**Autonomous Vehicles Lane change Prediction and Obstacle Prediction using Q-Learning**

**19CSE495 PROJECT PHASE 1 REPORT**

**Submitted by**

|  |  |
| --- | --- |
| **ROLL NUMBER** | **NAME** |
| CB.EN.U4CSE20148 | POLA YOGESWARA REDDY |
| CB.EN.U4CSE20644 | PALDUGU SUCHAL |
| CB.EN.U4CSE20649 | PUTTA JYOTHSNA |
| CB.EN.U4CSE20653 | SANGIREDDY SAINATH REDDY |

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641112

DECEMBER 2023

AMRITA VISHWA VIDYAPEETHAM

*TABLE OF CONTENTS*

|  |  |
| --- | --- |
| SI.No | Title |
| 1. | Abstract |
| 2. | Introduction  2.1 Need for the project  2.2 Problem description  2.3 Problem Use case  2.4 Problem statement/ Research Objectives |
| 3. | Literature Survey |
| 4. | Proposed Methodology |
| 5. | Design: Architecture diagram/Flow chart/ DFD |
| 6. | Explanation of Algorithm /techniques |
| 7. | Prototype of 30% of Design |
| 8. | Novelty of the solution |
| 9. | Work plan and publication plan for Phase-2 |
| 10. | Abstract Of the Article for publication |
| 11. | Conclusion |
| 12. | References |

**1. Abstract:**

Lane change and obstacle prediction are critical aspects of autonomous vehicle technology, where an agent needs to make optimal decisions for changing lanes while anticipating and avoiding obstacles on the road. Reinforcement learning, particularly Q-learning, offers significant advantages over traditional supervised and unsupervised learning methods in the context of lane change and obstacle prediction. Among Q-learning algorithms, Q-Learning is a well-established technique that can greatly assist in addressing the challenges of lane change and obstacle prediction. These algorithms play a crucial role in ensuring the safe navigation of autonomous vehicles in dynamic environments. Lane change and obstacle prediction algorithms are essential for autonomous vehicles to navigate effectively in traffic. These algorithms enable self-driving cars to make informed decisions regarding lane changes and anticipate potential obstacles or hazards, reducing the risk of collisions. The advancement of autonomous vehicles has brought about the need for efficient and safe lane change maneuvers, as well as accurate obstacle prediction algorithms. Q-learning, a reinforcement learning technique, has emerged as a promising approach to address these challenges.

The lane change process involves the autonomous vehicle making decisions on when and how to change lanes while considering various factors such as traffic conditions, speed, and safety. Q-learning algorithms enable the vehicle to learn optimal lane change policies through trial and error, maximizing long-term rewards while minimizing risks. Obstacle prediction is crucial for ensuring the safety of autonomous vehicles. By utilizing Q-learning, the vehicle can learn to predict the movements and intentions of surrounding obstacles such as other vehicles, pedestrians, and cyclists. This predictive capability allows the autonomous vehicle to proactively plan its actions and avoid potential collisions.

Recent advancements in Q-learning techniques, such as deep Q-networks (DQNs) and double Q-learning, have further improved the accuracy and efficiency of autonomous vehicle lane change and obstacle prediction systems. These techniques leverage deep neural networks to approximate the Q-values, enabling more complex decision-making processes.

Furthermore, the integration of real-time sensor data, such as LiDAR, radar, and cameras, enhances the performance of Q-learning-based systems. By continuously updating the state-action values based on the sensor inputs, the autonomous vehicle can adapt to dynamic environments and make informed decisions. Autonomous vehicle lane change and obstacle prediction using Q-learning have become critical research areas in the field of autonomous driving. The combination of reinforcement learning techniques, advanced algorithms, and real-time sensor data enables autonomous vehicles to perform safe and efficient lane changes while accurately predicting and avoiding obstacles. These advancements contribute to the overall goal of achieving fully autonomous and reliable transportation systems.

**2. Introduction:**

Autonomous vehicles (AVs), sometimes known as self-driving automobiles, can sense their surroundings and move on their own. The development and testing of self-driving automobiles, trucks, and other vehicles is the current trend in AVs. Companies and researchers are advancing the technology and running tests in various settings. Self-driving cars are not created intelligent; they are constructed using a certain algorithm. The algorithms that direct and manage their motions are constantly improving as the technology becomes more prevalent in automobiles like Tesla. The architecture of autonomous cars includes executive control, decision planning, and environmental awareness. Through sensors and highly accurate maps, the car gathers environmental information from its surroundings. Through trial and error, an agent learns how to behave in each environment using the machine learning process known as Q- learning (QL).

Autonomous vehicles have made significant progress in recent years, transforming the transportation industry. One important aspect of autonomous driving is the ability of these vehicles to change lanes safely and efficiently. Lane changes require intelligent decision-making based on factors like traffic conditions, speed, and safety. Additionally, accurately predicting obstacles is crucial for autonomous vehicles to navigate their surroundings without collisions. To tackle these challenges, researchers have turned to a type of machine learning called Q-learning. This approach allows autonomous vehicles to learn optimal lane change strategies and predict the movements of obstacles. Q-learning involves making decisions through trial and error, maximizing long-term rewards while minimizing risks.

The main goal of using Q-learning for autonomous vehicle lane changes and obstacle prediction is to improve safety and efficiency. By utilizing Q-learning algorithms, autonomous vehicles can learn from past experiences and develop effective lane change strategies. This includes considering factors like the speed and path of nearby vehicles, identifying gaps in adjacent lanes, and following traffic rules. Ensuring safe lane changes involves several important factors. First, autonomous vehicles need accurate perception capabilities through sensors like LiDAR, radar, and cameras. These sensors provide real-time data about the surrounding environment, allowing the vehicle to detect and track nearby vehicles, pedestrians, and obstacles.

Furthermore, autonomous vehicles must have reliable decision-making algorithms that can evaluate the current traffic situation and determine the best time and method for a lane change. This includes considering factors like the distance and speed of nearby vehicles, using appropriate signals, and merging smoothly into the desired lane.To prevent collisions during lane changes, safety measures must be in place. Autonomous vehicles must constantly monitor their surroundings for potential hazards and react quickly to avoid accidents. This involves predicting the future movements of nearby vehicles and adjusting the lane change accordingly.

Additionally, communication between autonomous vehicles and infrastructure plays a crucial role in ensuring safe lane changes. Systems that enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication provide real-time information about traffic conditions, road hazards, and potential obstacles. This helps autonomous vehicles make more informed decisions during lane changes.

Compared to the other ML algorithms , Using Q-learning for autonomous vehicle lane changes and obstacle prediction is vital for improving safety and efficiency in autonomous driving. Q-learning offers several advantages over other machine learning algorithms for autonomous vehicle lane change and obstacle prediction. It is a model-free learning method, making it suitable for complex and dynamic driving scenarios. By allowing autonomous vehicles to learn through trial and error, Q-learning enables them to refine their decision-making process over time. The focus on maximizing long-term rewards ensures that lane change and obstacle prediction decisions consider overall safety and efficiency. Q-learning is adaptable to changing environments and can handle uncertainties, making it robust in real-world driving conditions. Its flexibility in handling both discrete and continuous action spaces allows for fine-grained decisions during lane changes. With efficient scalability, Q-learning can handle large state and action spaces, considering multiple factors simultaneously. Integration with deep learning enables Q-learning to handle high-dimensional input data, such as images or point clouds from sensors. Real-time decision-making, generalization across environments, adaptation to driver preferences, and continuous learning further enhance the effectiveness of Q-learning in autonomous driving. Its ability to balance exploration and exploitation, low computational requirements, robustness to noisy sensor data, interpretability, transfer learning capabilities, and handling of non-linear relationships make Q-learning a preferred choice for autonomous vehicle lane change and obstacle prediction tasks.

