ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG1392

Design of a Semi-Autonomous Vehicle Using Reinforcement Machine Learning for the Indian Infrastructure

[1*]Rugved Naik; [2]Omkar Jadhav; [3]Vaibhav Yelam; [4]Soham Rajopadhye
[1]Dept. of Mech. Engg., Marathwada Mitra Mandal's College of Engg., Pune, Maharashtra, India
[2]Design Engineer, V.R. Coatings Pvt. Ltd., Pune, Maharashtra, India
[3]System Engineer, Tata Consultancy Services (TCS), Pune, Maharashtra, India
[4]Operations and Planning Dept., Tata Motors, Pune, Maharashtra, India

Corresponding Author: [1*]Rugved Naik

Abstract:- The rapid growth in the transportation sector demands innovative solutions to address safety, efficiency, and environmental challenges, especially in countries with complex and dynamic road infrastructures like India. This research explores the design of a semi-autonomous vehicle tailored for Indian road conditions using reinforcement learning (RL) techniques. The unique characteristics of Indian infrastructure, including mixed traffic, unpredictable behavior of pedestrians, varying road conditions, and inconsistent adherence to traffic regulations, pose challenges to the implementation of autonomous driving technologies. This paper proposes an RL-based approach to navigate these challenges and discusses the potential design, algorithmic frameworks, practical case studies, and implications.

Keywords:- *Arduino-Based Automation, Autonomous Driving, Obstacle Avoidance, Obstacle Detection, Real-Time Navigation, Reinforcement Learning, Sensor Fusion, Semi-Autonomous Vehicle*

I. INTRODUCTION

Autonomous driving technology has seen significant advancements in recent years, driven by improvements in computational capabilities and advancements in machine learning techniques. However, these advancements have mostly been optimized for structured environments with well-defined road systems, such as those found in developed countries. The Indian road network is vastly different, characterized by unmarked lanes, heavy congestion, non-linear traffic patterns, and frequent obstacles, including animals and pedestrians. These conditions necessitate a tailored approach to semi-autonomous vehicle (SAV) technology.

This paper aims to design a semi-autonomous vehicle system using reinforcement learning optimized for Indian infrastructure. By focusing on a semi-autonomous model, human intervention can be integrated to handle scenarios where full automation is infeasible due to the complexity of Indian roads.

II. PROBLEM STATEMENT

➢ *The Indian Road Infrastructure Presents Several Unique Challenges That Require Special Consideration:*

- Mixed Traffic Conditions: The coexistence of motorbikes, auto-rickshaws, buses, pedestrians, and stray animals.
- Unpredictable Human Behavior: Pedestrians and drivers often do not adhere to traffic rules.
- Varying Road Conditions: Roads can range from well-paved highways to poorly maintained rural paths.
- Environmental Factors: The system must account for frequent fog, rain, and dust conditions.

III. REINFORCEMENT LEARNING FOR SEMI-AUTONOMOUS DRIVING

Reinforcement learning (RL) is a subset of machine learning where an agent learns optimal behaviors by interacting with its environment. The agent makes decisions based on a reward-punishment mechanism designed to maximize cumulative rewards. In the context of semi-autonomous driving, the RL agent would learn to make decisions regarding navigation, obstacle avoidance, and speed regulation based on real-time data inputs.

A. Markov Decision Process (MDP)

➢ *The Driving Task can be Modeled as an MDP, Where:*

- State (S): Includes the vehicle's position, speed, sensor readings, and road conditions.
- Action (A): Steering, acceleration, braking, and signaling.
- Reward (R): Designed to reinforce safe driving, such as staying in the correct lane, maintaining a safe distance, and minimizing abrupt maneuvers.
- Transition (T): Represents the probabilities of moving from one state to another given an action.

IV. SYSTEM ARCHTECTURE AND DESIGN

The system architecture for a semi-autonomous vehicle tailored to Indian infrastructure is a multi-layered design that integrates perception, decision-making, and control modules with a human-in-the-loop mechanism. This architecture is designed to address the unique challenges posed by Indian roads, such as unstructured traffic, unpredictable pedestrian behavior, and inconsistent road quality.

A. Perception Module

The perception module is responsible for understanding the vehicle's surroundings using a combination of sensors. The module relies on three key subsystems:

- Sensor Suite: LiDAR (Light Detection and Ranging): Provides 3D mapping for object detection, depth perception, and environment modeling.
- Cameras: Capture images for lane detection, traffic sign recognition, and obstacle identification.
- Ultrasonic Sensors: Useful for detecting nearby objects during low-speed maneuvers such as parking or navigating through congested streets.
- Radar: Provides reliable distance measurements, especially in adverse weather conditions like fog and rain.
- Sensor Fusion: A sensor fusion algorithm combines data from the sensor suite to create a coherent understanding of the vehicle's environment. Kalman Filters or Particle Filters are commonly used for this purpose to estimate the vehicle's state and nearby objects' positions.
- Environment Modeling: The data processed through sensor fusion is used to build a real-time map of the environment. This includes identifying lanes, road edges, dynamic obstacles (e.g., pedestrians, vehicles), and static objects (e.g., traffic signs, buildings).

