iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72769

Robust Lane Detection for Autonomous Vehicles Using Deep Learning Techniques in the UDACITY Simulator

Arnav Patil[1], Atharv Waghmale[2], Siddheya Masurkar[3], Vedant Gogawale[4]
[1,2,3,4]*Electronics and Telecommunication Department, Pune Vidyarthi Griha's College of Engineering, Technology and Management, India*

Abstract: This paper presents a deep learning based solution for real-time lane detection in autonomous vehicles using UDACITY self-driving car simulator. Our goal is to navigate the car on a road safely and independently by training the images captured from the three cameras mounted on a virtual car using the Convolutional Neural Networks (CNN). These images are paired with steering angles and used to teach the model for responding to different road edges and lane positions. The model is trained in training mode and then implemented to drive on its own in the autonomous mode of simulator. We have applied several pre-processing techniques in order to enhance the performance and robustness of the system. The results show that the system can make accurate driving decisions even in tough scenarios, highlighting the power of computer vision and deep learning in building smarter self-driving technologies.
Keywords: Autonomous vehicles, UDACITY simulator, Convolutional Neural Networks (CNN), deep learning, self-driving car, pre-processing.

I. INTRODUCTION

A. Background on Autonomous Vehicles

Autonomous vehicles (AVs), commonly referred to as self-driving cars, are rapidly changing our approach towards modern development. They aim to minimize the human error, lower the number of road accidents, and make traffic flow more efficient. To drive themselves, these vehicles rely on a combination of different technologies like cameras, sensors, actuators, LiDAR, GPS, and logical algorithms. By integrating these systems an autonomous vehicle understands its surroundings, environments and make decisions so as to move on the road safely. Among all these capabilities, one of the most important aspect for a self-driving vehicle is the ability to recognize and understand the road environment accurately and in real time.
This document is a template.

B. Importance of Lane Detection in Autonomous Driving:

Lane detection is a fundamental task in autonomous driving, crucial for maintaining the vehicle's position on the road, enabling lane keeping, lane exceed warnings, and path planning. It allows the vehicle to stay within its driving boundaries and adapt to road geometry. Without reliable lane detection, the overall safety and performance of the autonomous vehicle can be significantly compromised, especially during high-speed scenarios, turning and lane changes.

C. Challenges in Real-World Lane Detection

Although we see lot of advancements in this technology, there are scenarios all around the world where the lanes on the roads are not marked properly or they are faded, blocked by the objects on the road, sharp turns, also during night times there are changes of street lights fluctuating, it becomes difficult to predict objects or lane during fog or rainy seasons. Due to all these unpredictable factors sometimes it becomes difficult for proper lane detection.

D. Objective of Research

The main objective of this research is to develop a real-time lane detection system using computer vision techniques that can accurately detect lanes even when they are not clearly visible or during challenging conditions such as poor lighting or bad weather.

II. LITERATURE SURVEY

Autonomous vehicles bring a wide range of benefits that have the potential to transform how we travel. One of the most important advantages is safety, these vehicles are designed to reduce road accidents caused by human errors such as over speeding, distracted behaviour and drunk driving. As they rely on advanced sensors and systems, they can make quicker and more accurate decisions than a human driver in many situations. Also autonomous vehicles helps in improving traffic efficiency, helps in mining and military work enhancing safety of humans [1].

To implement the autonomous vehicle, computer vision applications like image segmentation plays a major role. Image segmentation is improved by deep learning. Also the deep learning models like CNN, Mask R-CNN have advanced the field by using smart design approaches such as encoder-decoder structures, multiscale feature processing. The challenges that arises are high computational requirements, large amount of labelled data and difficulties in segmenting complex scenes [2].

The lane detection models perform well in ideal conditions, but their accuracy drops in bad weather, poor lighting or when the lane markings are faded or occluded. These issues make it hard for systems to detect lanes reliably. Also many algorithms struggle with real-time performance because of heavy processing requirements [3].

