the world's daily common problem happening all around the world. To normalize the traffic, traffic signals were made. Though it prevents heavy traffic, it is not sufficient. Mainly in India, traffic is one of the major causes for many pollution such as air and noise pollution and even causes accidents. This problem cannot solved with perfection in real time, but optimized with maximum efficiency with automation. With the help of Deep Reinforcement learning, this problem is optimized with maximum efficiency by assigning traffic signals in such a way that the waiting time for each vehicle is minimized on all side of the road.

I. INTRODUCTION

Artificial Intelligence plays a vital role now a day in our day to day life. There are some places where human is replaced by machines where they feel difficulty to solve some complex problems. One of the most common complex problems happening everywhere is Traffic. Traffic leads to many problems such as pollution, sometimes accidents and people are affected because of the delay due to it. Traffic signal plays a major role for controlling traffic lane by lane. But the old conventional method of traffic signal control is an inefficient solution leads to congestion of vehicles which also creates delay and wastage of energy for the travellers. It also doesn't have any real time problem solving technique. In some scenarios, the sensors like underground loop inductors are used to detect vehicles in front of the traffic signal, but they are processed only with the presence of vehicles where the duration of traffic signal is not determined. This method is improved by taking the real time information as input and adjusting the traffic cycle duration according to the data gathered. This work is developed with Deep Reinforcement learning network to control the traffic signal cycle.

II. LITERATURE REVIEW

OVERVIEW:

In 1993 S. Chiu and S. Chand proposed adaptive traffic signal control using fuzzy logic system. It is very simple way of approach to monitor the traffic flow with the help of fuzzy logic. In this, signal timing parameters uses only buffer (local) information

to increase efficiency of traffic flow dynamically. Also, intersection is assigned by cycle time and phase split, offset by three parameters. Changing signal is based on the current parameters smoothly. Main disadvantage is Fuzzy Logic system doesn't have the capability of machine learning as well as neural network type pattern recognition.

Karub et.al (1997) used Tensor flow for large-scale machine learning on heterogeneous distributed system. This Traffic signal control is based on the approach of Heterogeneous distributed system by. In this Heterogeneous distributed system is very flexible and it is used for many algorithms. Tensorflow is used for proceeding the machine learning algorithms and used for implementing such a algorithms. It also has many drawbacks in benchmark tests tensorflow lacks in both speed and using storage many user are using the windows environment only Tensorflow properly works in Linux Environment only. It is slow process.

Distributed intersection management protocol for safety, Efficiency and driver's comfort is proposed by X. Liang, T. Yan, J. Lee and G. Wang during 2018. Methodology used in DIMP was evaluated and result is both effective and very efficient in managing vehicle traffic at each intersections. DIMP are used to Instruct or adjust the vehicles speed dynamically is based on the driver's comfort and to follow the safety rules, both the constraints had been applied. Drawbacks are it needs high processors like Intel Xeon to execute and it needs more memory to access

Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow was proposed by RuiminKe, Zhibin Li, JinJun Tang, Zewen Pan and Yinhai Wang. Their work is about the estimation of traffic flow with the help of UAV video based ensemble classifier. Here the traffic density is identified from UAV in the form of video. Using YOLO (You Only Look Once) algorithm, the different kind of vehicles can be found. Though the traffic flow is constant in that particular area, we cannot say when it will happen. So it automatically leads to continuous monitoring of video from UAV. The main drawback is the cost. Though this technique is efficient, this kind of implementation needs a lot of money and in our project, it is minimized.

DRL STUDY:

Deep Reinforcement Learning is the combination of Deep learning and reinforcement learning. This is applied on many wireless communication system but not so much in Traffic system. In last 3 implementations, these are the main drawbacks. Firstly, these models are in a simple cross-shape intersection and only with traffic. Secondly they simply schedule the lanes but didn't mention about changing the duration of the signal. Then thirdly, DRL is a fast-developing technique which is not implemented in traffic scenario.

III. SYSTEM STUDY

OVERVIEW:

The System Study model describes the purpose of the project, functionalities and necessary requirements which are needed for developing the project.

PURPOSE:

The main purpose of the proposed work is to minimize the vehicle's waiting time to the best and to minimize traffic congestion thereby able to satisfy the vehicle drivers as well as passengers. Another purpose is to give more importance to emergency vehicles where there will be no need of neglecting the traffic signal by the emergency vehile drivers. This DRL network can implemented in every traffic signals available now and once the machine starts learning, there will be no need of training additional data here.