**2.1 Need for the project:**

There are several reasons why traffic accident cases are reported today, some of which include:

* Collisions caused by obstructions
* unexpected lane shifts
* driving technique and response time
* sudden braking

The difficulties that were raised cause mishaps and injuries. By removing human mistake, AVs have the potential to lower these accidents. AVs can learn from their experience and improve their Lane change detection approach based on incentives and penalties thanks to Q-learning algorithms like q-learning. AVs can adapt to changing circumstances, avoid obstacles, and select the most direct route to their objective by employing Q-learning. Traffic accidents occur due to various reasons, including collisions caused by obstacles, sudden lane shifts, driver errors, and abrupt braking. These factors often result in accidents and injuries. However, the emergence of autonomous vehicles (AVs) offers a potential solution by eliminating human errors.

AVs have the ability to learn from their experiences and continually enhance their ability to detect and execute lane changes. This is achieved through the use of Q-learning algorithms, which enable AVs to improve their decision-making based on rewards and penalties. By employing this algorithm, AVs can adapt to changing road conditions, avoid obstacles, and choose the most efficient routes.

One of the main advantages of using Q-learning in AVs is the optimization of lane change detection. By considering incentives and penalties, AVs can make more informed decisions when changing lanes. This helps prevent collisions caused by obstructions as AVs can anticipate and respond to potential obstacles in a timely manner.

Furthermore, unexpected lane shifts can be better handled by AVs using Q-learning algorithms. Through continuous learning and adaptation, AVs become more proficient at predicting and responding to sudden lane changes by other vehicles. This significantly reduces the risk of accidents caused by abrupt lane shifts.

Moreover, driving technique and response time are crucial for accident prevention. AVs equipped with Q-learning algorithms can improve their driving techniques over time by analyzing rewards and penalties associated with different driving behaviors. This allows them to optimize their response time and make more efficient decisions on the road. Sudden braking is another factor contributing to accidents. AVs utilizing Q-learning algorithms can learn to anticipate situations requiring sudden braking and adjust their driving behavior accordingly. This minimizes abrupt stops and reduces the risk of rear-end collisions.

By incorporating Q-learning algorithms in AVs has the potential to address various causes of traffic accidents. By eliminating human errors and enabling continuous learning, AVs can enhance their lane change detection, handle unexpected lane shifts, optimize driving techniques and response time, and mitigate sudden braking incidents. This ultimately leads to a safer and more efficient road environment for everyone.

**2.2 Problem Description:**

* The problem addressed in this project revolves around the critical aspect of Lane change detection in Autonomous Vehicles (AVs) . To ensure their safe and efficient operation in dynamic environments, it is imperative to develop robust High refined policy that can navigate AVs from avoiding obstacles and potential collisions .
* This project focuses on leveraging Q-Learning (QL) techniques, specifically Q-Learning , to Make dynamic decisions in continuous lane change detection in AVs. RL enables AVs to learn from their experiences , refine their decision-making processes, and optimize strategies based on rewards and penalties . By incorporating QL into AVs , the goal is to empower them to navigate complex environments , avoid obstacles, respond to emergencies, and ultimately contribute to safer and more efficient transportation systems .

**2.3 Problem Use case:**

1) Lane Change Optimization:

**Scenario:** Autonomous vehicles often face challenges in optimizing lane changes for efficiency and safety. The goal is to develop a system that uses machine learning, such as Q-learning, to determine optimal times for lane changes, considering factors like traffic flow, vehicle speed, and road conditions.

**Use Case Steps:**

* **Real-Time Traffic Monitoring:**

The autonomous vehicle continuously monitors real-time traffic conditions using

sensors.

* **Q-Learning Decision-Making:**

Q-learning processes the traffic data, learning optimal times and conditions for lane

changes.

* **Dynamic Decision Execution:**

The vehicle dynamically executes lane changes, minimizing disruptions to traffic flow

and maximizing efficiency.

#### 2) Obstacle Prediction for Collision Avoidance:

**Scenario:** To enhance safety, autonomous vehicles need to predict the movement of surrounding obstacles and proactively avoid collisions. Q-learning can be employed to develop an obstacle prediction system.

**Use Case Steps:**

* **Obstacle Tracking:**

The vehicle's sensors track the movement of nearby obstacles, including other vehicles, pedestrians, and cyclists.

* **Q-Learning for Prediction:**

Q-learning processes the obstacle movement data, learning to predict future positions and actions.

* **Proactive Maneuvering:**

The autonomous vehicle uses the learned predictions to proactively plan and execute maneuvers to avoid potential collisions.

#### 3) Adaptive Speed Control:

**Scenario:** Optimizing vehicle speed based on real-time conditions is crucial for both efficiency and safety. Q-learning can be utilized to develop an adaptive speed control system.

**Use Case Steps:**

* **Real-Time Traffic and Road Condition Monitoring:**

Sensors provide information on traffic density, road conditions, and obstacles.

* **Q-Learning Speed Control:**

Q-learning processes the data to determine the optimal speed for the current conditions, considering safety and efficiency.

* **Dynamic Speed Adjustment:**

The vehicle adjusts its speed in real-time based on the Q-learning recommendations, ensuring smooth and safe navigation.

#### 4) Traffic Signal Optimization:

**Scenario:** Autonomous vehicles can contribute to traffic flow optimization by adapting to traffic signal patterns. Q-learning can be employed to develop a system that optimizes interactions with traffic signals.

**Use Case Steps:**

* **Traffic Signal Recognition:**

The vehicle's sensors recognize and interpret traffic signal patterns.

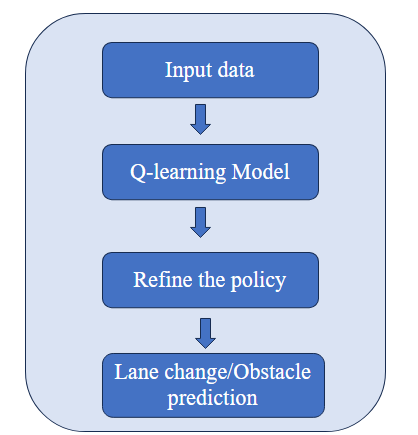
* **Q-Learning for Signal Interaction:**

Q-learning processes the signal data, learning optimal interaction strategies.

* **Adaptive Signal Response:**

The autonomous vehicle adapts its speed and approach based on learned Q-values, contributing to overall traffic flow optimization.





**2.4 Problem statement:**

To develop an optimal algorithm for Lane change prediction and obstacle prediction using network of Autonomous vehicles.

