Developing a Deep Q-Learning and Neural Network Framework for Trajectory Planning

Venkata Satya Rahul Kosuru and Ashwin Kavasseri Venkitaraman

Abstract — **With the recent expansion in Self-Driving and Autonomy field, every vehicle is occupied with some kind or alter driver assist features in order to compensate driver comfort. Expansion further to fully Autonomy is extremely complicated since it requires planning safe paths in unstable and dynamic environments. Impression learning and other path learning techniques lack generalization and safety assurances. Selecting the model and avoiding obstacles are two difficult issues in the research of autonomous vehicles. Q-learning has evolved into a potent learning framework that can now acquire complicated strategies in high-dimensional contexts to the advent of deep feature representation. A deep Q-learning approach is proposed in this study by using experienced replay and contextual expertise to address these issues. A path planning strategy utilizing deep Q-learning on the network edge node is proposed to enhance the driving performance of autonomous vehicles in terms of energy consumption. When linked vehicles maintain the recommended speed, the suggested approach simulates the trajectory using a proportional-integral-derivative (PID) concept controller. Smooth trajectory and reduced jerk are ensured when employing the PID controller to monitor the terminals. The computational findings demonstrate that, in contrast to traditional techniques, the approach could investigate a path in an unknown situation with small iterations and a higher average payoff. It can also more quickly converge to an ideal strategic plan.**

Keywords — **Autonomous Driving, Proportional Integral Derivative (PID), Q-Learning, Trajectory and Path Planning.**

I. Introduction

There is considerable interest in autonomous driving by authorities, business, and academics worldwide. Although the concept of an autonomous car has been around for almost a century, it wasn't until the 1980s, with the introduction of the PROMETHEUS project, that it gained popularity. Sensation, machine learning, decision, and impulse control are the four key layers that make up an autonomous vehicle's architecture. The functions of each level and their interconnections have previously been outlined and expanded in several research. these techniques to situations that deviate from the norm [1]. In a highly complicated system like an automobile, a high level of safety is described as low accident statistics. Because there's so many, fixing issues in a current scheme is frequently expensive. Although there are fewer issues as the system ages, consultative approach to them grows more challenging. The development of autopilot technologies is currently at this point. A vast amount of information is required for optimization in the inexperienced autonomous vehicle system. Currently, the majority of data are gathered through risky and expensive road tests. Extreme situations including crashes, near-collisions, complicated signal junctions, and roundabouts make it difficult to replicate tests [2].

Reinforcement learning has recently been used for robotic challenges. It is suggested in Deep Q-Network (DQN), which integrates Deep Neural Network and Reinforcement Learning and could handle continuous situations of discrete operations. Then the research continuous high dimensional state space-based off-line deep reinforcement learning method based on neural Q-Network is presented. expands DQN to accomplish numerous objectives. The suggests using previous experience replaying technology to enhance DQN's functionality [3]. For the purpose of increasing the effectiveness of specimen collection and usage, they suggest an expert experience replay technique. The action strategy is followed by the noise-adding Ornstein-Hollenbeck (OU) procedure. To enhance the effectiveness of the network, noise is added to the variable level. While DQN could enhance the processing capability of high-dimensional phase space, it remains challenging to cope with high-dimensional continuous action space. Q-learning could handle the low-dimensional issue in finite interval. Continuous action space could be handled by the actor critical technique; however, the adoption of a randomization strategy creates difficulties for the network to convergence [4].

The stochastic method and the deterministic method are the two broad technique that could be utilized to classify path planning systems. The stochastic technique, which is commonly understood as an estimate technique, just looks for a workable solution. In contrast, the predictable strategy, called as the exact approach, follows a series of precisely defined stages to develop a unique navigational route. Because of this, the result of the stochastic technique may not always offer the optimum option to meet the needs of the design. The stochastic technique has taken over as the primary method for maritime avionics due to its superior accuracy and completeness. Trajectory planning, which can be described as finding a temporal movement law along a given geometric path such that certain conditions set forth for the trajectory attributes are satisfied, is a fundamental issue in automation [5]. The goal of trajectory planning is to produce the standard inputs needed for the manipulator's control mechanism to carry out the movement. The trajectory planning computation sources are the geometric path, the kinematic restrictions, and the dynamically restrictions; its

Submitted on Novemeber 29, 2022.
Published on December 26, 2022.

V. S. R. Kosuru, Lawrence Technological University, USA.
(e-mail: vkosuru@ltu.edu)

A. K. Venkitaraman, University of Cincinnati, USA.
(e-mail: venkitak@mail.uc.edu)

DOI: http://dx.doi.org/10.24018/ejeng.2022.7.6.2944

output is the itinerary of the limbs (or of the output shaft), represented as a time sequence of location, speed, and measurement result. Trajectory planning in time-critical street situations, including lane changes in continuously adjusting road traffic, is among the difficult challenges and it is shown in Fig. 1. When a car wishes to leave the interstate or make a right turn at the subsequent crossroads, it must drive recklessly [6].

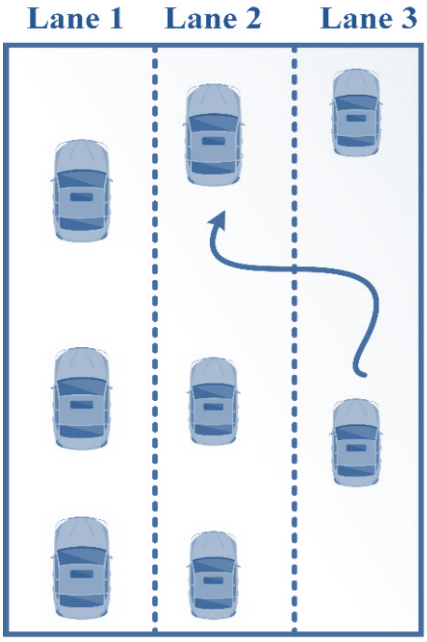

Fig. 1. Vehicle lane changing flow.