FUNCTIONAL REQUIREMENTS:

Python version 3 or higher and python navigator such as Anaconda is required and python libraries such as numpy, matplotlib, torch, TraCI, OpenCV, etc,

are some of the functional requitements needed for implementing this project.

NUMPY:

NUMPY is the library for scientific computation which can support n dimensional arrays and matrces and contains high level mathematical functions like linear algebra and Fourier functions. It also have tools for integrating C, C++ and FORTRAN. NumPy contains several complicated mathematical formulae as a function where using such functions, those complicated problems can be solved easily by simply giving input numbers or values as a parameters. In pythonNumpy is mainly used for accessing the arrays. It also deals with some extra functions like Domain of linear algebra and matrices etc. First we want to import the package numpy and then we want to pass the list into the array functions it start to converts it in to a numpy array. While performing the logical or mathematical operations. It has given a permission to access the n-number of array and matrices. In this we can also access the manipulate arrays also. There are some advantages in using the Numpy it has given the better performance for N-number of dimensional array object. It also has tools for separating the code from c languages and also used in c++ languages etc. It has contain broadcasting function. NumPy Encapsulates N-number of dimensional arrays for data types which is homogenous. It perform with many operations.

MATPLOTLIB: MATPLOTLIB is a python-written library for creating figures and plot points thus vector files can be produced effortlessly without the help of GPUs.

SUMO: SUMO is an acronym stands for Simulator for Urban MObility. It is an open source traffic simulator package where the project is to be implemented. It is highly portable and microscopic.

OPENCV: OpenCV is a python library contains programming functions mainly focused on real time computer vision developed by Intel Corporation.

TRACI: TraCI stands for Traffic Control Interface. It is the python library used for accessing the running traffic simulation in SUMO GUI and using this, we can able to retrieve the values from the simulation.

TQDM: TQDM is a Python Library used to track progress during training process which makes a progress meter with iterable using training dataset.

PYTORCH: PyTorchis the free open source python library known for deep learning. It is an end to end platform having its own flexible ecosystems and tools. It is developed by Facebook AI Research lab also known as FAIR. This package is torch library based one where deep learning is implemented for research purpose. PyTorch is recently used in Tesla Autopilot.

IV. SYSTEM DESIGN

This section is to represent the overview of the whole system. Section 4.1 explains the overall architecture flow of the system Section 4.2 describes the concepts of DEEP CNN and Section 4.3 explains the concepts of DEPP Q Network

4.1 OVERALL ARCHITECTURE

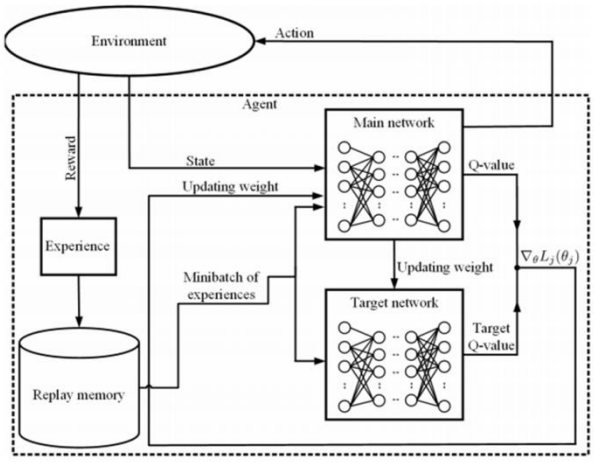

Fig. 1 System Architecture

Figure 1 shows the overall system architecture of the system where our input file video is divided into individual frames of 32x32. After that determine the count the vehicles cumulative waiting time for all the vehicles. Then find the Q value and based on the Q value, it determines the ranking based on the rank change the signal. After finding Q value each and every time, it automatically learns the experience from the Replay memory. The decision is taken with the process called Markov's Decision Process (MDP)

4.2 DEEP CNN

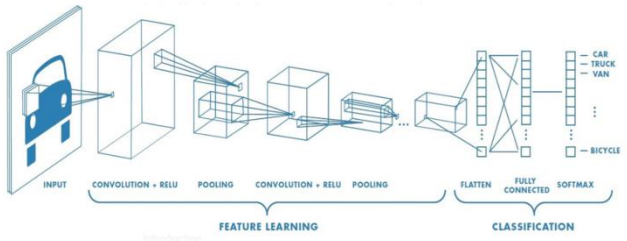

Fig. 2 Deep CNN

Deep Convolution Neural Network is a Neural Network which has one or more than one convolutional layers it majorly focusing on the image processing, segmentation, classification in Figure 2. It identify the vehicles on the road it has pooling layer with 2x2 filter that will reduce the Dimensionality by half eg. 16x16 and used to changed it into a linear array of 1D vector which will be useful to read the input to fully connected layer that can holds 1000 neurons with activation function as ReLu. In this output layer consists of 10 neurons that are labelled using the softmax activation function.