**3. Literature Survey:**

|  |  |  |  |
| --- | --- | --- | --- |
| TITLE | PROBLEM STATEMENT | ADVANTAGES | DISADVANTAGES |
| Autonomous Vehicle Path Planning using Q-Learning [1]  **REF#YEAR:2021** | To investigates the possibility of using Q-learning for solving the local path planning problem with obstacle avoidance. The goal is to find an optimal path for the vehicle to reach its target while avoiding obstacles in the environment. | * The research provides insights into the practical application of Q-Learning for autonomous vehicle path planning, contributing to the development of reliable navigation systems. * Q-Learning can also learn path planning in a randomized environment with obstacles and variable starting positions. It showed improvement in avoiding obstacles and reaching the goal, demonstrating its adaptability to different environments. * Q-Learning can learn the optimal path in a static environment with minimal movement steps, showcasing its ability to find the shortest path from any starting position. | * The training phase of Q-Learning requires exponentially increasing training time based on the system's state space. * In certain configurations, the vehicle can get stuck between obstacles with no feasible path or solution, leading to a failure in finding a path to the target. * The paper does not discuss potential limitations or challenges in implementing Q-Learning in real-world autonomous vehicles, such as dealing with dynamic obstacles or real-time decision-making |
| Reinforcement Learning for Autonomous Vehicles [2]  **REF#YEAR: 2018** | Autonomous vehicle control presents a significant challenge for artificial intelligence and control theory. The act of driving is best modelled as a series of sequential decisions made with  occasional feedback from the environment | * Utilizing stored instances of past observations as value estimates in reinforcement learning is an effective and practical means of controlling dynamical systems like autonomous vehicles. * The research explores the application of reinforcement learning techniques in solving complex control problems. * The use of prioritized sweeping in reinforcement learning can improve the estimation of expected values. | * Reinforcement learning methods may fall short in environments requiring continual operation and with continuous state and action spaces, such as driving. * The process of reinforcement learning utilizing stored instances of past observations as value estimates may have limitations in terms of scalability and efficiency in controlling dynamical systems like autonomous vehicles. |
| Highway Traffic Modelling and Decision Making for Autonomous Vehicle Using Reinforcement Learning [3]  **REF#YEAR:** 2018 | To model the decision-making problem of autonomous vehicles in traffic using reinforcement learning techniques. | * The inclusion of stochastic driving behaviours of other vehicles in traffic enhances the realism of the model and enables the autonomous vehicle to make informed decisions in dynamic environments. * The consideration of cornering strategies and the comparison and analysis of different driving strategies further contribute to the understanding and improvement of autonomous vehicle decision-making in highway traffic | * The paper does not address the potential limitations or uncertainties in accurately modelling the stochastic behaviours of other vehicles in traffic. * The paper does not provide a detailed evaluation of the proposed model's performance in real-world scenarios. * The paper does not discuss the computational complexity or scalability of the proposed model. |
| Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment [4]  **REF#YEAR: 2018** | To develop a system that can enable self-driving cars to navigate through complex urban environments, making decisions in real-time based on sensor inputs. | * The approach uses two types of sensor data, camera and laser sensors, as input for the DRL algorithm. * The paper presents a cost-efficient high-speed car prototype capable of running the DRL algorithm in real-time. * The design of the car prototype includes a camera and a Hokuyo Lidar sensor in the front, which provides a combination of visual and depth information for accurate perception | * The paper does not discuss the limitations of using Deep Reinforcement Learning for autonomous navigation and obstacle avoidance in a real-world urban environment. * The paper does not provide any analysis or evaluation of the performance of the proposed approach. * The paper does not provide any comparison or benchmarking with other existing approaches. |
| **Analysis of Reinforcement Learning in Autonomous Vehicles [5]**  **REF#YEAR: 2022** | Aims to review the reinforcement learning techniques used by these companies and propose alternative solutions to the challenges faced by most autonomous vehicles. | * Provides an analysis of reinforcement learning techniques and algorithms used by companies like Waymo, Tesla, and GM in autonomous vehicles. * Proposes alternative solutions to the problems faced by most autonomous vehicles. * Gives a basic understanding of the reward algorithm function used in autonomous vehicle projects, including lower-end systems and more advanced systems like Tesla | * The alternative solutions proposed by the paper may not address all the problems faced by autonomous vehicles, as it does not consider the specific use-cases and challenges of each company or vehicle. * The paper does not provide a detailed examination of the reward algorithm function used in autonomous vehicle projects, as it only gives a basic understanding and does not explore the more diverse and complex reward systems used by companies like Tesla |
| Research on Path Planning and Tracking Control of Autonomous Vehicles Based on Improved RRT\* and PSO-LQR[6]  **REF#YEAR: 2023** | To Develop a path planning and tracking control framework for autonomous vehicles based on an upgraded RRT\* and Particle Swarm Optimization Linear Quadratic Regulator (PSO-LQR) and optimize the path search speed and iteration number of the RRT\* algorithm | * The introduction of an improved artificial potential field method into the RRT\* algorithm improves its convergence speed. * Path optimization is achieved through path pruning based on vehicle steering angle constraint and the cubic B-spline algorithm, resulting in curvature continuous paths that conform to precise vehicle tracking. * Simulation results show that the improved RRT\* algorithm optimizes path search speed and iteration number. | * The paper does not provide a detailed analysis of the trade-offs between the improved RRT\* algorithm and other variants, such as RRT\*-Connect or RRT\*-Smart, which could provide insights into the strengths and weaknesses of different approaches. * The paper does not discuss the limitations or potential challenges of implementing the proposed framework in real-world autonomous vehicles, such as computational requirements, sensor limitations, or environmental uncertainties |
| Autonomous Driving using Safe Reinforcement Learning by Incorporating a Regret-based Human Lane-Changing Decision Mode [7]  **REF#YEAR: 2020** | How to enable autonomous vehicles (AVs) to safely and efficiently maneuver in mixed traffic where human-driven vehicles also exist. | * By using the regret-based human decision-making model, the number of collisions in training and testing is reduced to zero, while maintaining training accuracy * The proposed framework for SafeRL integrates a safety supervisor that uses the predicted human decisions to check if an action is safe, resulting in zero collisions during training and implementation | * The deep Q-network tested on autonomous driving in two-lane traffic still resulted in collisions, indicating that the AVs struggle to estimate the intentions of other vehicles. * The RL algorithm needs to reset the simulation whenever collisions happen, causing unstable training |
| Q-learning with Long-term Action-space Shaping to Model Complex Behavior for Autonomous Lane Changes [8]  **REF#YEAR: 2021** | Optimization of complex behavior for autonomous lane changes in autonomous driving applications. | * The algorithm is compared to reward shaping and Lagrangian methods in the application of high-level decision making in autonomous driving, considering rules for safety, keeping right, and comfort. * The method is trained and evaluated in both an open-source simulator (SUMO) and the real-world HighD dataset, demonstrating its real-world applicability and simplicity | * The results of reward shaping and Lagrangian optimization suffer from high variance and are not consistent * The VPE agent, which uses the Valid Policy Extraction method, shows the worst performance among the agents considered, staying on the initial lane with no applied lane changes over all training runs. |
| Developing a Deep Q-Learning and Neural Network Framework for Trajectory Planning [9]  **REF#YEAR: 2022** | A deep Q-learning and neural network framework for trajectory planning in autonomous vehicles, addressing the challenges of generalization and safety assurance in path learning techniques | * The approach can investigate a path in an unknown situation with small iterations and faster convergence to an ideal strategic plan, outperforming traditional techniques. * The proposed deep Q-learning and neural network framework enhances the driving performance of autonomous vehicles in terms of energy consumption, ensuring smooth trajectories and reduced jerk. | * Further research is needed to optimize the performance in dynamic environments * The Q-table used in Q-learning expands quickly due to the need to contain all possible state-action combinations, which can lead to computational complexity |
| A New Approach for Tactical Decision Making in Lane Changing: Sample Efficient Deep Q Learning with a Safety Feedback Reward [10]  **REF#YEAR: 2022** | The novel deployment of the state of art Q learning method, namely Rainbow DQN, that uses a new safety driven rewarding scheme to tackle the issues in a dynamic and uncertain simulation environment. | * The proposed algorithm shows superior performance compared to the baseline algorithm in challenging scenarios, even with a relatively low number of training steps. * deployment of the Rainbow DQN method with a safety-driven rewarding scheme improves the performance and sample efficiency of the agent | Further research is needed to optimize the performance in dynamic environments |
| Combining Planning and Deep Reinforcement Learning in Tactical Decision Making for Autonomous Driving [11]  **REF#2020** | Develop a framework that combines planning and learning techniques to improve the performance of autonomous driving systems in making tactical decisions | * The combination of Monte Carlo tree search (MCTS) and deep reinforcement learning in the framework allows for better performance and improved decision-making capabilities in diverse and complex driving environments * The framework is flexible and can be easily adapted to different driving scenarios, making it versatile for various autonomous driving applications | * The framework relies on a continuous state space and a not directly observable state, which may introduce challenges in certain scenarios * The framework requires training samples, although it is more sample efficient compared to a Deep Q-Network (DQN) agent applied to a similar case |
| Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness [12]  **REF#YEAR: 2021** | To address the risk-insensitive decision-making problem in lane change scenarios for autonomous vehicles (AVs) and to develop a strategy with the minimum expected risk for safe driving | * The probabilistic-model based risk assessment method considers position uncertainty and distance-based safety metrics, providing a comprehensive evaluation of driving risk * The risk-aware decision-making algorithm, based on deep reinforcement learning, generates robust safe driving strategies and outperforms previous methods in terms of driving performance | * The framework should Improve the sampling probability function for vehicle position initialization to enhance the accuracy of risk assessment and decision-making. * Optimize the hyper-parameters in the proposed methods to further enhance the performance and efficiency of the decision-making framework |

**4. Proposed Methodology:**

1. Data Collection:

* Gather relevant sensor data such as LiDAR, cameras, GPS, and other sensors.
* Collect high-definition maps, traffic information, and dynamic obstacle data.