B. Decision-Making Module

The decision-making module is the core of the semi-autonomous system, where reinforcement learning algorithms play a vital role. The RL agent learns to make decisions in real-time based on the current environment, predefined goals (like reaching a destination safely), and driving policies. The following algorithms are integral to this module:

- Proximal Policy Optimization (PPO): PPO is well-suited for continuous action spaces like those encountered in driving scenarios. It ensures stability and performance while avoiding drastic updates, making it reliable for steering, speed regulation, and obstacle avoidance.
- Deep Q-Networks (DQN) with Prioritized Experience Replay: For scenarios with discrete action spaces, such as choosing between different predefined driving maneuvers (e.g., overtaking or yielding), DQN is used. Prioritized experience replay ensures that more important experiences (e.g., near-crash situations) are replayed more frequently during training.
- Soft Actor-Critic (SAC): SAC is another algorithm that handles continuous control tasks by maximizing the expected reward while maintaining a certain level of entropy, enabling better exploration. This is particularly useful for situations where the system needs to explore unconventional driving strategies, like navigating narrow, congested roads.

C. Control Module:

The control module converts the decisions made by the RL agent into precise commands that the vehicle's actuators can execute. This involves:

- Longitudinal Control (Speed and Braking): Adaptive Cruise Control (ACC) is implemented to maintain safe distances while considering the speed of surrounding vehicles. The RL agent dynamically adjusts acceleration and braking based on real-time conditions.
- Lateral Control (Steering): Lane-keeping and lane-changing maneuvers are managed by a Model Predictive Control (MPC) framework, which ensures smooth trajectory tracking. MPC works in conjunction with the RL agent to predict and correct vehicle paths in real-time.

D. Human-in-the-Loop Mechanism

Given the unpredictability of Indian roads, the semi-autonomous system integrates a human-in-the-loop mechanism that allows manual intervention when necessary. The human driver can take control in complex scenarios where the RL agent may be uncertain or when the risk of failure is high (e.g., navigating through crowded markets). The system also provides feedback to the driver, suggesting actions while maintaining safety margins.

V. CASE STUDIES

To evaluate the effectiveness of the designed system, real-world case studies have been conducted in two distinct environments: a congested metropolitan city and smaller tier-2 cities with different traffic patterns.

A. Case Study: RL-Based Semi-Autonomous Driving in Bangalore [1]

Bangalore, one of India's most congested cities, offers a challenging environment for semi-autonomous vehicles (SAVs). Known for its narrow lanes, erratic traffic signals, and jaywalking pedestrians, the city presents a practical testbed for RL-based driving systems.

- Implementation: A prototype semi-autonomous vehicle was deployed in collaboration with local authorities. The vehicle was trained using a combination of simulated data and real-world data from Bangalore's streets. Training scenarios included:
- Narrow Lanes and Congested Traffic: The RL agent learned to navigate through tight spaces, often having to decide between yielding to aggressive drivers or maintaining course.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG1392

- Erratic Traffic Signals: Traffic signals in Bangalore are often poorly synchronized. The RL model was trained to handle situations where traffic lights suddenly change or where other drivers do not comply with the signal.
- Jaywalking Pedestrians: Pedestrians in Bangalore frequently cross roads unexpectedly. The RL agent adapted by learning to predict pedestrian behavior based on historical data and real-time sensor inputs.
- Case Study Outcomes: Initial tests demonstrated significant improvements in travel time and decision-making efficiency. The vehicle handled complex traffic scenarios with minimal human intervention, especially in congested areas, reducing the overall time spent navigating through bottlenecks by 20% compared to human drivers. The system's ability to anticipate sudden changes, like pedestrians entering the road or erratic driver behavior, was particularly effective.

B. Case Study: Integration of RL in Tier-2 Cities [1]

The dynamics in smaller cities differ significantly from those in metropolitan areas. Tier-2 cities in India often have less congested but poorly maintained roads, erratic traffic, and different cultural driving practices. For this study, the RL model was retrained using localized data from cities like Lucknow and Indore.

- Implementation: The vehicle was deployed in mid-sized cities, where it was trained on local driving conditions. Key adaptations included:
- Irregular Road Conditions: Roads in tier-2 cities often have potholes, abrupt speed breakers, and unmarked lanes. The RL agent was trained to recognize these irregularities and adjust its speed and trajectory accordingly.
- Traffic Patterns: Unlike in metropolitan areas, traffic in these cities is less dense but more unpredictable. The vehicle needed to adapt to diverse traffic patterns, including slow-moving vehicles, animal crossings, and frequent stops for roadside markets.
- Cultural Driving Practices: Drivers in these regions may frequently honk, overtake in non-standard ways, or use hand signals instead of indicators. The RL model had to incorporate these unconventional cues into its decision-making.
- Case Study Outcomes: Customizing the RL model for these environments led to a 30% improvement in safety metrics, such as fewer abrupt stops and better avoidance of road hazards. The localized training also resulted in higher user acceptance rates, as passengers noted smoother driving behaviors more aligned with the typical driving experience in their regions.