To solve these issues of lane detection, a prototype based approach for testing of self-driving cars can be used which includes the integration of camera modules and sensors that navigate the vehicle and provides an improved performance. For this purpose, a 3D virtual environment can be created which includes vehicles, obstacles and the primary objective should be that our vehicle must navigate smoothly through this environment by following the lane and avoiding collision with the obstacles [4].

III. METHODOLOGY

In this paper, we have used UDACITY simulator to drive the autonomous vehicle. This simulator provides a realistic virtual environment to train, test and run the vehicle autonomously. It basically has 2 modes of operation, first is the Training Mode and second is the Autonomous Mode.

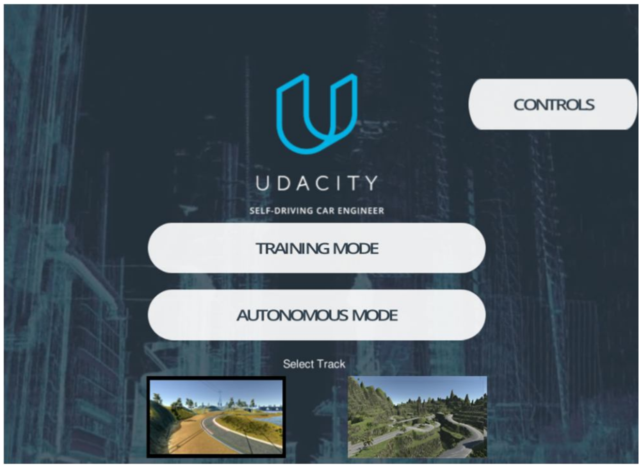


Fig. 1 UDACITY Simulator Mode Selection

A. Training Mode

Here the user or trainer manually drives the car using the forward (↑), backward (↓), left side (←) and right side (→) keys on the keyboard. During this drive, the simulator captures the data i.e. images of the road and lane with the help of 3 camera modules mounted at the centre on front hood, left hand side and right hand side mirrors of the vehicle. Also steering angle is captured during this process. Now these captured images are used to train the deep learning models.

B. Autonomous Mode

Here the simulator connects with the Python server using drive.py command and receives the steering and throttle commands from the trained model. Testing of the model is done in order to enhance the accuracy of the system. In this way the car drives autonomously based on the input.

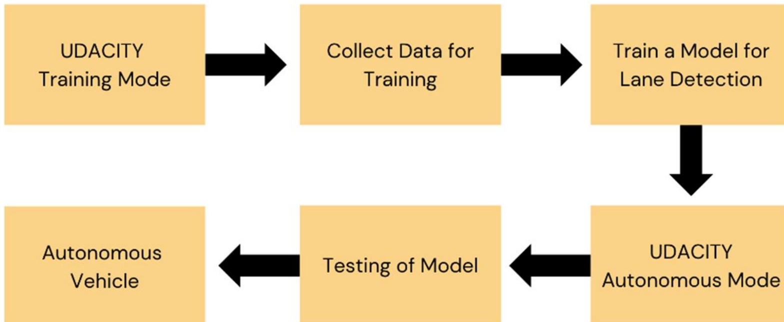


Fig. 2 Block Diagram of Autonomous Vehicle

This is the block diagram of the entire process to drive the autonomous vehicle. It includes,

1) *UDACITY Training Mode:* As discussed earlier, this is the mode where we train the car by controlling it with arrows on the keyboard.
2) *Collect Data for Training:* In this stage, the images are continuously captured using the cameras that are mounted on the vehicle at centre on front hood, right side mirror and left side mirror. Apart from these images, the corresponding steering angles for various road turns are also recorded. All the collected images and data are stored in a particular folder and a driving_log.csv file.