2. State and Action Space Definition:

* Define the state space to represent the vehicle's current position, orientation, speed, and surrounding environment.
* Specify the action space to include driving commands (throttle, brake, steering) or waypoints.

3. Q-Learning Model:

* Implement a Q-Learning model tailored for autonomous vehicle lane change and obstacle prediction.
* Create the Q-table structure and initialize Q-values.
* Define rewards and penalties based on lane change success, obstacle avoidance, and safety considerations.

4. Training the Q-Learning Model:

* Train the Q-Learning model using historical or simulated data.
* Use the collected data to update Q-values and optimize the model's decision-making process.

5. Lane Change Prediction:

* Utilize the trained Q-Learning model to predict the likelihood of a lane change at each time step based on the current state.
* Assign probabilities or confidence scores to different lane change actions (left, right, or no change).

6. Obstacle Prediction:

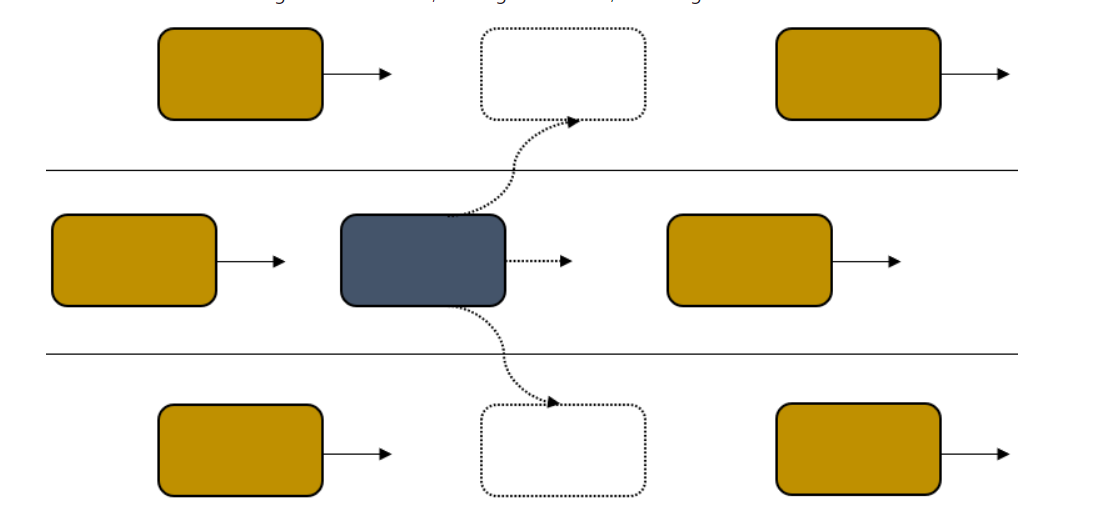
* Extend the Q-Learning model to predict and anticipate dynamic obstacles in the environment.
* Incorporate obstacle detection algorithms and sensor fusion techniques to provide accurate obstacle predictions.

7. Real-time Decision-making:

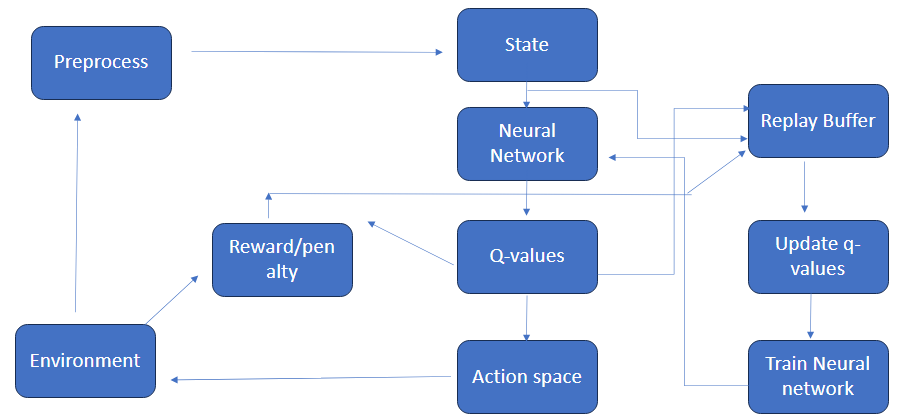
* Implement a real-time decision-making system that continuously evaluates the vehicle's state and adapts to changing road conditions.
* Utilize the lane change and obstacle prediction models to make informed decisions on when and how to change lanes to avoid obstacles.

8. Evaluation and Validation:

* Evaluate the performance of the autonomous vehicle's lane change and obstacle prediction system through extensive simulations and real-world testing.
* Measure metrics such as success rate of lane changes, obstacle avoidance accuracy, and overall system performance.
* Validate the system's safety and reliability under various scenarios and road conditions.



**5. Design: Architecture diagram/ Flow chart/ DFD**



**State:**

Represents the current state of the environment, which is typically an observation received from the environment.

**Neural Network:**

The state is fed into a neural network, which is the core component of the DQN. The neural network processes the input to produce Q-values for each possible action.

**Q-Values:**

The neural network outputs Q-values, which represent the expected cumulative future rewards for each action.

**Action Space:**

The agent selects an action based on the Q-values. This action is then executed in the environment.

**Environment:**

The selected action is applied to the environment, leading to a transition to a new state and the generation of a reward or penalty.

**Reward/Penalty**:

The environment provides a reward or penalty based on the agent's action and the new state.

**Replay Buffer:**

The experience tuple (state, action, reward, new state) is stored in a replay buffer. The replay buffer is a memory mechanism that helps break the temporal correlation in the training data.

**Update Q-Values:**

Periodically, the DQN retrieves a batch of experiences from the replay buffer to update the Q-values. This involves using the temporal difference error between the predicted Q-values and the target Q-values.

**Train Neural Network:**

The neural network is trained using the experiences sampled from the replay buffer. This involves adjusting the neural network's weights to minimize the difference between predicted and target Q-values.

**Preprocess:**

Before the next iteration, the new state may undergo preprocessing. This could involve any necessary transformations to prepare the state for input to the neural network.

Iteration:

The process repeats as the agent continues to interact with the environment, updating the replay buffer, training the neural network, and making decisions based on the learned Q-values. This iterative process of interacting with the environment, collecting experiences, updating the replay buffer, and training the neural network is the essence of the DQN algorithm. The goal is for the agent to learn a policy that maximizes its cumulative future rewards in the given environment.

**6. Explanation of Algorithm:**

### 1. Environment Setup:

**Step 1: Package Installation**

* Installs necessary packages: **python-opengl**, **ffmpeg**, **pyvirtualdisplay**, **highway-env**, **stable\_baselines3**, **gym**.

**Step 2: Virtual Display Configuration**

* Creates a virtual display using **pyvirtualdisplay** for rendering without a physical display.

**Step 3: Animation Utility Functions**

* Defines utility functions for creating and displaying animations.
* **create\_anim(frames, dpi, fps)**: Creates a matplotlib animation from a list of frames.
* **display\_anim(frames, dpi, fps)**: Displays the animation using **display.HTML**.
* **save\_anim(frames, filename, dpi, fps)**: Saves the animation to a file.

### 2. Environment Registration:

**Step 4: Register Custom Environments**

* Registers custom environments from the **highway\_env** package using **highway\_env.register\_highway\_envs()**.

3. Training DQN Model:

**Step 5: Initialize**

* Initialize the Q-network with random weights.
* Create a target Q-network with the same architecture and weights as the main Q-network.
* Initialize a replay buffer to store experiences (state, action, reward, next state, done).
* Set hyperparameters: discount factor (*γ*), exploration parameters (*ϵ*).

**Step 6: Collect Initial Data**

* Populate the replay buffer with initial experiences by taking random actions.