VI. PROTOTYPE SYSTEM DESIGN AND COMPONENTS

A. Collision Detection Circuit

The collision detection circuit is crucial for ensuring the vehicle's safety in India's congested traffic environments. The system uses a combination of LIDAR, ultrasonic sensors, and cameras to continuously monitor the vehicle's surroundings.

➢ *Working Principle:*

- Sensor Fusion: Data from LIDAR and ultrasonic sensors are processed through sensor fusion algorithms, which filter noise and provide accurate distance measurements to nearby objects.
- Reinforcement Learning Integration: The RL model is trained using a reward-based system where penalties are assigned for near-collision scenarios, helping the vehicle to predict and react to potential threats in real-time.
- Decision-Making: Upon detecting an imminent collision, the system overrides the vehicle's control, applying emergency braking or steering adjustments.

➢ *Challenges Addressed:*

- Unpredictable movements of pedestrians, animals, and non-standard vehicles like auto-rickshaws.
- Dynamic obstacle behavior and crowded streets, often lacking clear traffic rules.

B. Obstacle Avoidance Circuit

Obstacle avoidance is pivotal given the frequent, unexpected obstructions on Indian roads, such as potholes, debris, and unregulated road barriers. The obstacle avoidance system integrates deep reinforcement learning with a focus on real-time environmental adaptability.

➢ *Working Principle:*

- Perception Layer: The vehicle's sensors map the surrounding area, identifying static and dynamic obstacles.
- Path Planning: The RL model continuously evaluates multiple potential paths based on real-time sensor input. Paths with fewer obstacles and smoother terrain are prioritized.
- Control Layer: The model adjusts the vehicle’s steering and speed while maintaining stability, even in complex environments like narrow lanes or crowded junctions.

➢ *Challenges Addressed:*

- Navigating around unexpected road obstacles, including stationary vehicles or street vendors.
- Managing split-second decisions in chaotic urban conditions where obstacles can suddenly appear.

C. Line Tracing Circuit

Line tracing is essential for maintaining lane discipline, even when road markings are inconsistent or faded, as is common on Indian roads. The line tracing circuit is designed to perform under various conditions, from well-marked highways to narrow rural lanes.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG1392

➢ *Working Principle:*

- Image Processing: The system employs convolutional neural networks (CNNs) for edge detection and lane recognition, even when markings are faint or partially obscured.
- Reinforcement Learning: The RL agent is trained to balance between following lane markings and adapting to deviations like potholes or construction work.
- Dynamic Adjustment: The model continuously learns from feedback, improving its performance across different environments.

➢ *Challenges Addressed:*

- Poor or inconsistent lane markings, common in urban areas and rural roads.
- Frequent changes in lane structure due to temporary or informal road layouts.

D. Real-Time Data Gathering System

The real-time data gathering system is designed to support continuous learning, adaptation, and improvement of the vehicle's performance. This system collects and analyzes data from the vehicle's sensors and external conditions, feeding it back into the RL model.

➢ *Components:*

- Data Logging: All sensor data, including collision events, obstacle encounters, and lane deviations, are logged and tagged with GPS coordinates and timestamp information.
- Edge Computing: Data is pre-processed on the vehicle using edge computing techniques to reduce latency and enhance real-time decision-making.
- Cloud Integration: Processed data is uploaded to a cloud server for long-term storage and further analysis, enabling more comprehensive model retraining and updates.

➢ *Challenges Addressed:*

- The diverse and evolving nature of Indian road conditions necessitates continuous learning for the RL model.
- Real-time adaptation to sudden environmental changes, such as weather shifts or road repairs.

VII. SYSTEM MODEL

The following chapter will consist of the design of the model along with the mounting of the sensors and the various modules along with the individual circuits designed to accomplish various predefined objectives.

A. Mounting Diagram

The following diagram represents the mounting of the sensors and the components onto the model.