4.3 DEEP Q NETWORK

Figure 3 shows that Deep Q-Network it is End to End RL approach and it is quite flexible. It has capability to find the optimal solution for wide range of inputs. It is combination of reinforcement learning and deep neural networks. In learning process it use two neural networks

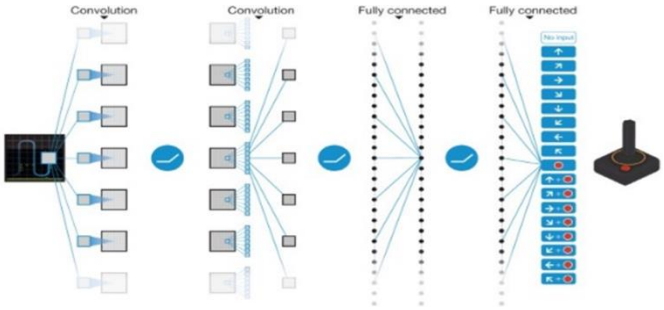

Fig: 3 Deep Q Network.

V. IMPLEMENTATION METHODOLOGY

OVERVIEW:

This section describes the Techniques and algorithms used and implemented in this project. This project is on the concept of DRL and it is implemented with the algorithm known as Doubling Duelling Deep Q Network. It is the combination of Doubling DQN and Duelling DQN.

DEEP REINFORCEMENT LEARNING:

Deep reinforcement learning is a one of the group of machine learning and artificial intelligence. It is framework to determine the sequential decision-making problems by learning the concept. It provides occasional rewards by Trial and error methods. Result is a combination of neural networks and a reinforcement learning. Result is also known as Reward based on that it is learning from interaction and learn what to do and how to map situation to actions. It has five sequential process namely Agent, Environments, States, Action and Reward. In the Environment is Physical world in which the agent operates and States Determines the current situation of the agent, Action In this set of state has finite number of elements Agent want to interaction with environment through action based on the current state. Agent is role is to identify action of state which is been performed in the environments. In this convolutional Neural Network is used to feed-forward artificial neural network that is used for analysing visual imagery. This Convolutional Neural Network consist of the three main layers, they are input layer,

hidden layer and output layer with multiple hidden layers such as convolution, ReLu (Rectified Linear Unit), Pooling layers, Normalization layers and Fully connected layer.

For example, consider a machine plays a game that should be completed by it with highest score. According to the rewards, the machine learns how to play and after several iterations, the machine will learn to find out how to achieve target score.

3DQN:

Deep Reinforcement learning algorithms are grouped Deep Q Learning with deep neural networks and implemented in high dimensional environment like robotics. In Doubling Duelling Deep Q Network (3DQN), the number of states will be higher which cannot be stored in a consistent way. For that, the data are stored in the form of experiences and we calculate the cumulative reward (Q value from DQN). With the state-of-the-art technique, the cumulative reward for the current target state is calculated. This algorithm is known as Double Duelling Deep Q Network. In Double DQN, the primary network produces target Q value by using the following equation.

$$Q_{target}(s, a) = r + \gamma Q(s', \arg_{a'}\max(Q(s', a'\theta))\theta^{-})$$

By using this, we can calculate the Q value and additionally use the e-greedy algorithm to maintain the balance between exploration and exploitation in choosing actions.

Duelling DQN has determined the Q value based on the current state. Each current actions are compared with the previous actions. The value of a state V(s;) represents the predicted overall rewards by taking probabilistic actions. Each and every actions is defined a A(s, a ; . The sum of the value V results as reward Q it has function A

$$Q(s, a, \theta) = V(s; \theta) + (A(s, a, \theta) - 1/|A| \sum A(s, a'; \theta)$$

If the A value of action is positive action has shown the better performance in numerical rewards compared to the average performance of all possible actions. In this DuelingDQN effectively increase the performance in reinforcement learning.

VI. EXPERIMENTAL RESULTS

OVERVIEW:

This chapter details our project results and for each step, screenshots of output are attached and explained below.

