IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Self-Driving Car Using Simulator

Sudhanshu Pandit
B.tech-CSE-AI
NIET, Greater Noida, UP

Ankit Kumar
B.tech-CSE-AI
NIET, Greater Noida, UP

Miss. Monika Mehra

Abstract- The Self-Driving Car is one of the most challenging and interesting topics in the field of Artificial Intelligence. Due to rapid technological growth in transportation, self-driving cars became the topic of concern. The main purpose of this project is to use the CNN and train the neural network in order to drive the car in autonomous mode in a simulator environment. Front camera of a car captures the images, and we use those captured images in order to train the model in short, we can say we have used the concept of behavioural cloning. In behavioural cloning, the system tries to mimic the human driving behaviour by tracking the steering angle. That means a dataset is generated in the simulator by a user driven car in training mode, and the deep neural network model then drives the car in autonomous mode

The project focuses on building core capabilities for autonomous vehicles, such as understanding the environment through sensors, making intelligent decisions, and accurately controlling the vehicle. Techniques like deep learning are used for recognizing objects, detecting lanes, and interpreting traffic signs. Sensor data form virtual LiDAR, cameras, and radar is combined for a complete understanding of the surrounding. Additionally, reinforcement learning helps the system adapt to complex driving situations and optimize performance.

By testing under diverse conditions in simulation, the system's reliability, safety, and efficiency are analysed. The findings demonstrate how simulators accelerate innovation in self-driving technology while addressing challenges like transitioning from virtual to real-world deployment of simulation in creating reliable self-driving cars and shaping smarter, safer transportation systems.

Keywords: Self-driving car, CNN, Image Processing, Dataset Generation, Real Time Data, Augmentation Techniques

I. INTRODUCTION

The Self-Driving Car has a very rich domain. There are several factors that affect the driving process such as the speed of the car, obeying the traffic rules, the position of other vehicles on the road, the speed at which those vehicles are operating and many more. The system is not fully aware of all the factors that affect its operation and many at times it is necessary to respond quickly to avoid any untoward incident. It is because of this reason that the domain is one of the most challenging and intriguing in the field of Artificial Intelligence. the state of the environment. To compensate for the lack of computer vision, our system utilizes event generators to generate random events to test the functionality of the system. Our project has two primary goals: Realize the potential of a hybrid system operating in a partially observable, continuous and dynamic environment. Evaluate the ability of the system to take accurate and quick decisions in such an environment. In our project, we have implemented the system using a hybrid approach of Rule based and Case based systems. The Rule based system handles the basic driving tasks such as stopping on a Red signal or a stop sign, adjusting the speed based on the speed limit sign and handling the lane change. The Case Based system handles the exceptional events such as stopping the car when a pedestrian crosses, taking evasive action upon sensing an emergency such as applying the brakes hard or steering away. The combination of the Rule based and cased based system integrates the basic driving task with the ability to take accurate and quick decisions.

Our system takes a flat file as input which contains the directions from source to destination along-with some meta data which describes the route. Our system does not make use of any hardware such as the camera or sensors to obtain the state of the environment. To compensate for the lack of computer vision, our system

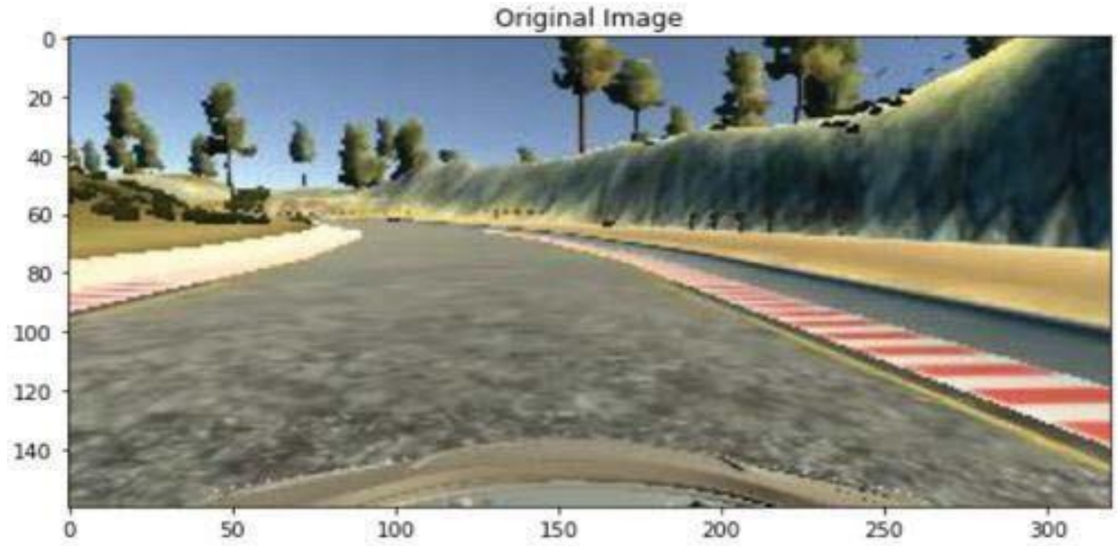
utilizes event generators to generate random events to test the functionality of the system. Our project has two primary goals: Realize the potential of a hybrid system operating in a partially observable, continuous and dynamic environment. Evaluate the ability of the system to take accurate and quick decisions in such an environment.