**Step 7: Training Loop**

**For each episode:**

* Reset the environment and observe the initial state (*s*0 ).
* **While the episode is not done:**
* **Exploration-Exploitation Tradeoff:**
* With probability *ϵ*, select a random action (exploration).
* Otherwise, select the action with the highest Q-value from the Q-network (exploitation).
* Execute the selected action and observe the next state (*s*′), reward (*r*), and whether the episode is done (*done*).
* Store the experience (*s*,*a*,*r*,*s*′,done) in the replay buffer.
* Sample a mini-batch of experiences from the replay buffer.
* **Compute Temporal Difference (TD) Target:**

A close-up of a blue background

Description automatically generated

* Update the Q-network by minimizing the mean squared TD error:

A white text on a black background

Description automatically generated

* Update the target Q-network weights every *C* steps

**Step 8: Repeat**

* Continue training for a specified number of episodes or until convergence.

4. Visualization:

**Step 9: Visualize Training Logs with Tensorboard**

* Uses the **%tensorboard** magic command to launch Tensorboard and visualize training logs.

**Step 10: Load Trained Model and Collect Frames**

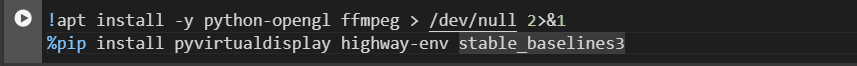
* Loads the trained DQN model using **DQN.load()**.
* Collects frames during the model's interaction with the environment.
* Closes the environment.

**Step 11: Display Animation**

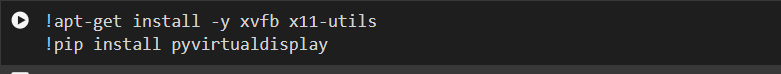
* Checks if frames are available and have a valid shape.
* Creates an animation and displays it using **display.HTML** if conditions are met.

**7. Prototype of 30%Proposed Design:**

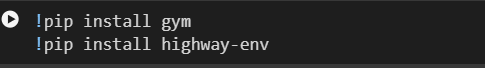
Installing required libraries :



* stable\_baselines3 is a set of high-quality implementations of reinforcement learning algorithms. It provides a collection of popular algorithms with a consistent interface, making it easy to experiment with and compare different approaches.
* Used in training and evaluating reinforcement learning models and experimenting with various algorithms on custom environments.

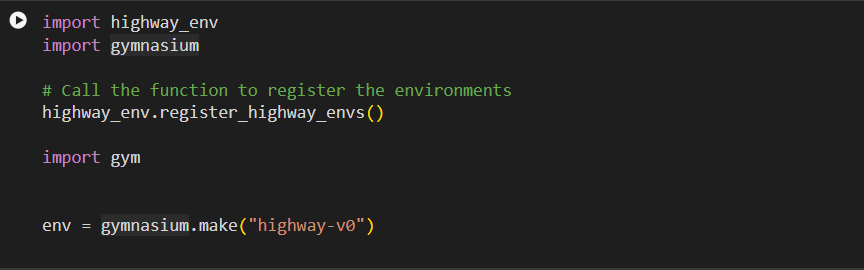


* The `pyvirtualdisplay` package provides a way to create a virtual display in Python, enabling the rendering of graphical user interfaces without a physical display. This is particularly useful in headless environments such as servers or cloud platforms.



* The highway-env package is a collection of highway driving environments for reinforcement learning. It provides a set of scenarios and agents to simulate complex traffic scenarios for training and evaluating reinforcement learning algorithms.

Settingup Highway Environment:



* Import the highway\_env and gymnasium modules.

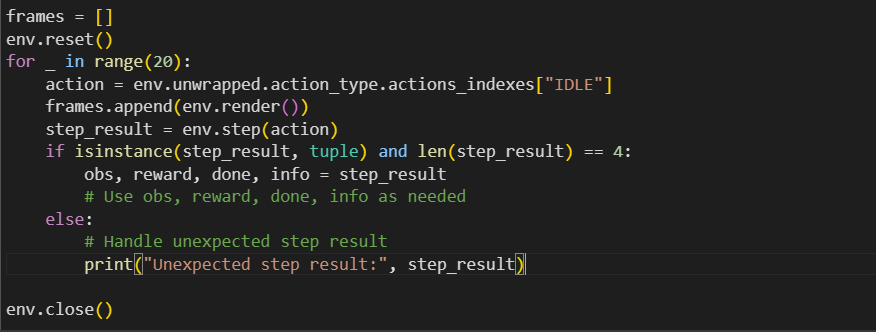
Environment Registration:

* Call the register\_highway\_envs function from highway\_env to register custom environments.

Gym Environment Creation:

* Using the gymnasium.make function to create a Gym environment with the specified ID ("highway-v0").

**Frames Initialisation:**



Environment Reset:

* Reset the environment to obtain the initial observation.

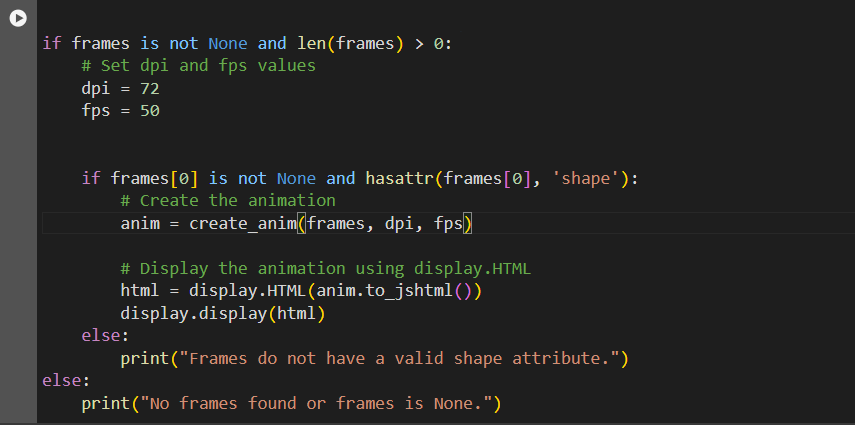
Frame Collection Loop:

* Loop for a specified number of steps (in this case, 20).
* Retrieve the action index for the predefined action ("IDLE").
* Collect the current frame by rendering the environment and append it to the frameslist.
* Take a step in the environment with the chosen action.
* Check if the step result is a tuple with the expected length.
* If successful, extract observation, reward, done flag, and additional info.
* If unexpected, handle the unexpected step result.

Environment Closure:

Close the environment after collecting frames.

Animation Creation:



Frame Availability Check:

* Check if the frames variable is not None and has a valid length.

Animation Setup

* Set the dpi (dots per inch) and fps (frames per second) values for the animation.

Valid Shape Attribute Check

* Check if the first frame in the frames list is not None and has a valid shape attribute.