Following are the images that are captured during the training mode of UDACITY simulator-

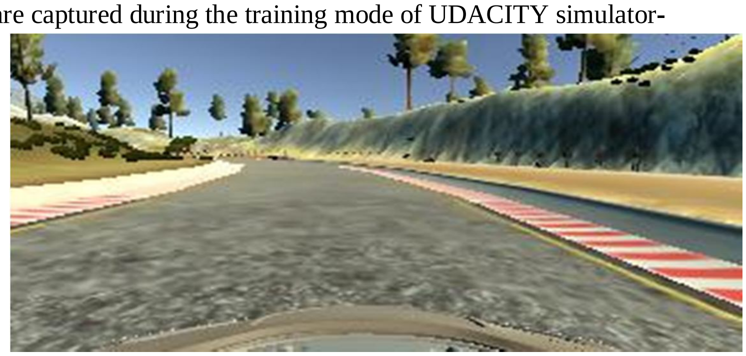


Fig. 3 Image captured from centre camera

Fig. 4 Image captured from right camera

Fig. 5 Image captured from left camera

Following is the driving_log.csv file which is generated during the training mode of UDACITY simulator-

	A	B	C	D	E	F	G	H
1	Image path Camera 1	Image path Camera 2	Image path Camera 3	Steering angle	Trottle Value	Deceleration Value	Vehicle Speed	
2	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.6497864	
3	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.6279418	
4	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.6229097	
5	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.6191623	
6	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.6154376	
7	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.6105061	
8	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.6068335	
9	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.6019711	
10	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0	0	0.59835	
11	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0.02400588	0	0.6206543	
12	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0.2027734	0	0.7070002	
13	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	0.3826937	0	0.9467989	
14	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.05	0.6427272	0	1.434013	
15	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.25	0.8633261	0	2.173052	
16	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.4	1	0	2.864847	
17	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.5354305	1	0	3.791584	
18	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.4319619	1	0	4.489107	
19	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.1909571	1	0	5.422441	
20	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.00613982	1	0	6.111838	
21	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	1	0	7.026899	
22	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	1	0	7.709986	
23	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	0	1	0	8.614678	
24	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.05	1	0	9.288085	
25	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.25	1	0	10.16289	
26	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.4176539	1	0	10.792	
27	C:\Users\Amer\Deskt	C:\Users\Amer\Deskto	C:\Users\Amer\Deskt	-0.148133	1	0	11.50283	

I. Driving Log File generated

C. Train a model for Lane Detection:

The major steps involved to train a model for lane detection includes-

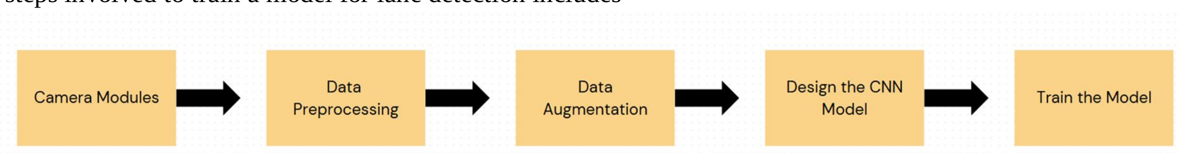


Fig. 6 Block Diagram for Lane Detection

The first step required for training a model is to have the datasets i.e. images which we have already collected from the camera modules mounted on the car in the simulator. The next and the most important step is the Data Pre-processing and Augmentation. It is used to improve the quality and diversity in the dataset. This includes resizing of the image, flipping of image and brightness adjustment to simulate the different driving conditions.

1) *Resizing of Image* – Resizing of image means changing its dimensions by width and height to a particular size which helps to improve training efficiency and performance.

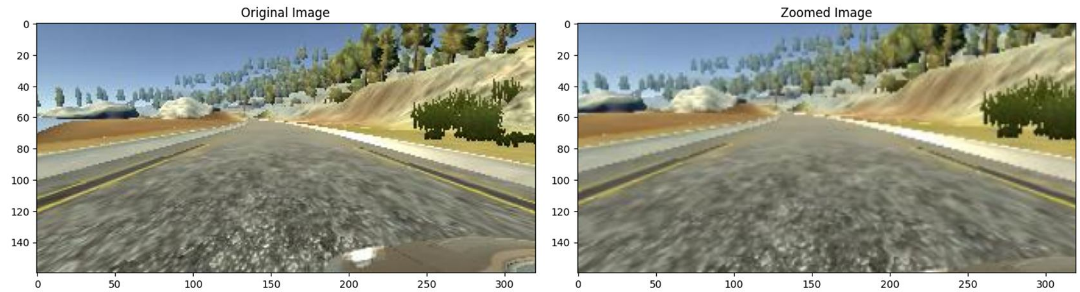


Fig. 7 Resizing of Image

2) *Flipping of Image* **-** Flipping of image is a technique used to increase the diversity of training dataset. Generally horizontal flipping is used which creates mirror image of original one.