Additionally, the variety of driving conditions brought on by varied roadways, traffic laws, obstacles, and traffic participation make trajectory forecasting more challenging. The planner would have to provide safe and operationally feasible trajectories for a variety of driving circumstances in order to handle the complicated and time-critical circumstances. Additionally, real-time capabilities are needed for this planner to guarantee a prompt response to changes in the driving experience. Numerous motion planning strategies have been investigated recently to address these dynamic challenges [7]. These strategies can be divided into two distinct categories sampling-based strategies and optimization-based strategies. By choosing the ideal choice from the traffic-free trajectory candidates, the sampling-based approach produces the best path. These characteristics render the aforementioned tactics inappropriate for on-road driving situations. Road safety should be prioritized in on-road path planning with both the ability to design a traffic free planning path, while still considering passenger satisfaction and the usage of road structural features into account [8].

The capacity of the automobile to respond quickly enough to avoid impediments or other route occupants was the focus of some published literature. Through superimposing the roadblocks and the roadmap, with the roadmap's downward orientation defining the accident safe path, the artificial potential field was created. With appropriately developed barrier restrictions, the model predictive control (MPC) was used, integrating path prediction and monitoring.

Due to its ability to calculate a plausible trajectory and a series of direct signals to monitor it concurrently, model predictive control (MPC) approaches are frequently employed in the research. Nevertheless, high-speed trajectory management necessitates a detailed modelling of the automobile to take into consideration its dynamic constraints, which mostly result from the intricate and highly nonlinear interplay between tires and surface. Wheel kinematics are typically far quicker than alterations in the vehicle's spatiotemporal condition, which makes things much more challenging. As a result, the available research is typically split into two categories: short-term path monitoring for high-speed or low-adherence implementations utilizing wheel dynamics modelling, and medium-term (a few minutes max) path intending such as avoiding obstacles for low-speed implementations, primarily reliant on simple robot manipulator models [9].

When contrasted to other strategy instruction, imitation learning is simpler, without need for formal scripting nor rigorous mathematical analysis. For instance, designing a probabilistic model is required in reinforcement instructional methods, which can be a difficult undertaking. On the contrary hand, a wealth of sensory input, including aesthetic, temperature, topological, and many other distinctive properties of work situations, is now available thanks to developments in analytical techniques. Because of this, computers can interpret the gathered data with ease and produce the required decision orders for a specific activity [10]. Autonomous systems frequently employ imitation techniques. Replication methods are receiving a lot of interest in implementations since they are easy to use and compatible with many different learning strategies. The expectations and limitations in situations where real-time observation of a following request is essential are significantly relaxed by learning from illustration. For example, in fast-response and time-sensitive systems, like autonomous vehicles, it may become physically challenging to investigate an appropriate and generalizable objective function. The learning algorithm for an imitating technique, on the other extreme, must be properly constructed in order to produce reliable and potent models. To address this issue, academics have typically created imitation techniques combined with other intelligence algorithms, such as supervised learning, deep reinforcement learning, parallel supervised learning, deep reinforcement learning, and several others [11].

Lattice organizers, in contrast are excellent at producing workable pathways and integrating restrictions, but they could produce partial graphs that result in curvature discontinuity. Techniques for machine learning are the alternative strategy for determining trajectory. Imitation Learning, a supervised training technique, has produced some encouraging outcomes. This approach does not, however, ensure stability or an ideal resolution and may not transfer to well difficult environments. So, to overcome these issues the following method is developed. This study uses experience replay and situational knowledge to overcome these challenges and proposes a novel deep Q-learning approach. The network edge node's deep Q-learning approach to path planning is suggested as a way to improve the energy efficiency of autonomous vehicles' navigation. The proposed method mimics the path that used a proportional-integral-derivative (PID) concept controller when linked vehicles maintain the advised speed. When the PID controller is used to monitor the terminals, a smooth trajectory and less jerk are guaranteed [12].

The section I provides the introduction on the various technique involved in autonomous driving and trajectory planning. Section II provides the related works. Proposed methodology is presented in the section III and the result obtained is presented in section IV. Finally, the paper is concluded.

II. Related Work

Given its great practicability, sampling-based motion planning (SBMP) is a significant path planning strategy in autonomous vehicles. Probability sampling, which is at the center of SBMP systems, is what determines if a pleasant and traffic-free path could be established in real-time. Although certain bias sampling techniques have already been discovered in the research to speed up SBMP, the path produced by these techniques may result in abrupt lane changes. They suggest a new learning paradigm for SBMP to tackle this issue. Specifically, to increase the precision in predicting the purpose of nearby vehicles, they build a unique automatic labelling technique and a 2-Stage prediction method. Then, using the knowledge gained from the experiences of the human operators, researchers create an emulation learning system to produce data samples. By intelligently choosing the required samples that could produce a seamless and traffic-free trajectory and prevent sharp lane changes, researchers design a new bias sample technique to speed up the SBMP method. The suggested sampling technique performs better than existing sampling techniques in terms of computation duration, trip time, and path uniformity, according to data-driven studies. The outcomes also demonstrate that our system outperforms actual drivers [13].

The lack of information in images of low-light road scenarios might make networked automated driving more likely to crash (CAVs). Consequently, a low-light image improvement model that is efficient and economical is required for safe CAV driving. Despite several attempts, image augmentation remains to be a good solution, particularly in conditions with very little light (e.g., in rural areas at night without streetlight). Researchers created a light improvement net (LE-net) predicated on a convolutional network to solve this issue. In order to create set of images for simulation process, they firstly developed a production pipeline to convert daylight photographs into low-light images. The produced low-light images were then used to train and evaluate our suggested LE-net [13].

For real-world automated driving technologies to be safe and effective, it is essential to be capable of anticipating the paths of the nearby cars. Deep neural network models for forecasts have been provided in past projects employing a thorough prior map that explicitly describes the laws of the road, such as lawful traffic direction and legitimate roundabout routes, and includes driving lanes. Research utilizes a map created from purely perceptual data because it would be impossible to presume that all places have detailed earlier maps. Prediction problems are made more challenging by the fact that such maps do not directly indicate traffic laws. We suggest a brand-new approach built on generative adversarial networks (GAN) to address this issue. In our architecture, a differentiator could determine whether anticipated trajectories adhere to traffic laws, and a generation could forecast trajectories that do. By projection paths onto to the map using a variational functional and establishing positional interactions among paths and barriers on the mapping, the method implicitly retrieves road rules. In order to forecast different future paths, researchers additionally expand the paradigm to include multimodal forecasts. In terms of path inaccuracies and the percentage of paths that fall on navigable lanes, empirical data demonstrate that our system performs better than other cutting-edge methods [14].