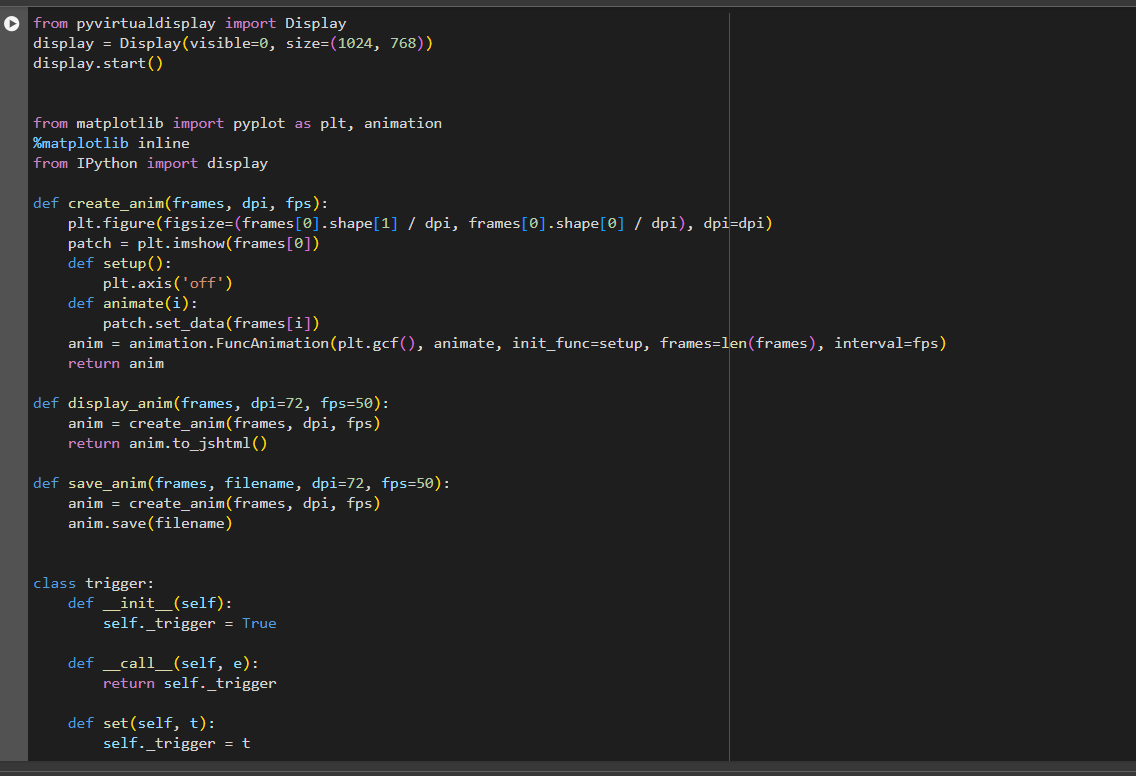
Animation Creation and Display

* If all conditions are met, create the animation using the create\_anim function.

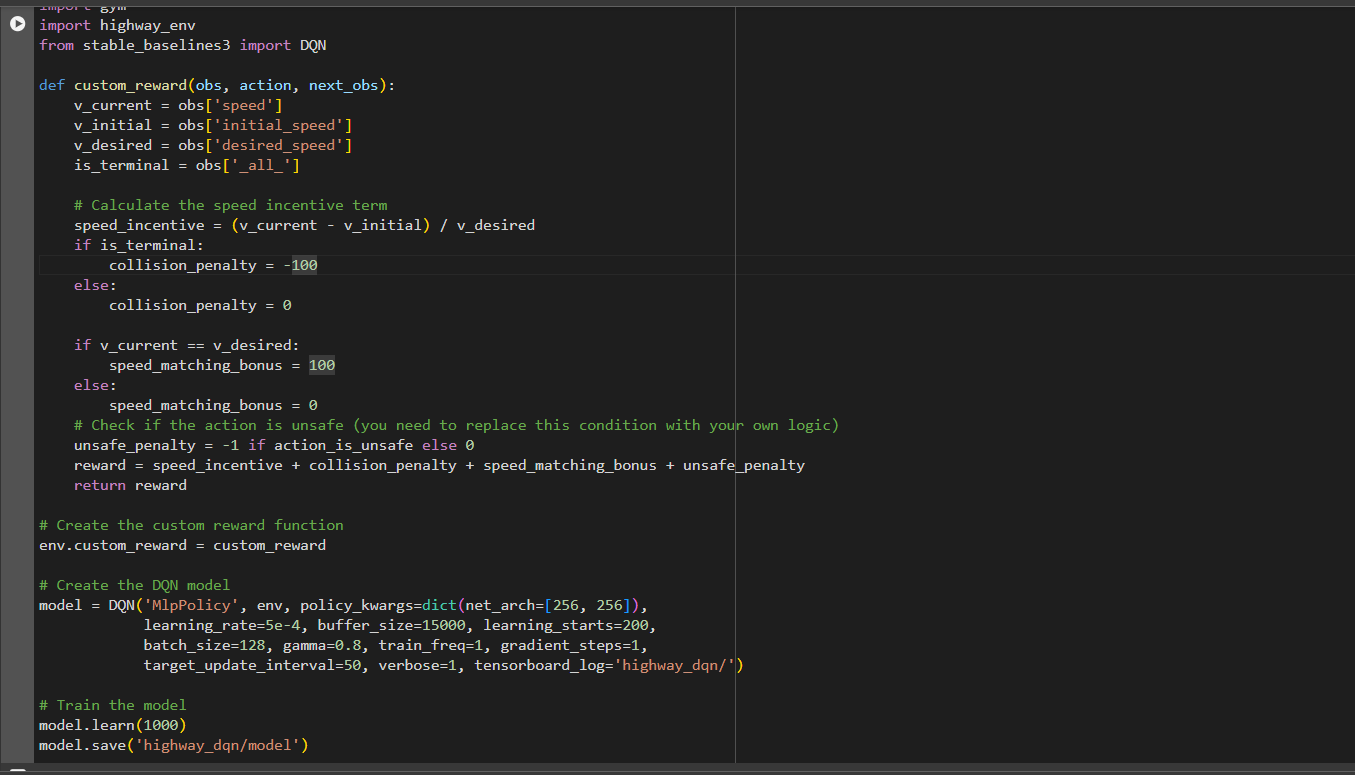
Error Handling:

If frames do not have a valid shape attribute or no frames are found, print an appropriate message.

**Animation Utility :**



**Model Training:**



This showcases the integration of a custom reward function into a reinforcement learning environment using the `stable\_baselines3` library. The custom reward function is tailored for the `highway-env` environment and incorporates key terms such as speed incentive, collision penalty, speed matching bonus, and unsafe penalty.

1. Define the Custom Reward Function

A custom reward function, `custom\_reward`, is created. It calculates terms such as speed incentive based on current and initial speeds, collision penalties for terminal states, speed matching bonuses for reaching the desired speed, and a penalty for unsafe actions.

2. Apply the Custom Reward Function to the Environment

The custom reward function is assigned to the environment using `env.custom\_reward = custom\_reward`. This step ensures that the specified reward logic is used during the agent's interactions with the environment.

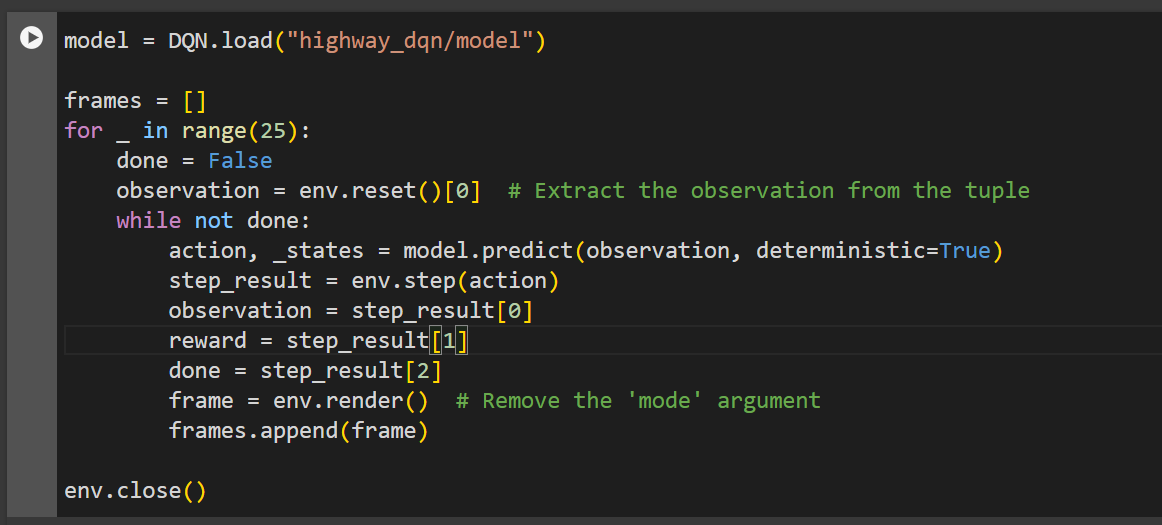
3. Create the DQN Model

A DQN (Deep Q-Network) model is instantiated using the `DQN` class from `stable\_baselines3`. It includes parameters such as the policy type, learning rate, buffer size, and architecture of the neural network.

4. Train the Model

The model is trained using the `model.learn` method for a specified number of iterations. Training results are saved to the 'highway\_dqn/model' directory.