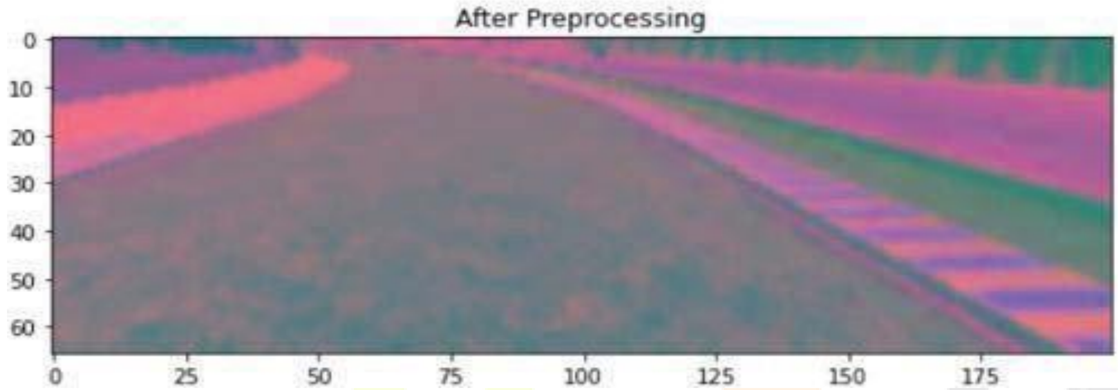
2. RESEARCH METHODOLOGY.

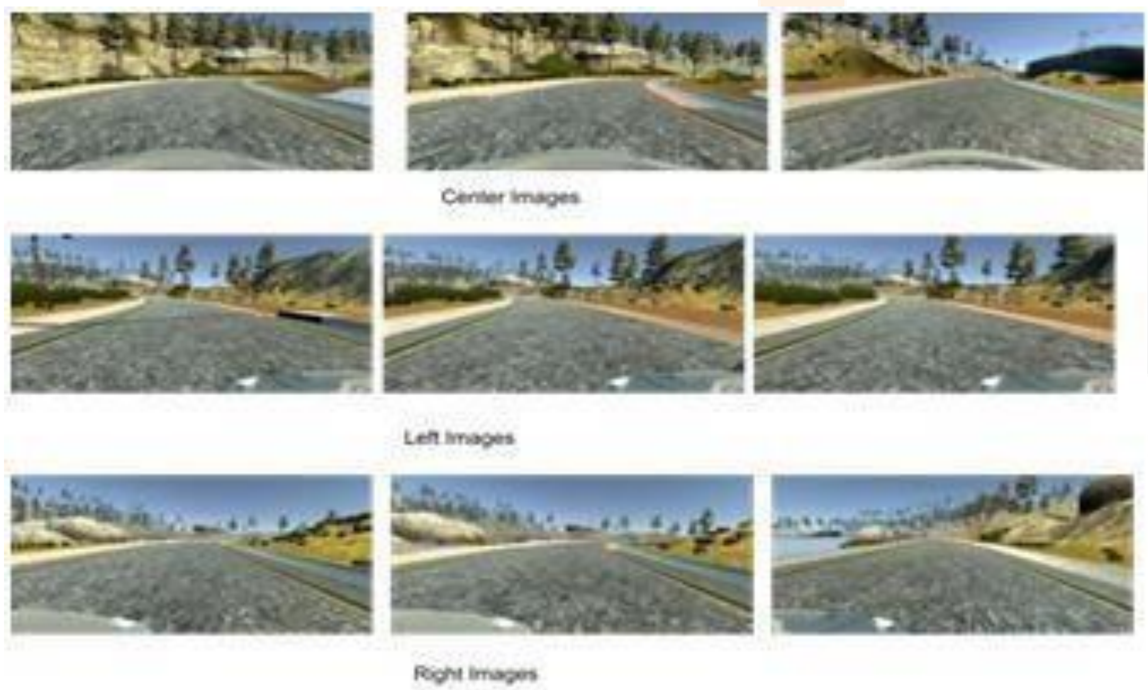
2.1 Data Collection

The data collection process involved utilizing a simulator environment to capture images from the front camera of a car. The simulator provided a realistic virtual driving experience, allowing for the generation of a diverse