In this paper, researchers suggest a fresh method for employing control strategy to move a car quickly along a predefined course. Researchers utilize a simple second order integrator framework, which is restricted to fit the vehicle's possible dynamic range, in place of precisely simulating the motion of the vehicle, which severely limits the design range that could be considered in real-time. This approach additionally contains velocity planning, enabling the vehicle to constantly adjust its movement to the design of the road, in contrast to conventional MPC systems that only accept a desired speed as information. The method may be utilized in real-time to produce plausible paths that could be monitored utilizing a straightforward control scheme, according to modeling findings on a very precise particular vehicle. In addition, contrasted to kinematic systems frequently employed in path planning, the plan is more reliable and produces good paths when the reduced approach is employed. While still primarily hypothetical, this approach opens up a number of avenues for further study. Firstly, the straightforward dynamic model's effectiveness even at high velocities enables imagining longer planning horizons without compromising computational efficiency. Future studies should investigate the validity of employing fully benchmark, which can be paired with effective mixed-integer optimization methods to enable optimal decision-making, for example for passing or lane-change decisions [15].

The movements of pedestrians as well as other motorists should be considered for automated vehicle movement planning and management to be preemptive and secure. In this research, researchers provide a trajectory-tracking control-based framework for vehicle dynamics planning and regulation that takes moving impediments into account. The projected movement of each pedestrian is converted into restrictions for the MPC issue using the modelled pedestrian variables that are given into the forecast layer. To demonstrate the effectiveness of the architecture, simulation and experimental verification was carried out with fictitious pedestrians crossing the street. According to the results of the experiments, the controller could remain stable even when there are considerable input delays little while requiring very little processing power. The created approach was also further validated in simulations using actual pedestrian data. The upcoming study will focus on investigating scenarios in which pedestrians unexpectedly emerge to the architecture as a result of sensor occlusion. Additionally, researchers intend to further validate the architecture in more difficult crossings with pedestrians walking. Last but not least, future studies will concentrate on creating a strong control mechanism with recursive viability guarantees [16].

Previous to 2010, different image identification challenges were addressed in the field of image identification by mixing classification algorithms created image local features—and machine learning approaches. However, numerous deep learning-based picture identification techniques have been presented since the year 2010. In generic visual recognition challenges, deep learning-based image recognition techniques have surpassed prompts by a considerable margin. In order to better understand how deep learning is used in the domain of image identification, this paper will also examine the most recent developments in deep learning-based autonomous vehicles. The advancement of end-to-end learning and deep reinforcement education algorithms for "judgment" and "regulation" of driverless driving is expected to meet great standards in the years ahead. Future prospects include "recognition" for input photos as well. It is also desirable to move beyond visual explanation to the verbal description via interaction with natural language processing systems. Citing judgment reasons for outputs of deep learning and reinforcement learning is a big issue in real implementation [17].

One of the most challenging and potentially disruptive issues that the robots and artificial intelligence services are now dealing with is autonomous driving. Self-driving cars (SDVs) are predicted to reduce traffic fatalities, save thousands of lives, and enhance the standard of living for a larger number of people. There remains to be accomplished in order to design a system that can perform as well as the greatest human operators, despite ongoing awareness and a variety of industry firms working in the driverless area. This is due, among other things, to the significant degree of unpredictability in traffic behaviour and the wide range of conditions that an SDV might experience on the roadways, which makes it exceedingly challenging to develop a completely generally applicable solution.

A driverless car must take this unpredictability into consideration and foresee a wide range of potential traffic player actions in order to assure safe and effective functioning. They tackle this important issue and provide a mechanism to forecast numerous potential actor paths while also assessing their chances. The technique converts the environment of each player into a vector file, which deep CNN models then utilize input to generate the necessary features for the task automatically. The approach was effectively evaluated on SDVs in closed-course tests after rigorous offline assessment and comparison to cutting-edge baselines. Autonomous vehicles must take into account a variety of potential future paths of the surround players owing to the intrinsic ambiguity of traffic behaviour in order to provide a safe and effective ride [18].

III. Methodologies

A. Proposed Method

In reinforcement instructional strategies, the agent learns action plan from the mapping of the surroundings to actions to maximize the value of the reward in a reinforcement learning program and the rewards offered by the environment serve as assessments of the effectiveness of activities. In a setting of action and assessment, RL systems learn new information and develop better action plans to adjust to the surroundings. In this section, the author presents the development of deep Q-learning on autonomous driving vehicle and trajectory planning using the open CV data. An approach to learning that comes close to evolutionary algorithms is reinforcement learning. It works to discover the largest cumulative rewards in each state as its optimization problem and selects the best course of action step-by-step. Reinforcement intelligence enables a robot to automatically learn an ideal behaviour through trial-and-error adaptation to the environment without the need for either positive or undesirable labels. The reinforcement learning structure is shown in Fig. 2, where the agent chooses an action based on the Q-table and performs it. The environment subsequently provides the agent with a state and a reward. Q-learning is the reinforcement learning algorithm that is most frequently employed.

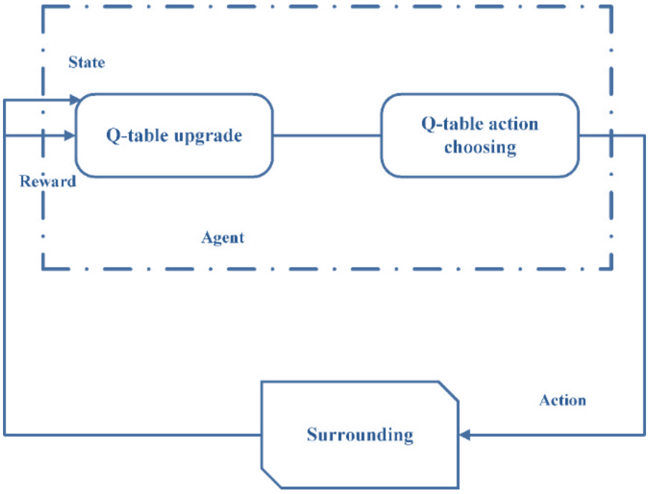

Fig. 2. RL block diagram.