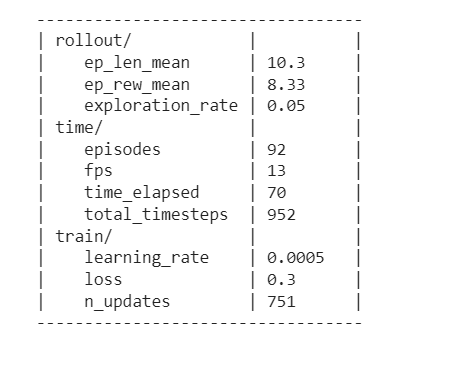
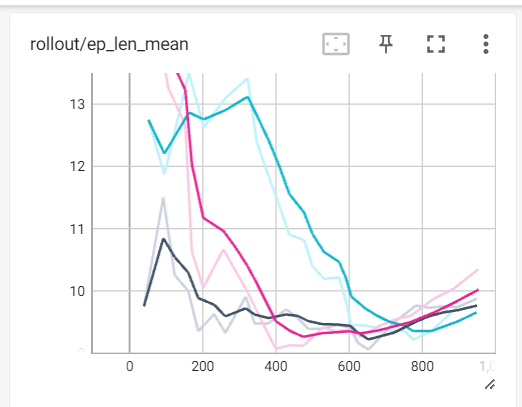
**Model Testing:**



This section illustrates the process of testing a pre-trained DQN (Deep Q-Network) model and collecting frames to visualize the agent's behavior. The loaded model undergoes inference in the environment, and frames are captured at each step to provide insights into the agent's decision-making process.

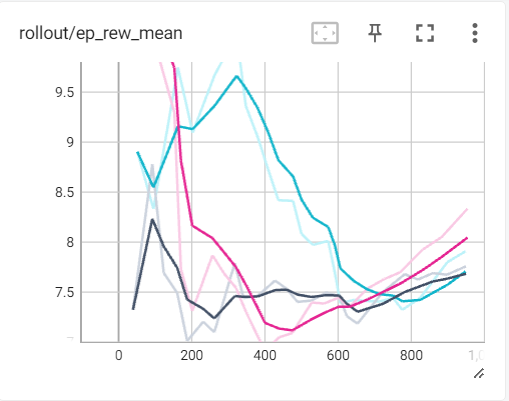
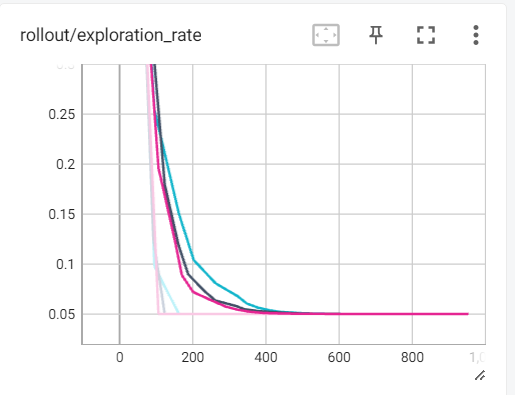
**Conclusion and Results:**

**Training Results:**

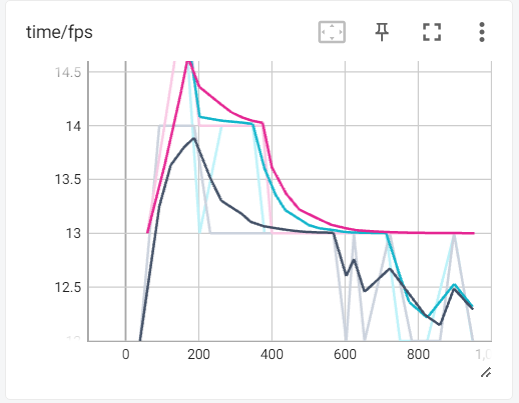
 

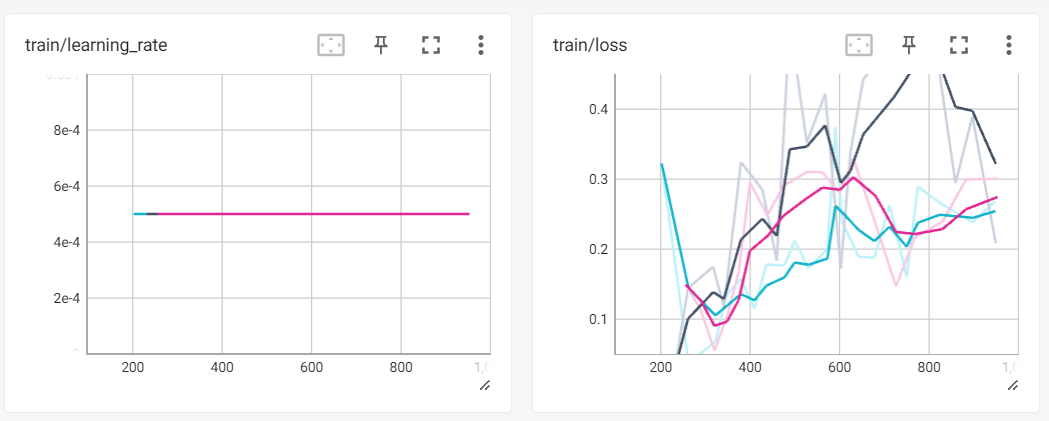
  

Mean Episode reward graph Exploration rate:

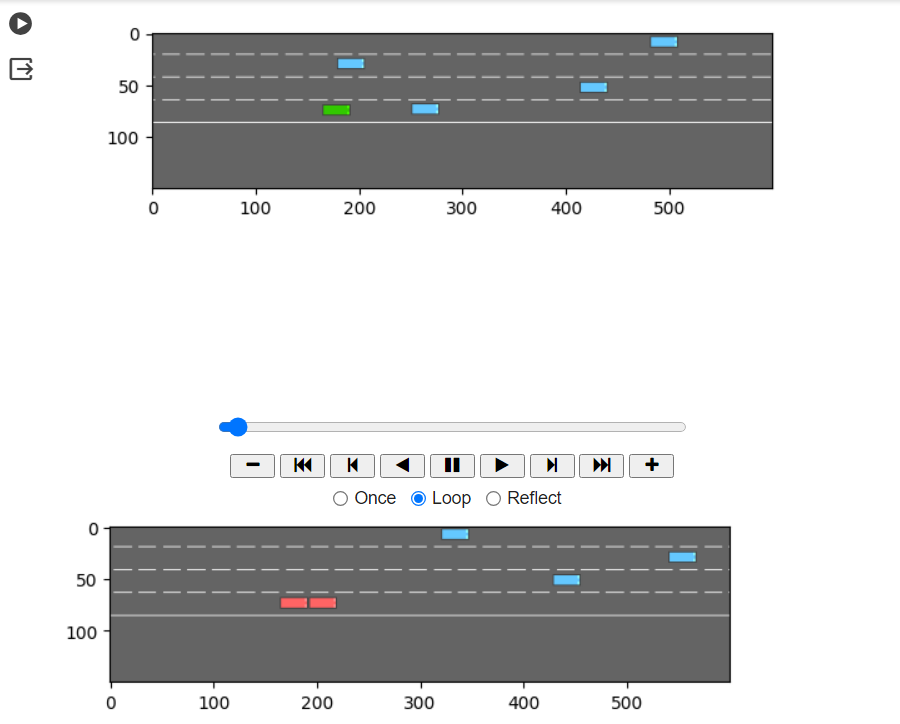
 

Frames per second:





Testing Results:



**8. Novelty of the Solution**

8.1 Enhanced Model Training:

In pursuit of optimizing model training, the primary focus lies in the development of a more efficient reward function. By refining the reward mechanism, the Q-learning model will receive clearer guidance, fostering improved decision-making and enhancing its learning capabilities. This strategic adjustment aims to elevate the overall performance of the autonomous vehicle system.

8.2 Synergistic Q-Learning Models:

A key objective is to combine multiple Q-learning models to achieve synergistic benefits. The integration of diverse Q-learning approaches will capitalize on their individual strengths, fostering a collective intelligence that contributes to superior results. This collaborative model fusion is expected to enhance the robustness and adaptability of the autonomous system.