dataset representative of various driving scenarios. The car's front camera recorded images as the vehicle traversed through different tracks with varying road conditions, including straight stretches, curves, barriers, heights, and shadows.







2.1 Preprocessing

Prior to training the neural network model, the collected images underwent preprocessing steps to enhance their suitability for training. This preprocessing included normalization to standardize pixel values, resizing to ensure uniform image dimensions, and data augmentation techniques to increase the diversity of the dataset. Augmentation techniques such as random rotation, flipping, and brightness adjustment were employed to simulate real-world variations in lighting and perspective.

2.2 Model Architecture

The architecture of the convolutional neural network (CNN) used in this research consisted .of multiple layers designed to extract relevant features from the input images and make predictions regarding steering angles. The CNN architecture comprised convolutional layers for feature extraction, followed by pooling layers for dimensionality reduction, and fully connected layers for steering angle prediction. Specific configurations, including the number of layers and filter sizes, were determined through experimentation to optimize model performance

2.3 Training Procedure

The training procedure involved feeding the preprocessed images into the CNN model to learn the mapping between input images and corresponding steering angles. The Adam optimization algorithm was employed to minimize the mean squared error loss function, which quantified the disparity between predicted and actual steering angles. A batch size of [specify batch size] and a predefined number of epochs [specify number of epochs] were used during training to iteratively update the model parameters and improve performance.

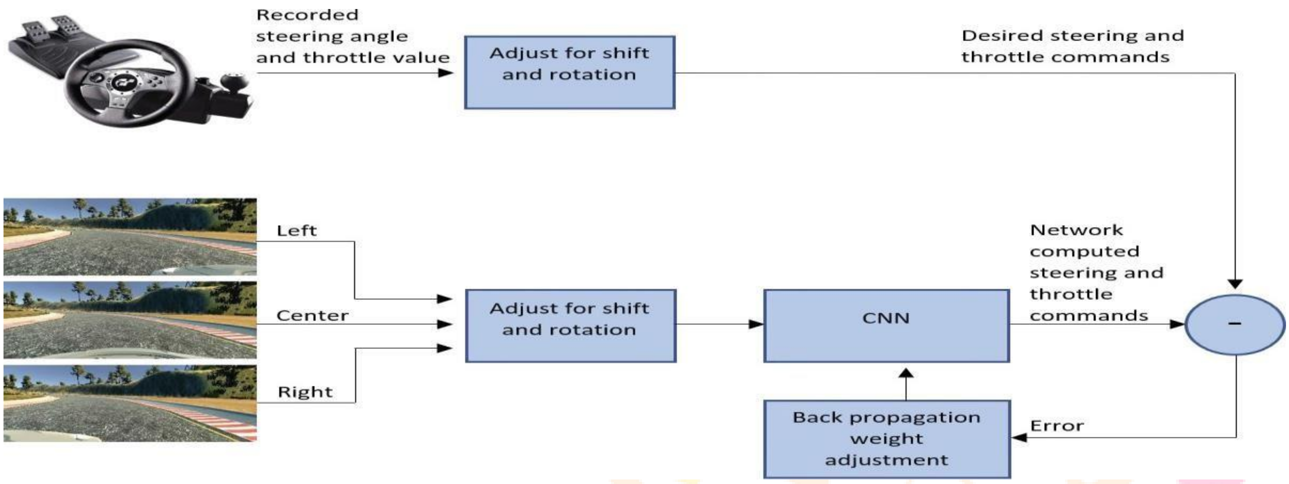
2.4 Validation and Testing

To comprehensively evaluate the effectiveness of the trained model, a systematic approach was adopted, involving the segregation of the dataset into distinct subsets for validation and testing purposes. This division facilitated rigorous scrutiny of the model's performance under various conditions.

The validation dataset played a crucial role in the iterative training process by serving as a means to monitor the model's performance and prevent overfitting. By periodically assessing the model's performance on this subset during training, adjustments could be made to optimize its learning dynamics and prevent it from memorizing noise in the training data. This iterative refinement process ensured that the model's predictive capabilities were honed to generalize well to unseen data.