The state of the surroundings (S), action (a), reward (r), strategy, value, attenuate component of reward (μ), transitional type of surroundings ($T_{s\prime}^{a}$), and exploration rate are the eight fundamental components of reinforcement learning.

The agent's surroundings are represented by the environment state (s). The agent may be in various environmental conditions at various points in time. S_t, which denotes a state in the surroundings state set, is the agent's surroundings state at time t.

Each individual action (A) representation of a behaviour in various states. For instance, the action performed by the agent at time t is noted as A_t.

Environment reward is the reward value input the robot receives from the surroundings after attempting to carry out tasks during environment exploration. Positive feedback could be specified as a number, and negative feedback would be specified directly as a negative value or zero.

Value represents the value that the agent possesses after doing the corresponding action (A) in accordance with the policy (α) and state. In most cases, the value function is represented by $v_\alpha(s)$, which also embodies the assumption of the reduction reward function. The current and previous rewards have an impact on the value of the value function.

After multiplying the previous reward by a discount factor and adding the total, the following action will result in a delay reward of U_{i+1}. Equation (1), where is the discount component, shows the general statement of the functional form.

$$v_\alpha(s) = E_\alpha(U_{i+1} + \mu^2 U_{i+1} + \cdots | S_t = s) \quad (1)$$

B. Q-learning

Researchers initially attempted to combine RL and neural networks. However, the combination of Off-Policy, linear regression, and bootstrapping reveals RL inconsistency or even diverge. The Deep reinforcement learning field wasn't set off until the Deep Mind Team developed Deep-Q learning. Q-learning is the reinforcement training algorithm that is most frequently employed. The reinforcement learning architecture is shown in Fig. 2. The agent chooses and performs an action in accordance with the Q-table, after which the surrounding provides the agent with a state and a reward. The deep-Q learning network has since undergone extensive development. The Q-table is an ideal strategy action significant demand in Q-learning, and it get an upgrade as mentioned in (2).

$$Q(s,a) \leftarrow Q(s,a) + \delta\left[r + \mu \max_{a'} Q(s',a') - Q(s,a)\right] \quad (2)$$

In (2), the learning rate is denoted as δ with the discount factor μ and the instant reward and the next state after execution action and instant reward is represented as $s' and\ a$. The selected action state is denoted as a' and the maximum cumulative reward value to the corresponding state is denoted as $\max_{a'} Q(s',a')$.

A redesigned reinforcement learning powers the smart vehicle system shown in Fig. 3. In our method, researchers replace the Q-table with a neural network and incorporate heuristic knowledge. In this research, the vehicle current state serves as the network's input, and its return is the projected cumulative reward for each action. The vehicle selects actions immediately in accordance with the actual output of the neural net or predictive knowledge rather than by querying the Q-table.

Neural network model process humongous amount of data to process the data inputs, but when a vehicle investigates an unfamiliar environment, it is difficult to have enough training dataset sets ready in preparation. As a result, the autonomous vehicle gathers experience data produced during movement in the form of (s', a'', r, s) and records them in replay memory. The quantity of training samples is ensured in this method [19].

The expected return $E[U_i | s_i = s, a_i = a]$for a state-action pair succeeding a policy δ, where R_i reward, S_i = state, and a_i = action, is represented by the action value function Q (s, a) in Q-learning. The best course of action is determined by choosing the action with the highest value max a Q (s a) at each time step given an optimal value function Q (s, a). Its foundation is the discovery of a function Q (s, a) leading to an estimation of the function value (Q-value). The utility of performing action an in-state S is denoted by the function Q (s, a). The best course of action is determined by choosing the action associated with each state's strongest correlation accumulated value given the function Q (s, a). Equation (3) is used to modify the function Q (s, a) to account for temporal differences.

$$Q(s_i,a_i) = Q(s_i,a_i) + \delta(u_{i+1} + Q_{max}(s_{i+1},a) - Q(s_i,a_i)) \quad (3)$$

If all following judgments were the best ones, (3) modifies Q (s, a) according to the present and anticipated reward. In this view, the function Q (s, a) tends to the function's ideal values. The Q-values can be utilized by the deep learning model to assess each option that is feasible in each state. The best choice is the one that gives the highest Q-value.

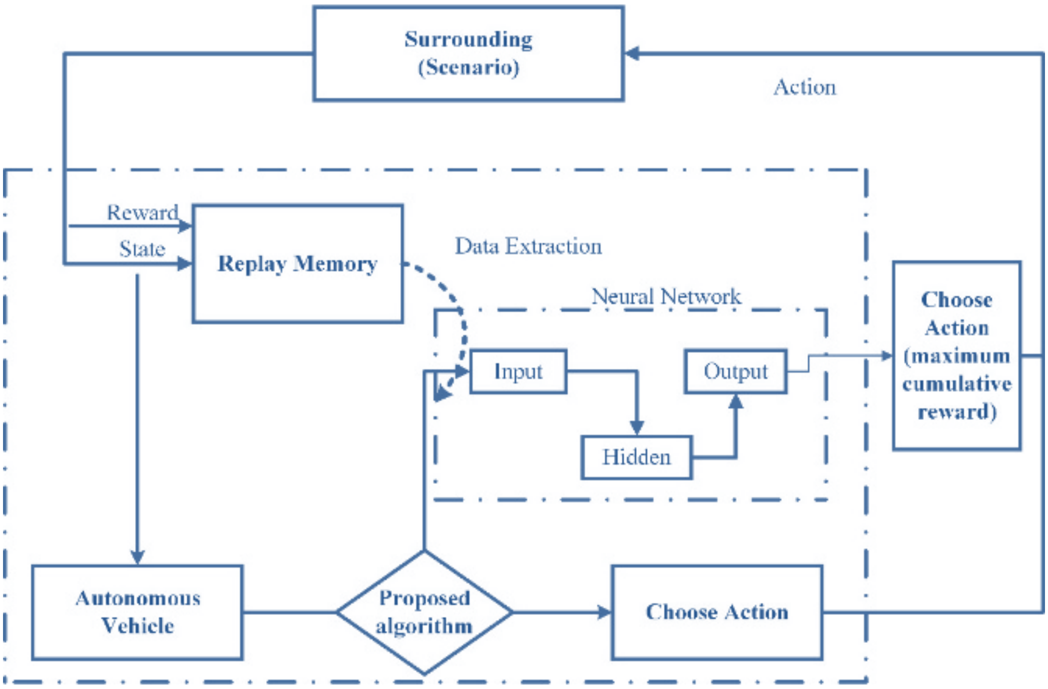

Fig. 3. Framework of intelligent system.