Fig: 4 main.py Execution

Figure 4 is the opening page of the Anaconda prompt where our project starts to execute. The Anaconda environment should have both conda and Python pip environment (Since TraCI is available in pip)

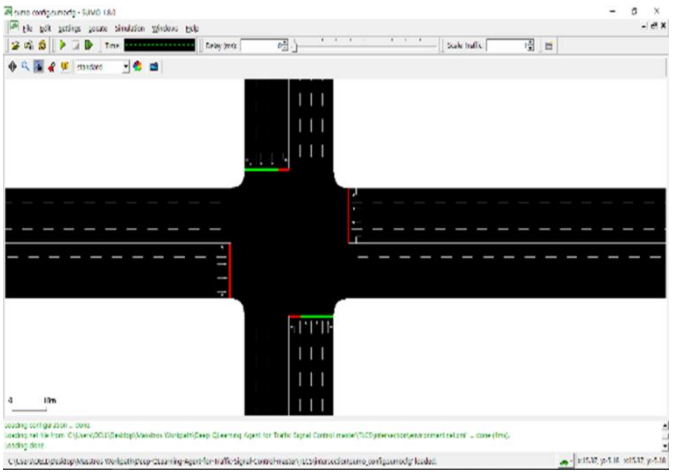

Fig:5 SUMO Environment

Figure 5 is the simulated traffic scenario model. It contains two intersections one after the other and designed with the help of SUMO environment where there are 3000 steps. The vehicle types included are Car, Truck/Bus and emergency Vehicle. In this

simulation, emergency vehicles are in red, cars are in white and trucks are in green color.

Fig: 6 Initial Result (0th Episode)

Figure 6 is the initial result of the signal control. The machine starts to learn the traffic control, and the performance is initially poor. In this case, he rewards will be low for the machine.

Fig: 7 Result after 100 Episodes

Figure 7 represents the result after 100 repeated iterations. Once the reward becomes low, the machine aims for higher reward by performing certain actions thereby gets the higher reward. In our project, the less the minimal waiting time the vehicle achieves, the more the reward the machine gets. Then it gives the satisfying controlled scenario.

VII. EXPERIMENTAL COMPARISON

OVERVIEW:

This module explains the result and efficiency of our project comparing to the conventional method in the form of graphical representation.

HIGH TRAFFIC SCENARIO

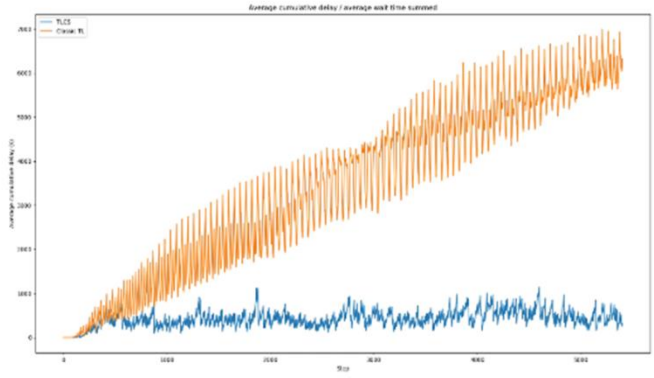

Fig:8 Queue length comparison in high traffic (lower is better)

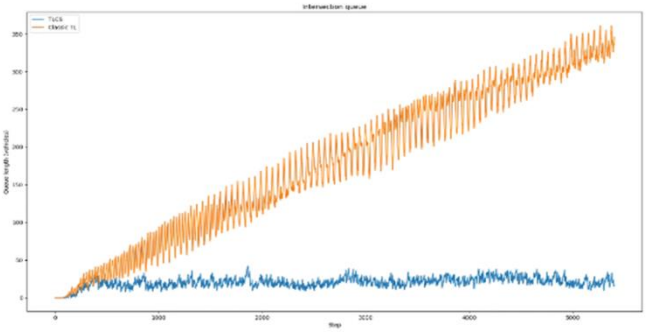

Fig:9 Comparison of average waiting time of vehicles in high traffic (lower is better)

This graph represents the queue length of the road, which is also represented as Traffic density. Comparing to conventional method, this method is way more efficient and we can see that th for the first few steps, the values are close and density in conventional mode becomes higher and TLCS remains the same.

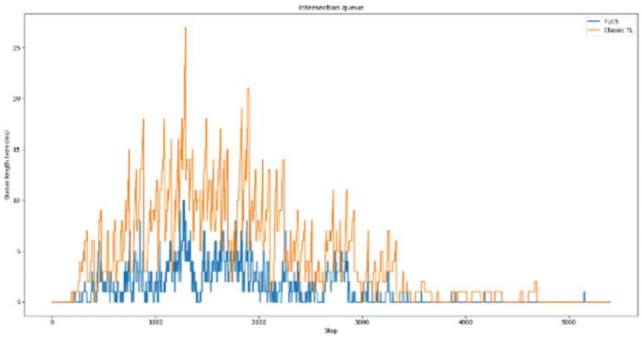

Fig: 10 Queue length comparison in low traffic (lower is better)

This graph shows the comparison of average waiting time of the vehicle on the road. Comparing to conventional technique, with the application of DRL technique. the waiting time is comparitively very low thereby the optimal scheduling of traffic light and time duration is achieved in higher traffic scenario.