Subsequently, the testing dataset served as the ultimate litmus test for the model's generalization ability. By evaluating the model on previously unseen data, its capacity to make accurate predictions in real- world scenarios was rigorously assessed. Performance metrics such as accuracy, mean squared error, and root mean squared error were meticulously computed to quantitatively gauge the model's predictive prowess across various evaluation criteria.

This comprehensive evaluation framework provided valuable insights into the model's strengths and limitations, enabling informed decisions regarding its suitability for real-world deployment. Additionally, it facilitated the identification of areas for further improvement, guiding future research efforts aimed at enhancing the robustness and efficacy of autonomous driving systems.



3. System design.

The Self-Driving Car Simulator that we developed follows a Hybrid approach consisting of the Rule Based and Case Based systems. Such an approach allows for a clear division

of tasks between the two systems with the Rule based system handling the basic driving rules and the Case Based system taking care of the exceptional events. Figure 1. shows the architecture of our system.

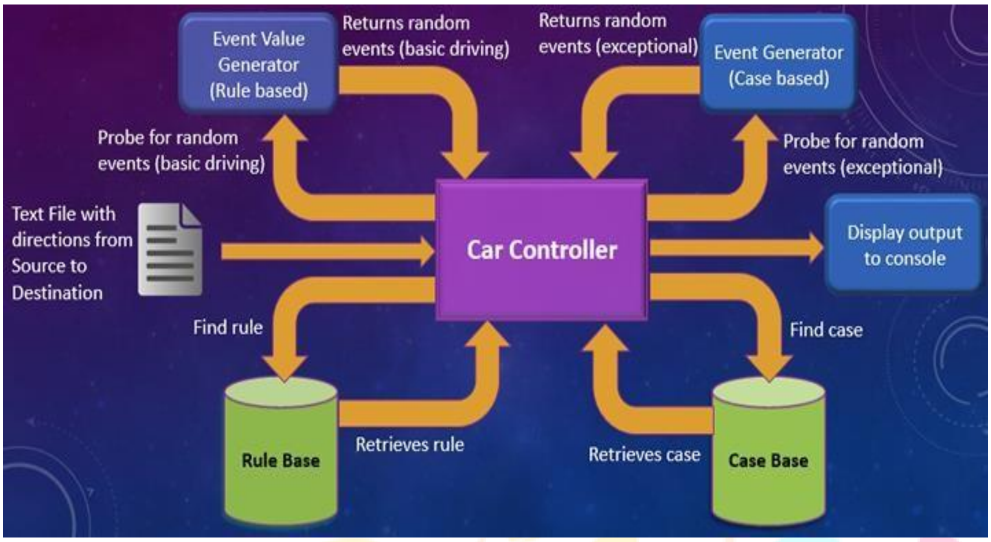


Figure 1: High level system model The high-level overview of the system is as follows: -

The system starts by taking a text file as input which contains the directions from the source to the destination. The file contains meta-data related to the route such as the number of lanes, whether the lane is single-lane or multi-lane, whether a signal or a stop sign is present at the end of the lane, whether the lane has a speed limit and the length of the lane. The event generator generates exceptional events periodically. In case of a random event generated by event generator, the system checks in the case base for similar rules and retrieves the case with a similarity greater than a pre-defined threshold. The actions taken by the system is output to the console.

The components of the system are as follows: -

i. **Car Controller:** The car controller is the core component which drives the system. It processes the lane related information, one lane at a time and co-ordinates the activities between the Rule based and the Case Based system.

ii. **Event Value Generator:** It is named so because it generates values corresponding to the events and not the events. For example, if the signal is visible to the car (which is determined when the car crosses a threshold distance on that lane), the event value generator outputs a random signal value (RED, AMBER or GREEN). If the event value generator is probed again for a value, it outputs the current value based on the previous value. For example, if the previous value output was RED, then the current value will either be RED or GREEN and not AMBER.

iii. **Event Generator:** It generates random exceptional events to test the functionality of the case- based system. The event generated by the generator contains information such as the object (pedestrian, vehicle etc.), distance from the car, speed of the object and the direction with respect to the car.

iv. **Rule Base:** It contains the rules to be matched based on the condition.

v. **Case Base:** It contains the pre-loaded cases which will be assessed for similarity with the random case generated by the event generator

It is to be noted that the Controller probes the Event Value Generator for random values corresponding to basic driving events. The data related to these events are available in the lane information. Based on the random values provided by the generator, the rule base is checked, and the corresponding action is returned. For exceptional events, the Event Generator generates random cases; the Controller never probes the Event Generator. So when the Event Generator produces the events, the task is directed to the Case Base System.