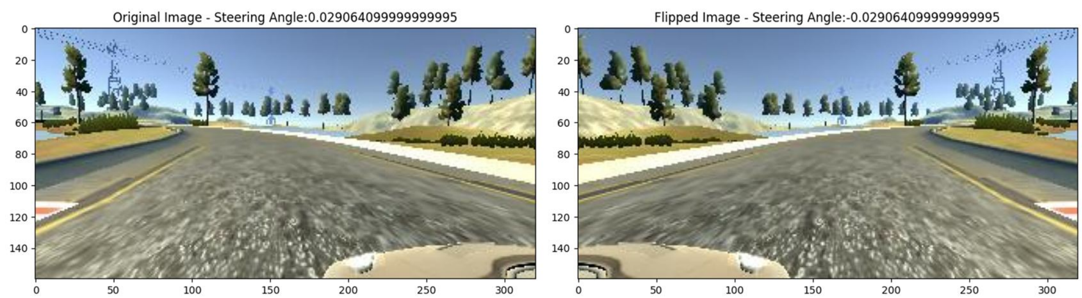


Fig.8 Flipping of Image

3) *Brightness Adjustment of Image* **-** Here the pixel intensity values of image are increased or decreased which helps the model to become more robust to changes in ambient light when detecting lanes.

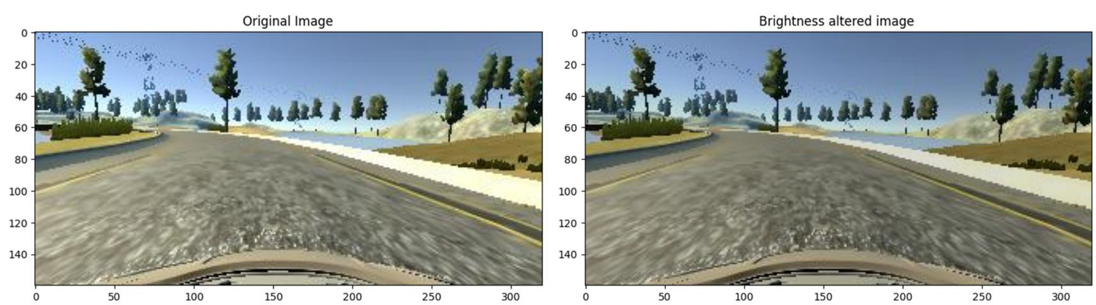


Fig. 9 Brightness Adjustment of Image

After applying all the image augmentation processes we get the final pre-processed image which clearly defines the lane to be followed by the vehicle.