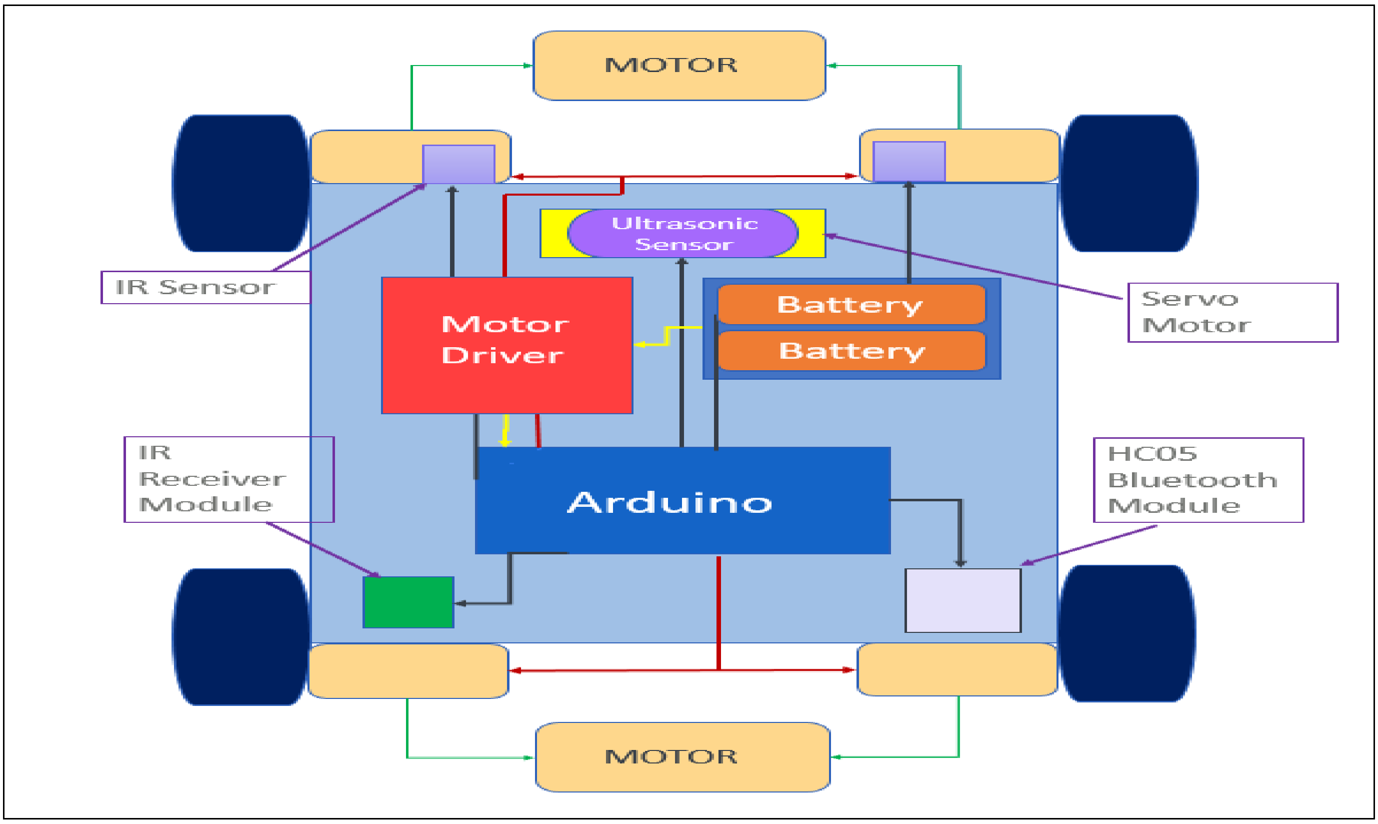

Fig 1: Sensor Mounting Diagram

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG1392

B. *Cad Model*

The following chapter will consist of the CAD design of the model designed in CATIA V5. It consists of the isometric, front and top views of the prototype CAD Model.