Algorithm 1 illustrates the complete model operational processes for this opinion based on the Q-learning algorithm.

ALGORITHM 1: DEEP Q-LEARNING ALGORITHM

```
learn_pol = 0.1
learn_rate = 0.1
epochs = 0
satisfied_obj = 0
While satisfied_obj ! =1:
# Environment data recollected (Speed, distance, state, reward, objective satisfied)
        state, reward, satisfied_obj, info = read_surrounding
        if (state not in q_table):
        else:
        action =np.argmax(q_table[state])
#Action = (accelerate, speed, distance)
#Apply action
        Control(action)
#Compute new q-value
        present_q = q_table [old_state, old_action]
        new_q = (1-alpha) *old_value + (sum of reward and gamma * max_q)
new_q = (1-learn_rate)*present_q+learn_rate*(sum of reward + discount max_q)

        q_table [state,action] = new_value
        past_state = state
        past_action = action
        epoch + =1
```

The environment's instant reward plus the highest value of Q for the new state attained make up the Q-value that results from the accomplishment of an action. Function T, which is influenced by the discount factor or variable g, determines the transition from one state to the other.

$$s_{i+1} \leftarrow I(s_i, a_i); Q(s_i, a_i) = u_{i+1} + \mu\, Q_{max}(s_{i+1});\; 0 \le \mu \le 1, \tag{4}$$

In (4) the Q-value would be upgrade using (5).

$$Q(s_i, a_i) = Q(s_i, a_i) + \delta(u_{i+1} + Q_{max}(s_{i+1}, a) - Q(s_i, a_i)) \tag{5}$$

The variable factor δ is used for setting the learning mechanism.

The changed Q function values are grouped as a table with details on the novel states and actions being investigated in the algorithm that is being provided. Thus, each column contains data regarding the worth of the acts, and each row represents a different state. Particularly, the value of acting from state s_m if the action is a_n is represented by table element (m, n). Table I is a Q-table that was created by employing the algorithm in any of the total states that it has learned. All the state-action combinations must be stored, which causes this Q-table to develop quickly.

Due to the need to contain all possible state-action combinations, this Q-table expands quickly.

TABLE I: STATE OF Q-TABLE

State	Action 1	Action 2	Action 3
s_1	[-0.38: -0.164]	[0.14: -0.1148]	[-0.28: -0.1368]
s_2	[0.08: -1.112]	[0.25: -0.415]	[-0.25:0.417]
s_3	[0.18: -0.119]	[-024: -0.134]	[0.22: -0.128]

C. Neural Network Mechanism on Q-learning

In contrast to conventional assessment, researchers substitute the Q-table in deep Q-learning with experience replay information using a neural network. The neural network must be taught while the vehicle is in motion if there are no prior experience training dataset sets. The neural network's value changes at every training stage. This paper lacks target values for the learning of neural networks procedure. A neural network has trouble resolving if researchers train it with a set of constantly shifting variables as the target value. Due to a feedback loop between the target value and the total value, the system might not always function properly. In order to finish error backpropagation algorithm and change the weights, researchers therefore employ two neural networks. To supply target values and gradually refine the neural network's weights, researchers adopt a slower-updating system [20].

As depicted in Fig. 4, those two neural network's functions. Among this the estimated value is generated using ass_net and this is denoted as q_ass and the other network is called as object_net and this helps to produce the target Q value and it is represented as q_object. These have precisely the same framework. The new weight is presented in the ass_net and get upgraded. The historical version of ass_net is known object_net, it keeps track of the ass_net previous values and upgrades from time to time. At the start of training, researchers initialize both neural networks with identical random weights. Researchers consider the disparity in output values produced by the two neural nets to be an error throughout learning and propagate it back to the. The mistake is reduced by altering each neuron's weight [21].

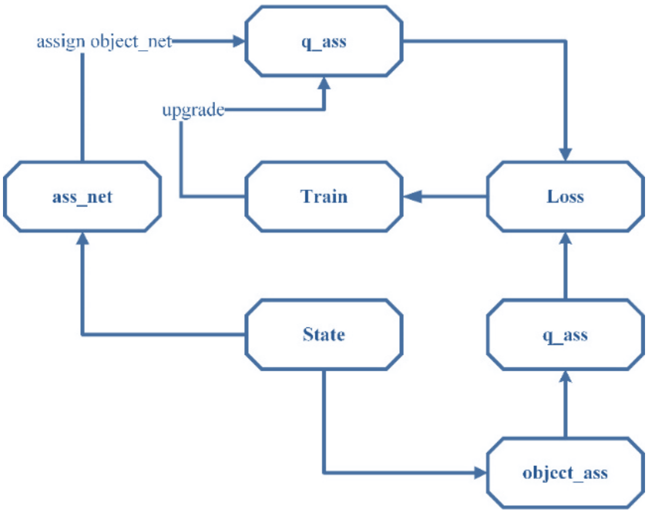

Fig. 4. Deep Q-learning neural network model.

D. Scenario

The research chose a situation of changing; the following Fig. 5 shows the high-level situation, the auto car starts in the center lane and its necessary to change the lane. In this issue, research provide our automobile two excellent possibilities for the movement. Lane follow is the first option, the lane change move is the second and lane change is the wait. Fig. 3 depicts the high-level organization of the proposed deep Q-learning trajectory planner. For the path planning process, the research considers the lane change, follow and wait. Let assume that the auto car either change it lane or follow the front car based on its speed, distance and acceleration. Given that X autonomous cars travel at a steady speed s on a road with length f, the entry route can be employed to regulate the density of traffic by regulating the number of vehicles on the highway. Consider there are u lanes, each with n vehicles, and the presence of vehicles in each lane. When the vehicles are in a lane, the spacing between them is as follows in (6).

$$d_{u,u-1}(t) = p_{i-1}(t) - p_i(t) - f_i \tag{6}$$

$$V(d) = \begin{cases} o, & d \leq d_{pt} \\ \frac{v_{max}}{2}\left[1 - \cos\left(\pi\left(\frac{d-d_{pt}}{d_{do}-d_{pt}}\right)\right)\right] & d_{pt} \leq d \leq d_{do} \\ v_{max} & d \leq d_{d0} \end{cases} \tag{7}$$

In (6) the distance between the two vehicle is denoted as (d), the length and position of the vehicle is denoted as f and p. The function connection among predicted speed and distance is given by the velocity range V(d) mentioned in (7). d_{d0} is the fair distance among automobiles in a scarce atmosphere, whereas d_{pt} is the safe distance among vehicles in congested surroundings.