Rule Based System:

The Rule based system consists of the Controller, the Event Value Generator and the Rule Base. The car controller operates based on the data present in the file. The controller starts the car simulator and for each related lane data in the text file, the controller controls the simulator accordingly. For example, if there is a STOP sign at the end of the lane, the car upon reaching a threshold distance (braking distance), would check the rule base for the corresponding rule and perform the necessary action, in this case, to stop the car. The controller would then probe the event value generator periodically until an" all clear" value is provided by the generator.

Case Based System:

The Case based system consists of the Controller, the Exceptional Event Generator and the Case Base. The exceptional event generator generates events from a random set of events. To avoid cold start problems, the case base is pre- loaded with cases to start with so that it can handle the random events generated by the event generator. As accurate and quick responses are required in case of an exceptional event, the case- based system assesses the similarity between a candidate case and the exceptional event until the first case which is more than 90% similar to the random event is searched. The case base might contain cases with better similarity, however, as quick response is needed, a trade-off between accuracy and retrieval time is employed. If no cases with similarity greater than the minimum threshold of 50% similarity is found, then the system defaults to applying the brakes to stop the car in the minimum possible time. The evaluation section discusses the time taken to search the entire case base and if the car can successfully handle such an exceptional event (an event for which no satisfactorily matching case is found in the case base). Each matching case retrieved from the case base goes to the modifier part of the case-based system. The modifier modifies the" priority" and required final speed of the car depending on the current state of the car to the controller. The" priority" is rate of acceleration/deceleration which indicates the degree by which speed of the car should be increased or decreased to handle the event.

Why a Hybrid System?

vi. The main advantage of the hybrid system approach is that there is a clear division of tasks between the rule based and case-based systems. This allows the events to be directed towards either the rule based or the case-based system de- pending on whether the events are basic driving rules or exceptional events. The second benefit of this approach is that the search space has been reduced. Had the exceptional events been also included in the rule base, then the number of rules would have increased. As a direct consequence of the reduced search space, the rule/case retrieval time is reduced. The hybrid approach also enabled efficient knowledge representation. Representing the cases as rules would have resulted in a large number of antecedents for the rules and would have also resulted in long chains. The hybrid approach helped to avoid such a scenario. Finally, the approach enables the system to give compelling explanations for its actions.

4. Future Directions.

The current research lays a solid foundation for advancing autonomous driving technology, yet there are several avenues for future exploration and improvement. Here, we outline comprehensive directions for future research

1. Real-Time Deployment:
- Transitioning the developed model from simulation to real-world application is paramount. Integrating the trained neural network into an actual self-driving vehicle and evaluating its performance in real- time driving scenarios will provide invaluable insights into its practical viability and effectiveness.

2. Robustness Enhancement:
- Enhancing the model's robustness to various uncertainties, including adverse weather conditions, changing road surfaces, and unexpected obstacles, is critical for ensuring safe and reliable autonomous driving across diverse environments. Investigating robust optimization techniques and sensor fusion strategies can fortify the model against these challenges

3. Dynamic Environment Adaptation:
- Developing adaptive algorithms that enable the model to dynamically respond to changes in the driving environment, such as construction zones, temporary road closures, and pedestrian crossings, will bolster its versatility and responsiveness. This may involve incorporating reinforcement learning mechanisms or advanced planning algorithms capable of handling dynamic scenarios.

4. Multi-Agent Interaction:
- Exploring methods for enabling autonomous vehicles to interact intelligently with other agents, including vehicles, pedestrians, and cyclists, in shared spaces is essential for enhancing safety and efficiency in urban driving scenarios. Research in cooperative decision-making, negotiation strategies, and communication protocols can facilitate harmonious interactions among multiple agents on the road.

5. Continual Learning Frameworks:
- Implementing continual learning frameworks that enable the model to adapt and improve over time through exposure to new data and experiences is crucial for ensuring continual refinement and optimization of autonomous driving capabilities. This involves developing algorithms capable of incremental learning, knowledge transfer, and adaptation to evolving environments.

6. Hardware Optimization:
- Optimizing the hardware architecture and computational efficiency of onboard systems is imperative for meeting the computational demands of real-time processing and decision- making in autonomous vehicles. Research in efficient neural network architectures, hardware acceleration techniques, and energy- efficient computing platforms can facilitate scalable and cost-effective deployment of autonomous driving systems.