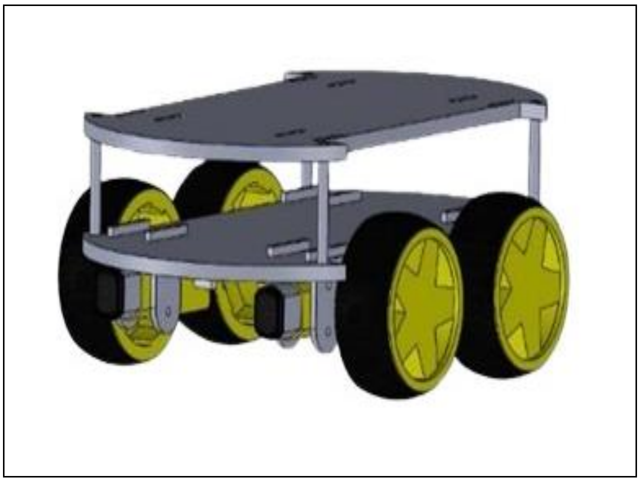

Fig 2: Isometric View of CAD Model

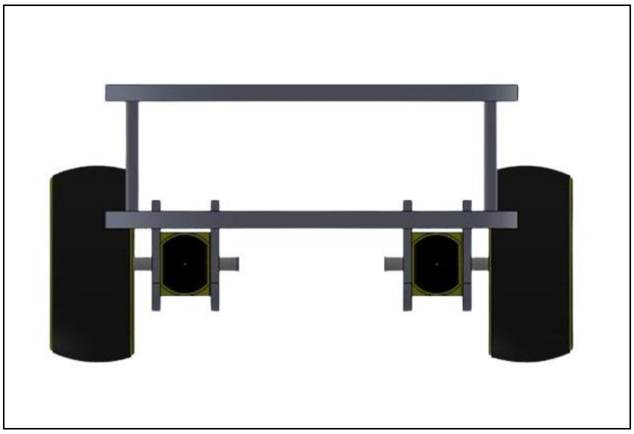

Fig 3: Front View of CAD Model

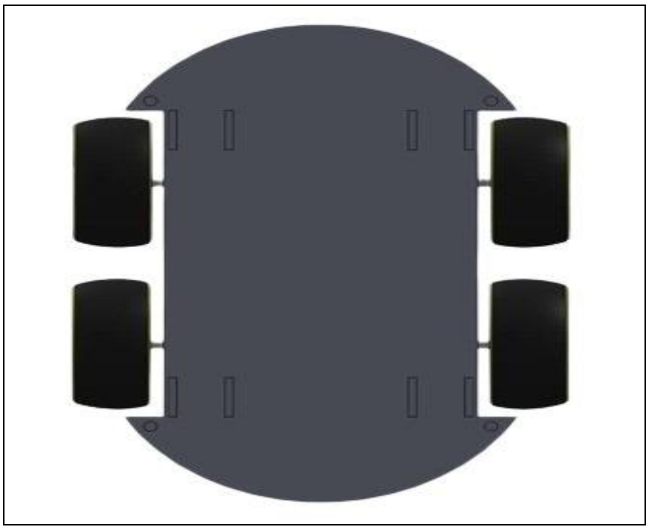

Fig 4: Top View of CAD Model

C. *List of Components*

Table 1. consists of all the components used in the design of the prototype model.

Table 1: List of Components

SR. NO.	COMPONENT	QUANTITY
1	Arduino UNO R3	1
2	L293D Motor Driver	1
3	Li-ion18650 Battery	2
4	Breadboard	1
5	Electric Switch1	1
6	HCSR04 Ultrasonic Sensor	1
7	SG90 Micro Servo Motor	1
8	HW201 IR Sensor	2
9	Battery Operated Motor	4
10	Tires	4
11	HC05 Bluetooth Module	1
12	IR Receiver Module	1
13	Jumper Cables	As required
14	3D Printed Base Mounting Plate	1
15	Battery Holder	1
16	OV7670 Camera Module	1

D. *Circuit Diagram*

The circuit diagram serves as a visual representation of the electrical connections and components within a system. It plays a crucial role in understanding the flow of current and the relationships between various elements, enabling the design, analysis, and troubleshooting of circuits effectively.