8.3 Collision-Free Lane Changes:

The implementation of smooth lane change algorithms takes center stage in ensuring collision-free maneuvers. The focus is on developing algorithms that facilitate seamless transitions between lanes, mitigating collision risks during the lane-changing process. This emphasis on safety aligns with the broader goal of creating a reliable and secure autonomous driving system.

**9. Work Plan and Publication Plan for phase 2:**

#### **1. Implementing Double DQN Algorithm:**

**Objective:** Implement Double DQN for enhanced Q-learning in lane change and obstacle prediction.

**Steps:**

* **Literature Review:** Understand principles and advantages of Double DQN.
* **Algorithm Design and Implementation:** Design and code Double DQN.
* **Integration:** Integrate Double DQN into the existing Q-learning system.
* **Testing:** Conduct rigorous testing for validation.

#### **2. Implementation of Rainbow DQN Algorithm**

**Objective:** Implement Rainbow DQN to further improve Q-learning capabilities.

**Steps:**

* **Literature Review:** Explore Rainbow DQN components and advancements.
* **Algorithm Design and Implementation:** Design and code Rainbow DQN.
* **Integration:** Integrate Rainbow DQN into the autonomous vehicle system.
* **Testing:** Perform comprehensive testing for validation.

#### **3. Integrating Rainbow DQN with Other DQN Variants**

**Objective:** Explore synergies by integrating Rainbow DQN with other DQN variants.

**Steps:**

* **Literature Review:** Understand potential synergies and challenges.
* **Algorithm Integration Design and Implementation:** Design and implement integration with other DQN variants.
* **Testing:** Validate performance through testing.

#### **4. Implementing an Efficient Algorithm for Obstacle Avoidance**

**Objective:** Develop and implement an efficient obstacle avoidance algorithm.

**Steps:**

* **Literature Review:** Explore existing obstacle avoidance algorithms.
* **Algorithm Design and Implementation:** Design and code an efficient obstacle avoidance algorithm.
* **Integration:** Integrate the algorithm into the autonomous vehicle system.
* **Testing:**Conduct thorough testing for effectiveness.

**10. Abstract for Article publication:**

This article presents a novel approach for enhancing road safety through the integration of lane change detection and obstacle prediction using Q-learning algorithms. The study focuses on the development of an optimal algorithm leveraging Rainbow DQN and other Q-learning variants. The proposed methodology utilizes reinforcement learning techniques to efficiently analyse and predict dynamic changes in traffic scenarios.

Our research addresses the challenges associated with real-time decision-making in autonomous vehicles by employing Q-learning to capture complex interactions during lane changes. The integration of Rainbow DQN enhances the learning process by combining various Q-learning algorithms, resulting in a more robust and adaptive model. The algorithm's effectiveness is evaluated through extensive simulations and real-world scenarios, demonstrating its ability to accurately detect lane changes and predict potential obstacles.

The findings suggest that our approach significantly improves the accuracy and reliability of lane change detection and obstacle prediction compared to traditional methods. Furthermore, the versatility of Rainbow DQN allows for seamless adaptation to diverse driving conditions and environments. This research contributes to the advancement of intelligent transportation systems, laying the groundwork for safer and more efficient autonomous navigation

**11. Conclusion**

In conclusion, the project "Autonomous Vehicle Lane Change and Obstacle Prediction using Q-Learning" aims to enhance the capabilities of autonomous vehicles in making safe and efficient lane change decisions while predicting and avoiding obstacles. By utilizing Q-Learning, a reinforcement learning technique, the project proposes a methodology to train an intelligent model that can analyse the vehicle's state, predict lane change likelihood, and anticipate dynamic obstacles.

The project's methodology involves collecting relevant sensor data, defining the state and action space, and implementing a Q-Learning model tailored for lane change and obstacle prediction. The model is trained using historical or simulated data, and Q-values are updated to optimize decision-making. Additionally, the trained model is utilized to predict lane change probabilities and assign confidence scores to different actions.

To further enhance the system, the methodology extends the Q-Learning model to include obstacle prediction. By incorporating obstacle detection algorithms and sensor fusion techniques, the system can accurately anticipate dynamic obstacles in the environment. Real-time decision-making is implemented to continuously evaluate the vehicle's state and adapt to changing road conditions. The trained models are used to make informed decisions on when and how to change lanes to avoid obstacles, ensuring safe navigation.

The project concludes with an evaluation and validation phase, where extensive simulations and real-world testing are conducted. Performance metrics such as success rate of lane changes, obstacle avoidance accuracy, and overall system performance are measured. This validation process ensures the system's safety, reliability, and effectiveness under various scenarios and road conditions.

Overall, the proposed project aims to enhance autonomous vehicles' capabilities in lane change decision-making and obstacle prediction using Q-Learning. By implementing this methodology, autonomous vehicles can navigate roads more efficiently, avoid collisions with obstacles, and provide a safer driving experience for passengers and other road users.

# **11. REFERENCES**

[1] Ibrahim, M. M., Atia, M. R., & Fakhr, M. W. (2021, December). Autonomous Vehicle Path Planning using Q-Learning. In *Journal of Physics: Conference Series* (Vol. 2128, No. 1, p. 012018). IOP Publishing.

[2] Forbes, J. R. N. (2002). Reinforcement learning for autonomous vehicles. University of California, Berkeley.

[3] Highway Traffic Modelling and Decision Making for Autonomous Vehicle Using Reinforcement Learning

[4] Fayjie, A. R., Hossain, S., Oualid, D., & Lee, D. J. (2018, June). Driverless car: Autonomous driving using deep reinforcement learning in urban environment. In 2018 15th international conference on ubiquitous robots (ur) (pp. 896-901). IEEE

[5] Jebessa, E., Olana, K., Getachew, K., Isteefanos, S., & Mohd, T. K. (2022, January). Analysis of Reinforcement Learning in Autonomous Vehicles. In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0087-0091). IEEE.

[6] Zhang, Y., Gao, F., & Zhao, F. (2023). Research on Path Planning and Tracking Control of Autonomous Vehicles Based on Improved RRT\* and PSO-LQR. Processes, 11(6), 1841.

[7] Chen, D., Jiang, L., Wang, Y., & Li, Z. (2020, July). Autonomous driving using safe reinforcement learning by incorporating a regret-based human lane-changing decision model. In 2020 American Control Conference (ACC) (pp. 4355-4361). IEEE.

[8] Kalweit, G., Huegle, M., Werling, M., & Boedecker, J. (2021, September). Q-learning with Long-term Action-space Shaping to Model Complex Behavior for Autonomous Lane Changes. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5641-5648). IEEE.

[9] Kosuru, V. S. R., & Venkitaraman, A. K. (2022). Developing a deep Q-learning and neural network framework for trajectory planning. European Journal of Engineering and Technology Research, 7(6), 148-157.

[10] Yavas, U., Kumbasar, T., & Ure, N. K. (2020, October). A new approach for tactical decision making in lane changing: Sample efficient deep Q learning with a safety feedback reward. In 2020 IEEE Intelligent Vehicles Symposium (IV) (pp. 1156-1161). IEEE.

[11] Hoel, C. J., Driggs-Campbell, K., Wolff, K., Laine, L., & Kochenderfer, M. J. (2019). Combining planning and deep reinforcement learning in tactical decision making for autonomous driving. IEEE transactions on intelligent vehicles, 5(2), 294-305.

[12] Li, G., Yang, Y., Li, S., Qu, X., Lyu, N., & Li, S. E. (2022). Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness. Transportation research part C: emerging technologies, 134, 103452.

[13] Das, L., & Won, M. (2021, May). D-ACC: Dynamic Adaptive Cruise Control for Highways with Ramps Based on Deep Q-Learning. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1572-1578). IEEE

[14] Quek, Y. T., Koh, L. L., Koh, N. T., Tso, W. A., & Woo, W. L. (2021). Deep Q‐network implementation for simulated autonomous vehicle control. IET Intelligent Transport Systems, 15(7), 875-885