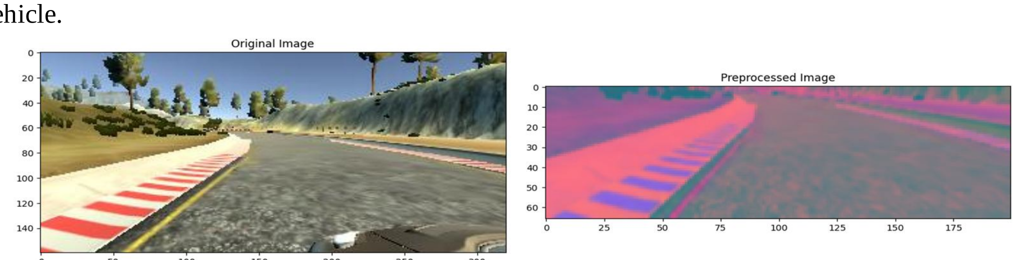
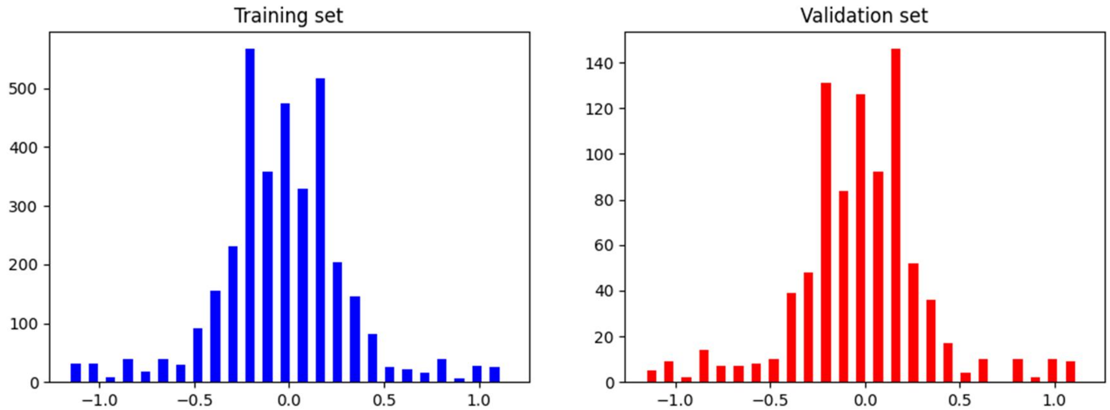


Fig. 10 Pre-processed Image

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Before designing a CNN model, it is important to understand how the steering angles are distributed. Following are the histograms that show how often different steering angles appear in the training and validation sets. This helps to spot any imbalance like too many straight-driving samples and ensures the data is well-prepared for training.



II. Histogram of Training Set and Validation Set

Here X-axis represents the steering angle values and Y-axis represents the number of samples for each steering angle range for both the training and validation sets. The negative values represent left turn for vehicle and positive values of X-axis represent right turn taken by the vehicle and 0 is for straight driving, so change in the direction helps model to drive in different scenarios and not just the straight line. As we can see that there is large value in straight direction as well at 0, so to balance this we use the Data augmentation processes.

Now the most important step is designing a Convolutional Neural Network (CNN) model based on all the above processes and outcomes. All the images that are pre-processed and augmented are passed as input to the CNN model, which learns to predict the correct steering angle for the vehicle. The CNN extracts important features such as lane lines and road edges through the layer of convolution, pulling and activation. A widely used architecture like NVIDIA's model is adopted, where the final output layer gives a single value representing the steering angle. The model is trained using Mean Squared Error as the loss function and the Adam optimizer, using the collected images and corresponding steering angles. Once trained, the model receives real-time images from the simulator in autonomous mode, predicts the steering angle, and sends it back to control the car, allowing it to navigate lanes smoothly. This enables the vehicle to adapt to different road scenarios and lighting conditions effectively.

Below is the CNN model Architecture that takes an image as input and predicts a single steering angle value.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 31, 98, 24)	1,824
conv2d_1 (Conv2D)	(None, 14, 47, 36)	21,636
conv2d_2 (Conv2D)	(None, 5, 22, 48)	43,248
conv2d_3 (Conv2D)	(None, 1, 18, 64)	76,864
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 100)	115,300
dense_1 (Dense)	(None, 50)	5,050
dense_2 (Dense)	(None, 10)	510
dense_3 (Dense)	(None, 1)	11

Total params: 264,443 (1.01 MB)
Trainable params: 264,443 (1.01 MB)
Non-trainable params: 0 (0.00 B)
None

III. CNN Model Architecture

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

This model consists of convolutional layers, conv2d, conv2d_1, conv2d_2 and conv2d_3 that extracts the road features and then uses the dense layers, dense, dense_1, dense_2 and dense_3 to decide the most appropriate steering action. It basically has 1.01 MB of trainable weights that can train upto 2,64,443 images or datasets which indirectly makes it efficient for real-time autonomous driving in the simulators.