LOW TRAFFIC SCENARIO:

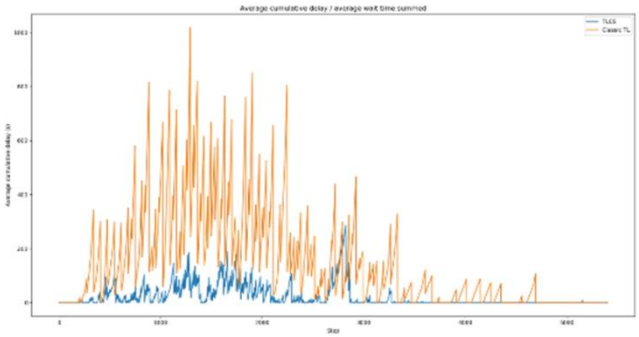

Fig: 11 Average waiting time comparison of vehicles in low traffic (lower is better)

In case of low traffic scenario, queue length will be shorter than the length in higher traffic in conventional method. Here the optimal scheduling is achieved and the queue length is reduced.
Here in low traffic scenario graph, generally the waiting time will be less in conventional method. By applying TLCS, it is further reduced optimally.

VIII. CONCLUSION

In our work, Deep reinforcement learning and the Deep Q Network is implemented for controlling Traffic Signal. This project has no particular datasets since the system trains itself. Coming to the current traffic state scenario, actions are done with the help of previous experiences which acts as a dataset. Based on the density of the lane, priority/weight of the vehicle, cumulative average idle time of the vehicles, the signal and the duration of the signal is made. This achieves the minimal optimal waiting time and reduction of the traffic density.

IX. FUTURE WORK

In future, this algorithm can be implemented in such a way that, the data can be collected from videos and from that, the waiting time, traffic density is retrieved and implements the real time scenario. Comparing the last work, instead of simply locating the vehicles and doing actions, the learning process is also done and the optimal results can be obtained with this algorithm.

WEB REFERENCES:

1. https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html
2. https://www.hindawi.com/journals/jat/2020/6505893/
3. https://sumo.dlr.de/docs/TraCI/Protocol
4. https://wiki.pathmind.com/deep-reinforcement-learning
5. https://www.towardsdatascience.com

X. REFERENCES

[1]. S. S. Mousavi, M. Schukat, P. Corcoran, and E. Howley, “Traffic light control using deep policy-gradient and value-function based reinforcement learning,” IET Intell. Transp. Syst., vol. 11, no. 7, pp. 417–423, Sep. 2017

[2]. Xiaoyuan Liang , Xunsheng Du , Student Member, IEEE, Guiling Wang, Member, IEEE, and Zhu Han , Fellow, IEEE, “A Deep Reinforcement Learning Network for Traffic Light Cycle Control”, IEEE, Transaction on Vehicular Texhology Vol. 68, No. 2, Feb 2019

[3]. W. Genders and S. Razavi, “Using a deep reinforcement learning agent for traffic signal control,” unpublished paper, 2016. [Online]. Available:https://arxiv.org/abs/1611.01142

[4]. X. Liang and G.Wang, “A convolutional neural network for transportation mode detection based on smartphone platform,” in Proc. IEEE 14th Int.

Conf. Mobile Ad Hoc Sensor Syst., Oct. 2017, pp. 338–342.

[5]. M. Abadiet al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," in Proc. 12th USENIX Conf. Oper. Syst. Des. Implementation, Nov. 2016, pp. 265–283.

[6]. KE et al.:Real-time traffic flow parameter estimation from UAV video based on Ensemble Classifier and Optical flow,IEEE Transaction on Intelligent Transportation Systems, Jan - 2019

[7]. A real-time deep learning approach for intelligent traffic control," IEEE Wireless Commun., vol. 25, no. 1, pp. 154–160, Feb. 2018.

[8]. M. Abdoos, N.Mozayani, and A. L. Bazzan, "Holonic multi-agent system for traffic signals control," Eng. Appl. Artif. Intell., vol. 26, no. 5, pp. 1575–1587, May/Jun. 2013.

[9]. I. Arel, C. Liu, T. Urbanik, and A. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," IET Intelligent Transport. Syst., vol. 4, no. 2, pp. 128–135, Jun. 2010.