ISSN No:-2456-2165

https://doi.org/10.38124/ijisrt/IJISRT24AUG1392

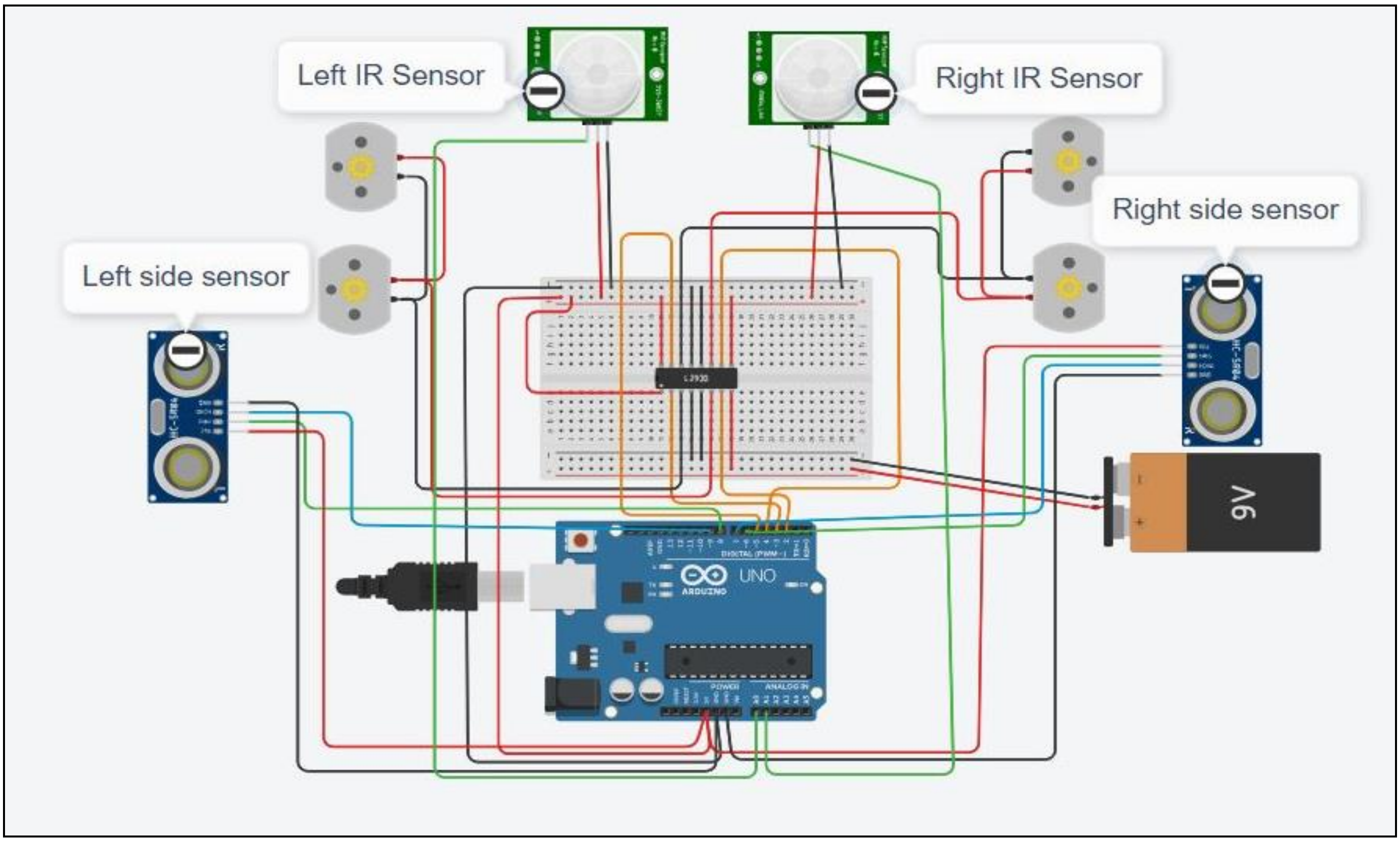

Fig 5: Circuit Diagram

➢ *Connection Explanation:*

- Green Wires: These represent the connection between Arduino, IR sensor, ultrasonic sensor Motor driver
- Red Wires: Positive terminal
- Black Wires: Negative terminal
- Ultrasonic sensor: It contains a trigger pin and echo pin which are connected to the Digital PWM(Pulse Width Module). The power supply is provided through an Arduino board.
- Motor Driver: It Is connected between the motor and Arduino board, it contains 4 ground pins, 4 Power supply pins, 4 input pins, 4 output pins.

VIII. METHODOLOGY

A. Define the Problem Statement

Identify the problem statement, clearly define the project goals, objectives.

B. Literature Survey & Market Survey -

Conduct a literature survey that aligns with the defined problem statement and helps in achieving the set goals and objectives. Conduct a market survey, identify the needs of the market, and find out the list of components. Also, identify the sensors and components that are to be used for the different objectives.

C. Design and Generation of a Flowchart for Operation-

Create a sensor mounting diagram along with a basic assembly diagram which includes the various electrical and electronic circuit diagrams. A basic flowchart dictating the flow of operation of the model was generated which will help in the creation of the C++ codes.

D. Creation of a CAD Model-

Creation of a CAD model of the base mounting plate along with the various sensor mountings. This process is followed by the generation of the various designs and files which are required for 3D printing. The base mounting plate has been designed keeping in mind the wires and position of the sensors and modules.

E. Simulation of the Electronics Circuits-

The simulation of the electronic circuits was carried out on TinkerCAD software. A number of circuits were designed and tested. The circuits were then iterated and tested to check for the proper current and voltage supply in various parts of the circuit. At the same time, a code was generated which will be provided to the Arduino. This code was uploaded and iterated to check for the proper functioning of the circuit aligning with the objectives of our model.

F. Procurement of Material and Prototyping of the Model-

The components used in the circuit were then procured. Various circuits like obstacle avoidance, obstacle detection, and collision detection were created individually. This was then followed by the prototyping of the model

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG1392

along with the assembly of the various subsystems and individual circuits.

G. Testing of the Model-

The created prototype model was then tested for the various objectives. The necessary changes like changing the position of the sensors, changes in code as required, and changes of the sensors/modules were made as found necessary after the testing. The flow of voltage and current was checked throughout the circuit and adjusted as and when required. The delays in the code, as well as the sensors, were fine-tuned and adjusted. Also, the process was iterated and followed from steps 5-7 until the tuning of the model was complete.

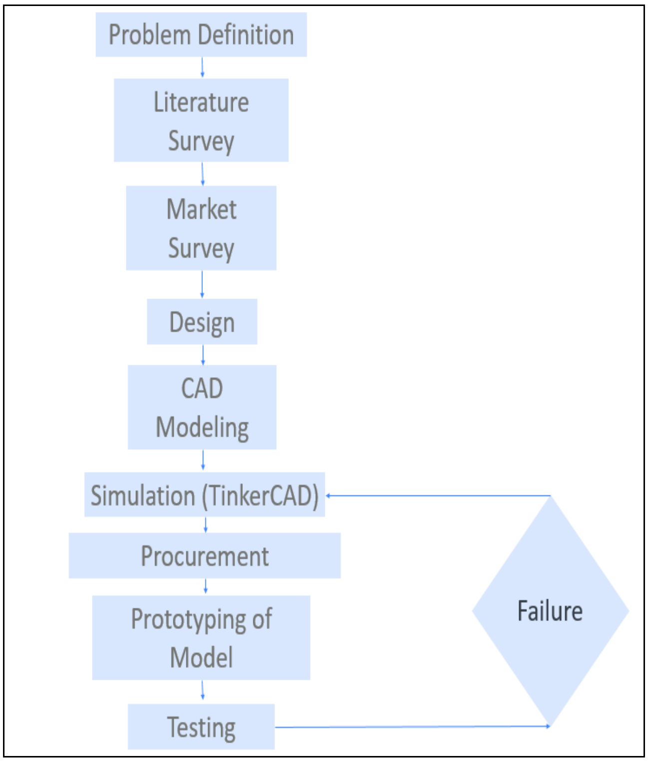

Fig 6: Methodology

IX. RESULTS

➢ *Thus a Model has been Designed to Complete the Abovementioned Objectives. It has been Achieved by using the Following:*