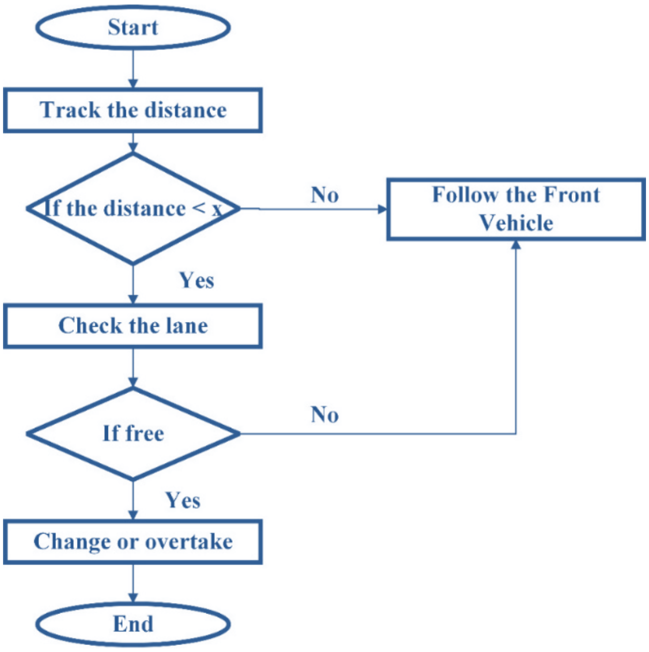

Fig. 5. Flowchart on Proposed method.

Based on the Fig. 5, by using the open CV data the research starts by monitoring the distance between the auto car and the front vehicle where the required distance is mentioned as x if the distance between the front vehicle is greater than the required distance between the auto car and front vehicle checks the lane whether it is free to go or not. If the lane is free the auto car overtakes the front car or change the lane. If the lane is not free or with traffic it follows the front car. The process continues until the autonomous vehicle reach the desired destination.

E. *Trajectory Planning*

The separation of path into two high-level alternatives (lane follow, stop, and lane change) aids the auto-car in learning a policy for both the high-level and low-level trajectory planners. After choosing the high-level option, the low-level trajectory planner chooses the last waypoint according to the network policy. The epsilon-greedy technique is used to base the choice on information about the condition of the car. In order to guarantee a smooth sub-trajectory, the target speed for the auto-car is determined after the chosen end waypoint utilizing the greatest acceleration it is capable. The PID controller is then provided the values for the target speed and final waypoint, which produces both horizontal and vertical regulation. A full trajectory, which includes lane follow, wait and lane change operations, is made up of all of these sub-trajectories.

F. *State Space*

In order to postulate the state space for the proposed model, the author employed the details of auto Car (Z), object car (X), object car (Y) based on the surrounding or case taken. This could be presented in the structure assessed below. The secure range of the auto car is $a \geq 15$ for wait and the moving vehicle is about $a \ \geq 8$, in which a is denoted as the distance between the auto car and the other vehicle. The structure of the state data is presented below in (8).

$$s = [v_z, lan_{ida}, v_f, d_{cr}, lan_{idb}] \tag{8}$$

In (8); the velocity of the auto car is defined as v_z and the lane -ID of the auto car and the target car is denoted as lan_{ida}. The velocity of the target car is denoted as v_f and the distance between the auto car and the object vehicle is denoted as d_{cr}.

G. *Reward and Action*

Five actions with a set reward are taught to the neural network. The network employs a 32-piece mini-batch that was trained using a 0.001 learning rate. The network setups for the training are shown in Table II. The reward policy functions in Q-learning as a fitness function from the perspective of an optimization model. Based on the present assert of the vehicle at the time, a double award system was implemented.

TABLE II: CONFIGURATION OF PARAMETER

Components	Value
Learn rate	0.01
δ	0.98
μ	11 → 0.1
Replay	15000
Batch size	32

One of the important components of the reinforcement-based learning method is action specification. Keep Moving (reward=speed/5), Left (reward=-0.6), Right (reward=-0.2), Accelerate (reward=+1), and Brake (reward=-0.4) are the 5 actions they select for the agent to be taught with Deep Q-learning. The agent's primary course of action is to continue traveling down the road without doing anything. A terrible reward with a value of 6 is given for a hit. It must learn how-to pick-up speed when there are no other vehicles or agents in front of it and slow down (brake) since there are presented.

A separate high-level choices reward and low-level trajectory choice reward make up the training heritage. The following fines and bonuses are applied at each time step while considering the autocar (z), moving car (x), and moving car (y). If the auto-car selects the incorrect high-level option or the incorrect low-level trajectory, it will be fined individually. If the chosen trajectory results in the sub-goal not being successfully completed, in this case collision against one of the target automobiles, the low-level option is penalized. Moreover, low-level decisions were punished if they were not necessary in order to plan safer and smoother trajectories. For instance, under option 1, selecting the low-level wait selection unintentionally results in a penalty. As the safe chase distance from other vehicles shrinks, the auto-car gets penalized more.

IV. Results and Discussion

The simulation results for the testing at constant speed show that the suggested strategy is workable and efficient during cruising. The proposed method has the capacity to adapt to unpredictable driving situations, according to the simulation findings for the varied speed testing. Additionally, it is clear from comparing various techniques that the classic acceleration undergoes changes during operations. However, the suggested approach prevents alterations through its learning process and interaction capacity, and this discovery also enhances the vehicle's comfort and adaptability. In this study, tests were run using a collection of realistic driving data that included environmental information and algorithms for forecasting vehicle speed.