D. UDACITY Autonomous Mode:

In this mode, the simulator takes help of the trained models and communicates with the python server using drive.py. This uses the model's predictions to drive or steer the vehicle without any human involvement making the vehicle autonomous.

E. Testing of Model:

The testing of model is done on different tracks keeping in mind the various real-world scenarios and thus checking the model's ability to stay in lane and make correct steering decisions.

F. Autonomous Vehicle:

After undergoing extensive testing, data processing, and model refinement, the autonomous vehicle is finally ready for deployment. This readiness comes as a result of rigorous evaluation of the trained model in various simulated driving conditions to ensure it can accurately interpret lane markings, make real-time steering decisions, and respond appropriately to different road scenarios. Multiple iterations of training, validation, and testing were carried out to optimize the performance of the neural network and minimize errors.

IV. RESULT AND ANALYSIS

After implementing all the deep learning techniques and training the model in Training Mode of UDACITY simulator, we get the following results by running the model in Autonomous Mode. The below figures demonstrate the trained CNN model. Here as we can see, the steering angle is dynamically adjusted based on the continuous predictions made by the three cameras in front of the vehicle. The output also displays the steering angle in (º) and speed of the vehicle in (mph). Thus the vehicle performs successfully in Autonomous Mode and predicts its environment accurately.

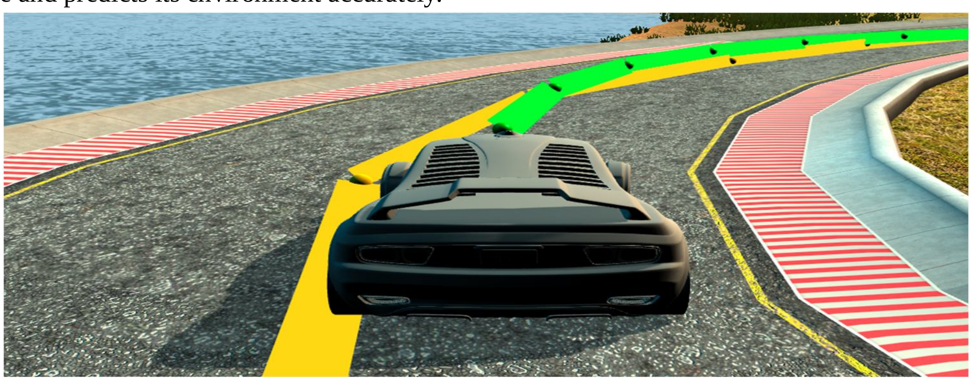


Fig. 11 Trajectory Visualization for Lane Detection

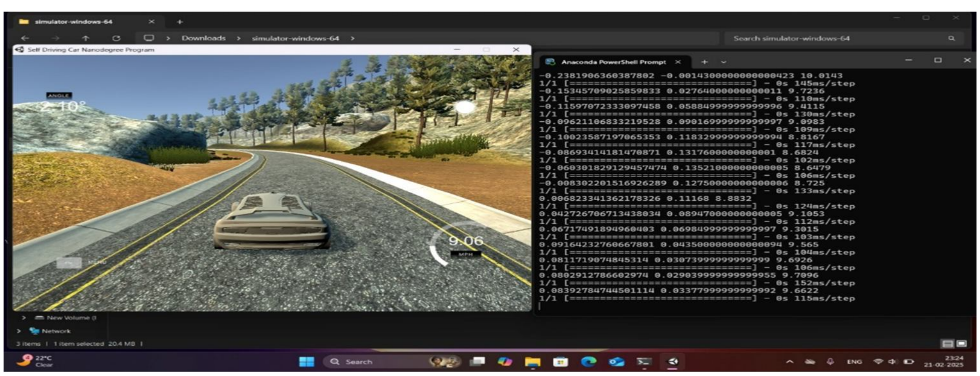


Fig. 12 Real-time Autonomous Vehicle Driving

V. CONCLUSIONS

We successfully developed the real-time lane detection system for autonomous vehicles using computer vision and deep learning algorithms. With the help of UDACITY simulator, we collected the image datasets and trained the Convolutional Neural Network (CNN) model to predict the steering angle based on road curves.