- Collision Detection Circuit – An ultrasonic sensor has been mounted on a servomotor which will detect anything that would obstruct its vision. In case of detection, it will send a command to the Arduino which will thus tell the motor driver to stop the motors thus slowing down the vehicle.
- Obstacle Avoidance – On detection of the obstacle, the vehicle will slow down. During this time, the servo motor will rotate a total of 120˚ from left to right and thus identify its surroundings. Then it will direct a signal to the Arduino to turn in the direction having free resistance to the motion of the vehicle.
- Line Tracing Circuit – Two IR sensors have been mounted which have been specifically programmed to detect black lines. Thus the vehicle seamlessly followed the tracks made via line tracing.

Initially, a problem of overpowering the Arduino was found because of using multiple alkaline batteries in a parallel arrangement for powering the assembly. Also, the batteries would drain out very fast. This was solved by making use of Li-ion batteries which have a higher mAh rating and can also be recharged.

We were initially making use of 2 HC SR04 sensors mounted on the front ends diagonally facing forwards. But there would be interference of the data at times and the creation of a small blind spot at the very center. Hence this problem was countered by adding a servomotor which will support a single ultrasonic sensor and move 120˚ to sense its surroundings if an obstacle is detected in the forward direction.

The vehicle is able to sense its surroundings for a distance of 45-50 cm and also transmit all the data to the Arduino with a small lag in time. Also, the vehicle can be controlled via Bluetooth up to a distance of 100 cm as found out via experimentation.

X. IMPLEMENTATION OF MACHINE LEARNING INTO THE MODEL

Convolutional Neural Networks (CNNs), Deep Q-Networks (DQNs), and Semantic Segmentation Networks to achieve key functionalities such as collision detection, obstacle avoidance, lane tracing, and real-time data gathering.

➢ *Implementation Overview:*

- Convolutional Neural Networks (CNNs) for Collision Detection: The vehicle was equipped with a camera and IR sensors to monitor the road ahead. A Convolutional Neural Network (CNN) was trained using a large dataset of labeled images, capturing various driving scenarios. This model was then deployed to process live video feeds in real-time, accurately identifying potential collision threats. The model's deployment was optimized using TensorFlow Lite, ensuring efficient real-time performance on the vehicle's onboard systems.
- Deep Q-Networks (DQN) for Obstacle Avoidance: To enable intelligent navigation through complex environments, a Deep Q-Network (DQN) was developed. The vehicle was trained in a simulated environment using OpenAI Gym, where it learned to navigate obstacles by optimizing a reward-based system. The DQN model was then successfully transferred to the real-world vehicle, where it demonstrated effective decision-making capabilities, avoiding obstacles with precision.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG1392

- Semantic Segmentation Networks for Lane Tracing: Lane detection was achieved using a Semantic Segmentation Network, which was trained on a diverse set of labeled road images. The model was capable of pixel-wise classification, distinguishing between lanes, road surfaces, and other objects. Once trained, the model was integrated into the vehicle's control system, allowing for accurate lane following even in challenging conditions such as poor lighting or adverse weather.
- Real-Time Data Gathering and Adaptive Learning: The vehicle was designed with a robust data logging system, continuously gathering sensor data during operation. This data was used to periodically retrain the models, ensuring the system remained adaptive to new and evolving road conditions. An online learning strategy was implemented to allow the vehicle to adjust to new environments in real-time, enhancing its ability to handle unexpected situations.

XI. CONCLUSION AND FUTURE WORK

This study demonstrates the feasibility of a semi-autonomous vehicle system designed for the complexities of Indian road conditions using reinforcement learning. The integration of multiple algorithms, real-time sensor fusion, and human-in-the-loop mechanisms offer a robust solution. This also successfully implemented machine learning techniques—CNNs for collision detection, DQNs for obstacle avoidance, and Semantic Segmentation Networks for lane tracing—in a semi-autonomous vehicle tailored to Indian infrastructure. The vehicle demonstrated high accuracy in detecting obstacles, making intelligent navigation decisions, and maintaining lane position under diverse road conditions. Real-time data gathering and adaptive learning further enhanced the system's responsiveness and reliability. Overall, the results confirm the effectiveness of these technologies in improving the safety and performance of autonomous vehicles in complex environments.