The best matrix Q-value is then generated in a straightforward static scenario using the -Q-learning procedure. Finally, trajectory planning in various dynamic situations is done using the best matrix Q value. Unlike previous methods, Q-learning may compute lot of information routes, preventing the failure to quickly switch tactics in an eventuality. Q-learning algorithm provides greater performance and cheaper cost comparing to k-shortest method. There are fewer tests required, expenditures are cut, and the random selection is diminished. Table III displays the deep-learning-based routing path when the road condition is similar to Fig. 5. The likelihood of selecting the primary path 1-3-5 declines as grows, while the likelihood of selecting other paths rises. Fig. 6 illustrates how small changes in won't significantly affect the outcomes μ but will highlight them. The right value should be chosen in accordance with the research observations without altering μ the outcomes. This research uses $\mu = 0:18$ because it cannot be too large because the benefit would be diminished and should not be too tiny since it might diminish the appearance of characteristics.

TABLE III: PATH SELECTION

Trajectory planning	μ	Time 10	Time 50
1-3-5	0.01	1	45
Other	0.01	0	4
1-4-5	0.18	9	43
43other	0.20	0	40
1-3-5	0.1	9	35

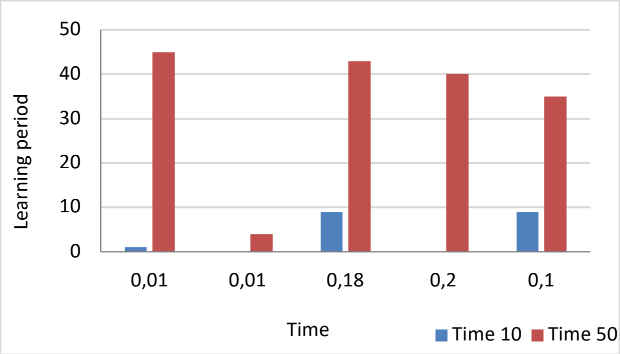

Fig. 6. Trajectory Planning Period.

Fig. 7, Fig. 8 and Fig. 9 show the learning environment with front car and the auto car. The trajectory planning section explains the process of how the autonomous car responds to the scenario mentioned as change, wait or follow.

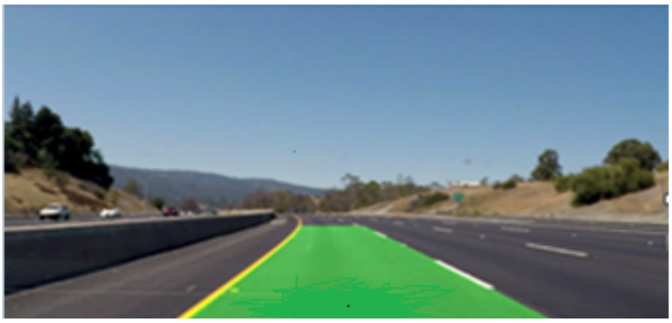

Fig. 7. Learning Environment.

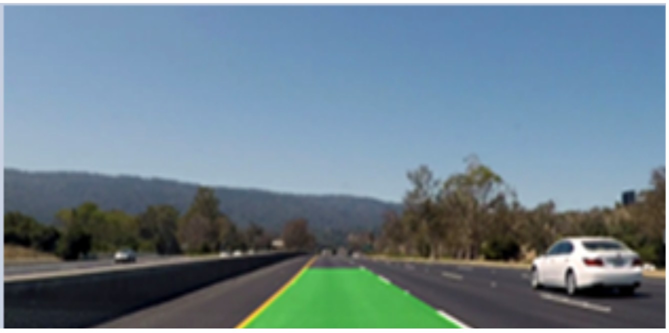

Fig. 8. Front Car.

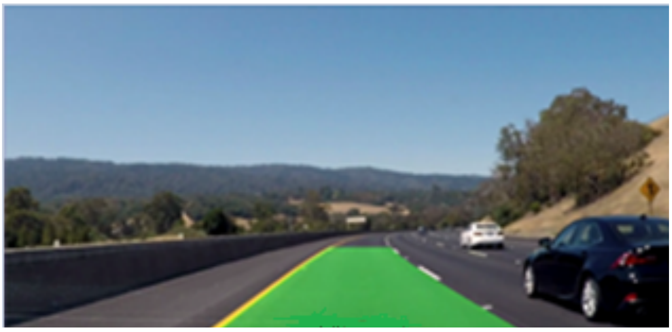

Fig. 9. Auto Car.

Fig. 10 represents the traffic-free zone in which the vehicle moves without any distraction this goes to the condition follow which means the lane is empty and the vehicle follows based on the given condition. Suppose the speed of the car is low when compared to the autocar the vehicle gets the condition overtake as in Fig. 11.

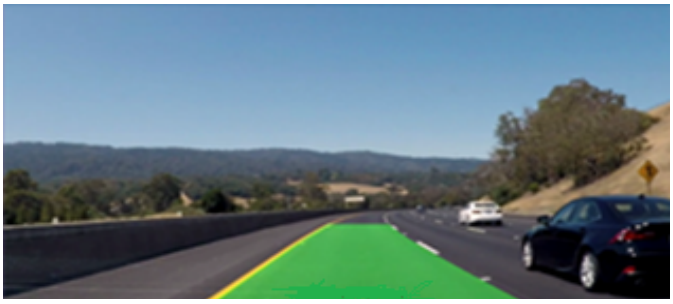

Fig. 10. Traffic Free Zone.

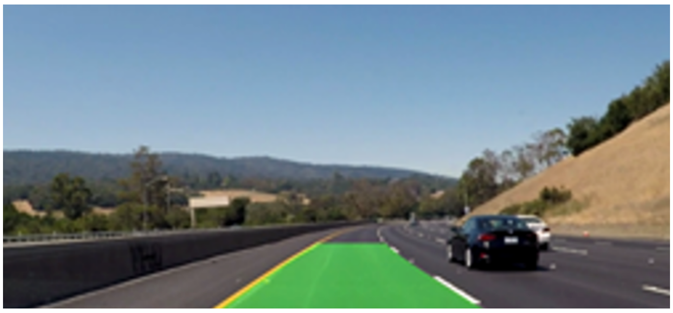

Fig. 11. Overtake/Change Lane.

This paper's main contribution is the creation of practical learning for consecutive and autonomous cars using instances in a virtual environment. In the trials, the deep Q-learning algorithm was used to design a guided policy in terms of tangible learning, which benefited the human perceptual. It is possible to achieve trajectories that match specific driving styles by changing the weights of the optimal solution.