Also implemented the key pre-processing steps like image resizing, flipping and brightness adjustment that enhanced the model robustness for different types of environment and also for various road conditions.

With the help of real-time testing in the simulator, it is clear that the vehicle can navigate the edges of road and maintain the lane alignment consistently and efficiently. Based on our study, it is clear that by implementing this technique of making the vehicle autonomous can help improve road safety, decision making ability, reduce traffic congestion, effective reduce the pollution and support for smart cities enabling the data-driven urban planning and efficient public transportation.

This research can be further expanded by integrating more sensors like LiDAR, GPS to enhance the perception in complex environments. Also by real-world testing on a prototype vehicle can significantly improve the system's capabilities.

VI. ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to Prof. T. Khatavkar ma'am of Pune Vidyarthi Griha's College of Engineering, Technology and Management, for her support, encouragement and guidance throughout the research. Her insightful suggestions and motivation played a crucial role in helping us understand and grow during this learning experience.

REFERENCES

[1] F. Khan, R. L. Kumar, S. Kadry, Y. Nam, and M. N. Meqdad, "Autonomous vehicles: A study of implementation and security," International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 4, pp. 3013–3021, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3013-3021.

[2] S. Minaee, Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, July 2022, doi: 10.1109/TPAMI.2021.3059968.

[3] N. J. Zakaria, M. I. Shapiai, R. A. Ghani, M. N. M. Yassin, M. Z. Ibrahim, and N. Wahid, "Lane Detection in Autonomous Vehicles: A Systematic Review," *IEEE Access*, vol. 11, pp. 3729–3765, Jan. 5, 2023, doi: 10.1109/ACCESS.2023.3234442

[4] B. Barua, C. Gomes, S. Baghe, and J. Sisodia, "A Self-Driving Car Implementation using Computer Vision for Detection and Navigation," in *Proceedings of the IEEE International Conference on Computing, Communication & Automation (ICCCA)*, Greater Noida, India, May 2019, pp. 1–6, doi: 10.1109/ICCCA.2019.9065627.

[5] https://youtube.com/playlist?list=PLkz_y24mlSJawbZz-fJrrxZi0QNmJ5aP6&feature=shared

[6] S. Minaee, Y. Boykov, Porikli, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press, 2021, doi: 10.1109/TPAMI.2021.9356353

[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," *arXiv*, vol. abs/1310.1531, Oct. 2013.

[8] M. V. Rajasekhar and A. K. Jaswal, "Autonomous vehicles: The future of automobiles," *2015 IEEE International Transportation Electrification Conference (ITEC)*, Chennai, India, 2015, pp. 1-6, doi: 10.1109/ITEC-India.2015.7386874.

[9] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *International Journal of Artificial Intelligence Research*, vol. 2, no. 1, p. 22, Jun. 2018, doi: 10.29099/ijair.v2i1.42.

[10] X. Li and Y. Shi, "Computer Vision Imaging Based on Artificial Intelligence," *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, Hunan, China, 2018, pp. 22-25, doi: 10.1109/ICVRIS.2018.00014.

[11] S. Lade, P. Shrivastav, S. Waghmare, S. Hon, S. Waghmode and S. Teli, "Simulation of Self Driving Car Using Deep Learning," *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, India, 2021, pp. 175-180, doi: 10.1109/ESCI50559.2021.9396941.