7. Ethical and Regulatory Considerations:
- Addressing ethical and regulatory challenges associated with autonomous driving, such as liability assignment, privacy protection, and safety standards compliance, necessitates interdisciplinary collaboration and policy development. Research in ethical decision- making frameworks, transparent accountability mechanisms, and regulatory frameworks for autonomous vehicles can ensure responsible deployment and societal acceptance.

5. Result

Lane marking detection, semantic abstraction, track planning are the methods which the CNN algorithm grasps. Atmospheric conditions vary from sunny, brisk wind, snowy to rainy. In these situations, limited training data is adequate to train the car. This information can also be used to train the car to travel on highways, local and residential roads. Sparse training is a method which is particularly used for steering, in this simulation and from which CNN algorithm Lane marking detection, semantic abstraction, track planning are the methods w CNN algorithm grasps. Atmospheric conditions vary from sunny, brisk wind, snowy to rainy. In these situations, limited training data is adequate to train the car. This information can also be used to train the car to travel on highways. Sparse training is a method which is particularly used for steering, in this simulation and from which CNN algorithm learns essential road features.

6. Conclusions

In this paper, we have built two different CNN architecture models and tested them on the simulator provided by Udacity. Both the architectures behave quite differently despite being from the same family of CNN. With an accuracy of 96.83%, model 'A' performed well in the simulation. Model 'B' obtained an accuracy of 76.67%. From the above test results, we can conclude that CNN architecture has been found to be helpful to predict the steering angle according to the track.

In comparison with both the models, model 'A' completed the track autonomously. Whereas we observed that model 'B' was wobbling along the track and eventually got off-track in the simulator multiple times As safety of user is the priority of self-driving cars, this model may not withstand some parameters of safety. But, by optimizing and modifying model A' and by hybridizing it with cutting edge technologies, this architecture can contribute in real-world application.

7. References

[1] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, 2021, pp. 1-6, doi: 10.1109/ICEngTechnol.2021.8308186.

[2] Bojarski, Mariusz & Testa, Davide & Dworakowski, Daniel & Firner, Bernhard & Flepp, Beat & Goyal, Prasoon & Jackel, Larry & Monfort, Mathew & Muller, Urs & Zhang, Jiakai & Zhang, Xin & Zhao, Jake & Zieba, Karol. (2020). End to End Learning for Self-Driving Cars.

[3] Dr Narapareddy Ramarao, B Vivek Bhat, Kartik Kulkarni, Ashley, Raban Akbary "Lane Detection for Autonomous Vehicle " Vol. 9 - No. 3 (March 2022), International Journal of Engineering Research and Applications (IJERA) , ISSN: 2248-9622 ,

www.ijera.com

[4] D. Dong, X. Li and X. Sun, "A Vision-Based Method for Improving the Safety of Self-Driving," 2023 12th International Conference on Reliability, Maintainability, and Safety (ICRMS), Shanghai, China, 2018, pp. 167-171, doi: 10.1109/ICRMS.2023.00040.

[5] T. Okuyama, T. Gonsalves and J. Upadhay, "Autonomous Driving System based on Deep Q Learnig," 2022 International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 2022, pp. 201- 205, doi: 10.1109/ICoIAS.2018.8494053.

[6] A. Mogaveera, R. Giri, M. Mahadik and A. Patil, "Self Driving Robot using Neural Network," 2023 International Conference on Information
, Communication, Engineering and Technology (ICICET), Pune, 2023,

pp. 1-6, doi: 10.1109/ICICET.2018.8533870.

[7] J. Barrozo and V. Lazcano, "Simulation of an Autonomous Vehicle Control System Based on Image Processing," 2022 5th International Conference on Frontiers of Signal Processing (ICFSP), Marseille, France, 2022, pp. 88-94, doi: 10.1109/ICFSP48124.2019.8938094.

[8] V. Swaminathan, S. Arora, R. Bansal and R. Rajalakshmi, "Autonomous Driving System with Road Sign Recognition using Convolutional Neural Networks," 2024 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2024, pp. 1-4, doi: 10.1109/ICCIDS.2019.8862152.

[9] Z. Qiumei, T. Dan and W. Fenghua, "Improved Convolutional Neural Network Based on Fast Exponentially Linear Unit Activation Function," in IEEE Access, vol. 7, pp. 151359-151367, 2021, doi: 10.1109/ACCESS.2019.2948112.