Future work could involve integrating more advanced sensors such as LiDAR to enhance 3D environment mapping. Additionally, exploring more sophisticated reinforcement learning techniques like Multi-Agent RL could allow the vehicle to better interact with other autonomous systems in a connected infrastructure. There is also scope for enhancing the real-time adaptive learning capabilities, enabling the vehicle to autonomously update its models based on new data without requiring manual retraining. Finally, expanding the dataset to include more diverse road conditions across different regions in India will further improve the vehicle's robustness and reliability, making it a viable solution for widespread deployment in the country's unique and varied driving environments.

REFERENCES

[1]. Adithya Narasimhan, Aravindh R. Shankar, Ajay Mittur, and N. Kayarvizhy "Reinforcement Learning for Autonomous Driving Scenarios in Indian Roads."

[2]. Kiran, B. R., et al. "Deep Reinforcement Learning for Autonomous Driving: A Survey." *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, 2022, pp. 665-687.

[3]. Bera, A., and Manocha, D. "Reinforcement Learning in Autonomous Driving: A Survey." *ACM Computing Surveys*, vol. 53, no. 4, 2021.

[4]. Saxena, R., and Sharma, D. "AI-Driven Semi-Autonomous Vehicles: A Path Towards Safer Roads in India." *International Journal of Artificial Intelligence and Applications*, vol. 9, no. 2, 2023, pp. 89-103.

[5]. Bhalla, M., et al. "Challenges in Designing Autonomous Vehicles for Indian Roads: A Review." *Journal of Road Transport*, vol. 14, no. 3, 2020, pp. 254-263.

[6]. Kumar, P., & Singh, R. (2023). Application of Machine Learning in Autonomous Vehicles: A Review. *Journal of Intelligent Transportation Systems*, 27(4), 254-270.

[7]. Narayanan, A., et al. (2022). A Deep Reinforcement Learning Approach for Autonomous Driving in Indian Traffic Conditions. *IEEE Transactions on Vehicular Technology*, 71(1), 340-351.

[8]. Sharma, V., & Gupta, S. (2021). Road Infrastructure Challenges in Developing Countries: The Case of India. *International Journal of Transport Development and Integration*, 5(3), 150-165.

[9]. Tanwar, S., & Kumar, N. (2020). Edge Computing in Autonomous Vehicles: Real-Time Data Processing Challenges. *IEEE Network*, 34(2), 23-29.

[10]. ALONZO KELLY AND ANTHONY STENTZ, Rough Terrain Autonomous Mobility—Part 2: An Active Vision, Predictive Control Approach, [Springer] 1998.

[11]. Liang Zhao, Student Member, IEEE, and Charles E. Thorpe, Senior Member, IEEE, Stereo- and Neural Network-Based Pedestrian Detection, [IEEE] 2000.

[12]. Chiung-Yao Fang (Associate Member, IEEE), Sei-Wang Chen(Senior Member, IEEE) and Chiou-Shann Fuh (Member, IEEE), Road-Sign Detection and Tracking, [IEEE] 2003.

[13]. R. MANDUCHI (University of California at Santa Cruz, Santa Cruz), Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation, [Springer] 2005.

[14]. Gabriel M. Hoffmann, Claire J. Tomlin (Department of Aeronautics and Astronautics, Stanford University, Stanford). Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing, [IEEE] 2007.

[15]. Dr. Yalcin Ertekin, Drexel University (Engineering Tech). An Autonomous Arduino-based Racecar for Final-Year Engineering Technology Students, [ASEE] 2014.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG1392

[16]. Victor A. Shia, Yiqi Gao, Ramanarayan Theresa Lin, Francesco Borrelli, and Ruzena Bajcsy, (Fellow, IEEE), Semi-autonomous Vehicular Control Using Driver Modeling, IEEE] 2014.
[17]. Kan Zheng (Senior Member, IEEE), Qiang Zheng, Haojun Yang, Long Zhao, Lu Hou, and Periklis Chatzimisios (Senior Member, IEEE), Reliable and Efficient Autonomous Driving: the Need for Heterogeneous Vehicular Networks, [IEEE] 2015.
[18]. Simon Ulbrich and Markus Maurer, Towards Tactical Lane Change Behavior Planning for Automated Vehicles, [IEEE] 2015.
[19]. Stephen M. Erlien, Susumu Fujita, and Joseph Christian Gerdes, Shared Steering Control Using Safe Envelopes for Obstacle Avoidance and Vehicle Stability, [IEEE] 2015.
[20]. Wilko Schwarting, Javier Alonso-Mora, Liam Paull, Sertac Karaman, Daniela Rus, Parallel Autonomy in Automated Vehicles: Safe Motion Generation with Minimal Intervention, [IEEE] 2017.
[21]. Constantin Hubmann, Marvin Becker, Daniel Althoff, David Lenz and Christoph Stiller, Decision Making for Autonomous Driving considering Interaction and Uncertain Prediction of Surrounding Vehicles, [IEEE] 2017.
[22]. Arnab Saha, Jeet Sanyal, Nousad Mondol, Santanu Ghosh, Obstacle Avoidance & Light Following Robot, [IJASRE] 2017.