Given that it has been shown to be jerk-optimal, the transverse motion was planned. Its coefficients were derived from the terminal values, which were based on the behaviour layer's choice of operating mode (velocity keeping, following/stopping, or merging). Changes in vary substantially and time length throughout the planning horizon was used to sample lateral and longitudinal trajectories. Following their integration, the sampling trajectory was assessed using a cost function that quantifies jerk, angular displacement, departure from the centerline, and intended speed.

V. Conclusion

In the context of practical applications for autonomous driving, reinforcement learning remains an evolving field. Despite a few commercially successful implementations, there is a dearth of literature and substantial public databases. Our motivation to codify and arrange RL applications for autonomous driving came from this. In this paper, we present a deep Q-learning and neural network-based trajectory planning technique for autonomous cars. The findings demonstrate that the suggested approach reduces convergence time and guarantees secure and efficient PID waypoint tracking. The suggested paradigm still has certain drawbacks. First off, due to its dynamic features, this model is better suited for conventional passenger vehicles. The reward functions also don't provide a thorough characterization of the many subsequent conditions. Furthermore, the model's durability is not confirmed. Future work can enhance incentive functions by taking comfort into account. A test using an actual car could confirm the model's stability.

References

[1] Huang Y, Ding H, Zhang Y, Wang H, Cao D, Xu N, Hu C. A Motion Planning and Tracking Framework for Autonomous Vehicles Based on Artificial Potential Field Elaborated Resistance Network Approach. IEEE Transactions on Industrial Electronics, 2020; 67(2): 1376–1386. https://doi.org/10.1109/tie.2019.2898599.

[2] Zhang Y, Zhang J, Zhang J, Wang J, Lu K, Hong J. A Novel Learning Framework for Sampling-Based Motion Planning in Autonomous Driving. *Proc. AAAI Conf. Artif. Intell.*, 2020; 34(01): 1202–1209. doi: 10.1609/aaai.v34i01.5473.

[3] Huang Y, Wang H, Khajepour A, Ding H, Yuan K, Qin Y. A Novel Local Motion Planning Framework for Autonomous Vehicles Based on Resistance Network and Model Predictive Control. *IEEE Trans. Veh. Technol.*, 2020; 69(1): 55–66. doi: 10.1109/TVT.2019.2945934.

[4] Li J, Chen Y, Zhao X, Huang J. An improved DQN path planning algorithm. *J. Supercomput.*, 2022; 78(1): 616–639. doi: 10.1007/s11227-021-03878-2.

[5] Chen C, Jiang J, Lv N, Li S. An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge. *IEEE Access*, 2020; 8: 99059–99069. doi: 10.1109/ACCESS.2020.2998015.

[6] Abdi A, Ranjbar MH, Park JH. Computer Vision-Based Path Planning for Robot Arms in Three-Dimensional Workspaces Using Q-Learning and Neural Networks. *Sensors*, 2022; 22(5): 1697. doi: 10.3390/s22051697.

[7] Kiran BR, Sobh I, Talpaert V, Mannion P, Sallab AAA, Yogamani S, Perez P. (2022). Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems,* 2022; 23(6): 4909–4926. https://doi.org/10.1109/tits.2021.3054625.

[8] Song R, Liu Y, Bucknall R. Smoothed A* algorithm for practical unmanned surface vehicle path planning. *Appl. Ocean Res.*, 2019; 83: 9–20. doi: 10.1016/j.apor.2018.12.001.

[9] Gu T, Snider J, Dolan JM, Lee JW. Focused Trajectory Planning for autonomous on-road driving. *2013 IEEE Intelligent Vehicles Symposium (IV)*. https://doi.org/10.1109/ivs.2013.6629524.

[10] Kebria PM, Khosravi A, Salaken SM, Nahavandi S. Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEECAA J. Autom. Sin.*, 2020; 7(1): 82–95. doi: 10.1109/JAS.2019.1911825.

[11] Fayjie AR, Hossain S, Oualid D, Lee DJ. Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment. *in 2018 15th International Conference on Ubiquitous Robots (UR)*, Honolulu, HI; Jun. 2018: 896–901. doi: 10.1109/URAI.2018.8441797.

[12] Altche F, Polack P, de La Fortelle A. High-speed trajectory planning for autonomous vehicles using a simple dynamic model. *in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Oct. 2017: 1–7. doi: 10.1109/ITSC.2017.8317632.

[13] Li G, Yang Y, Qu X, Cao D, Li K. A deep learning based image enhancement approach for autonomous driving at night. Knowledge-Based Syst., 2021; 213: 106617.

[14] Kawasaki A, Seki A. Multimodal trajectory predictions for autonomous driving without a detailed prior map. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021: 3723–3732.

[15] Altché F, Polack P, de La Fortelle A. High-speed trajectory planning for autonomous vehicles using a simple dynamic model. *in 2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*; 2017, pp. 1–7.

[16] Batkovic I, Zanon M, Ali M, Falcone P. Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users. *in 2019 18th European Control Conference (ECC)*; 2019, pp. 256–262.

[17] Fujiyoshi H, Hirakawa T, Yamashita T. Deep learning-based image recognition for autonomous driving. *IATSS Res.*, 2019; 43(4): 244–252.

[18] Cui H, Radosavljevic V, Chou FC, Lin TH, Nguyen T, Huang TK, Schneider J, Djuric N. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *in 2019 International Conference on Robotics and Automation (ICRA)*, 2019: 2090–2096.

[19] Cho RLT, Liu, JS, Ho MHC. The development of autonomous driving technology: perspectives from patent citation analysis. *Transp. Rev.*, 2021; 41(5): 685–711. doi: 10.1080/01441647.2021.1879310.

[20] Zhu S, Aksun-Guvenc B. Trajectory Planning of Autonomous Vehicles Based on Parameterized Control Optimization in Dynamic on-Road Environments. J. Intell. Robot. Syst., 2020; 100(3–4): 1055–1067. doi: 10.1007/s10846-020-01215-y.

[21] Naveed KB, Qiao Z, Dolan JM. Trajectory Planning for Autonomous Vehicles Using Hierarchical Reinforcement Learning. *arXiv*, 2020. [Online]. Retrieved from: http://arxiv.org/abs/2011.04752.