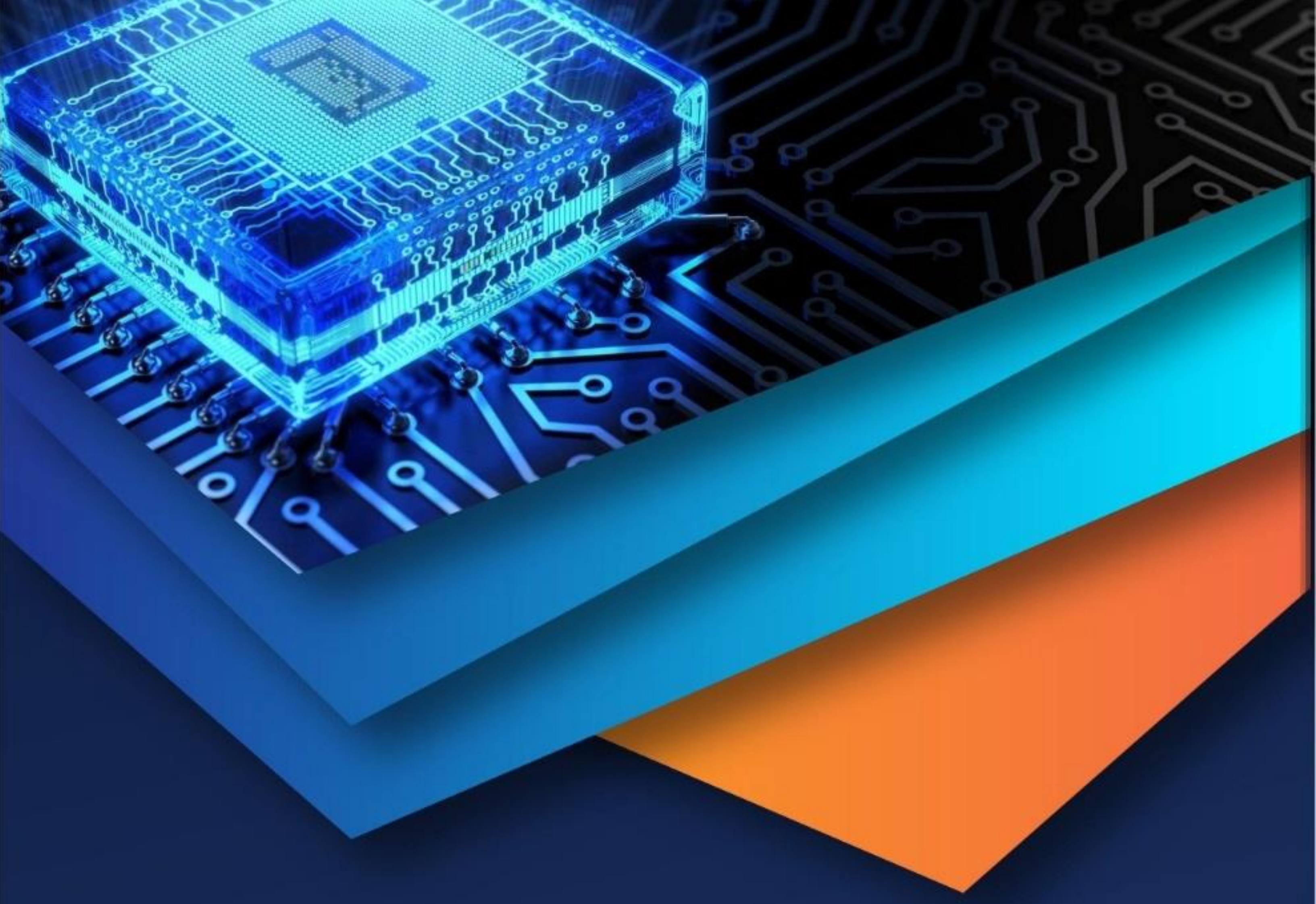
[12] M. Felisa and P. Zani, "Robust Monocular Lane Detection in Urban Environments," *IEEE Xplore*, 2010. [Online].

[13] N. Hortoványi, "Model Predictive Controller Project," *Nick Hortoványi's Blog*, Jun. 6, 2017. [Online].

[14] S. B. S. and R. Deepu, "Deep learning assisted active net segmentation of vehicles for smart traffic management," *Global Transitions Proceedings*, vol. 2, pp. 282–286, 2021. [Online].

[15] Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN COMPUT. SCI. 2, 420 (2021).

[16] Z. Hao, "Deep learning review and discussion of its future development," in *MATEC Web Conf.*, vol. 277, 2019, Art. no. 02035, doi: 10.1051/matecconf/201927702035.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY