IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Self-Driving Vehicle Using Deep Learning

[1]Sanchit Tate, [2]Aditya Raj Singh, [3]Gourish Dalvi, [4]Harsh Shah, [5]Sudheer Hirolikar

[1]Student, [2]Student, [3]Student, [4]Student, [5]Professor

[1]Department of Automobile Engineering,

[1]Dhole Patil College Of Engineering, Pune, India

Abstract: We developed a convolutional neural network (CNN) to convert front-facing camera data straight from raw pixels to steering commands. This end-to-end strategy turned out to be unexpectedly effective. The machine learns to drive in traffic on local roads with or without lane markings, on motorways, and with the least amount of training data from people.

By using simply the human steering angle as the training signal, the system automatically picks up internal representations of the necessary processing stages, such as spotting useful road elements. For instance, we never expressly trained it to recognize the outline of a road.

Our end-to-end approach simultaneously optimizes all processing steps in contrast to explicit decomposition of the problem, such as lane marking detection, path planning, and control. We contend that doing so will eventually result in bigger systems and better performance. The internal components will self-optimize to maximize system performance rather than improving intermediate criteria chosen by humans, such as lane detection, leading to better performance. It makes sense that these criteria be chosen for simplicity of human interpretation, but this does not imply that they will always result in the best system performance. Because the system learns to solve the problem with the fewest amount of processing steps possible, smaller networks are made possible.

For training, we used an NVIDIA Dev Box and Torch 7, and for navigation, we used an NVIDIA DRIVETM PX self-driving car computer running Torch 7.

Index Terms **- Simulation, Model Creation, Deep Learning, Self Driving.**

I.INTRODUCTION

Because of how quickly technology is developing, humanity has been waiting for several decades for self-driving cars. Recently, Tesla became the first company to provide the concept on a commercial scale.

One of the key technologies that made self-driving possible is deep learning. It is a flexible tool that can address practically any issue.

In this project, we'll concentrate on convolutional neural networks (CNN) and deep learning techniques for self-driving automobiles. These systems primarily employ CNN to identify and categorize various road segments and to arrive at the best judgements.

We'll see along the way how NVIDIA uses CNN algorithms to create autonomous or driverless vehicles.

The driverless car is often referred to as a robot car, an autonomous vehicle, or a self-driving car (SDC). The goal of an autonomous vehicle is to operate without a driver. The Self Driving Car can lower the cost of driving while simultaneously enhancing various aspects of mobility, such as road safety. According to McKinsey & Company, a realistic estimate of a 90% reduction in crash rates suggests that the widespread usage of robotic cars in the US may result in yearly savings of up to $180 billion in healthcare and automobile maintenance alone.

A vehicle with sensors and cameras for detecting its surroundings and the ability to travel without real-time input from a human is called a driverless automobile. Engineers that create self-driving cars utilize two methods: robotics, which involves analyzing data from several sensors and making a decision, and deep learning, which teaches the car to drive by mimicking human driving behavior.

The primary components of an autonomous vehicle are computer vision, which allows the vehicle to understand its surroundings, sensor fusion, which combines data from multiple sensors to help the vehicle make decisions, localization, which enables the vehicle to be located in its surroundings, path planning, which helps the vehicle navigate from point A to point B, and control steering of car.

Using an open source self-driving car simulator developed by Udacity, the project aims to leverage deep learning and computer vision techniques to assist the autonomous vehicle in navigating a closed area. This project also uses computational neural networks (CNNs), a deep learning method, to classify traffic signs and detect lanes using computer vision.

II. LITERATURE SURVEY

[1] Through this paper one can learn that for the safety of lane maintenance and cruise control functions in self-driving cars, a technique for determining the best mix of sensors, methodologies, and algorithms is provided.
[2] In this research, a CNN and LSTM-based end-to-end autonomous driving system is suggested. The steering wheel angle values of the vehicle could be deduced by taking into account the variations over time due to the use of CNN and LSTM that enabled the extraction of both the temporal and the spatial characteristics.
[3] Through this paper the ability of CNNs to learn the whole task of lane and road following without manual deconstruction into road or lane marking detection, semantic abstraction, path planning, and control has been empirically proven.
[4] This study uses a single camera to detect the car at first, and a matching algorithm to find a comparable vehicle on the other camera. Once the same vehicle has been identified by both cameras, the distance between them, the location of the vehicle in both cameras, and specific geometric angles are utilized as the basis for measuring the distance.
[5] In this paper, three different types of feature extraction networks are used to examine the performance of a system based on faster RCNN for detecting and identifying traffic signals in crossings.

III. METHODOLOGY

- **Extensive literature survey**

Looking up numerous research articles, comprehending them, going over them, and looking at the problem statement

- **Problem Identification**

After carefully examining all of the published papers, we compiled a list of the issues that appear in all of the study publications.

- **Data Selection for lane detection**

By running the model on a specified track number of kinds in the Udacity Simulator, we gather the data, which is then stored as photos and a csv log file.

- **Code for lane detection**

Once the data is saved coding the dataset will begin using Visual Studio Code

- **Understanding CNN Architecture**

We will comprehend and design a CNN architecture for our own model with the aid of the nVidia model.

- **Experimentation with CNN Architecture**

Once a model is created it is trained a number of times to gain accuracy and the outcome is analyzed..

IV. SIMULATION USING DEEP LEARNING AND UDACITY SIMULATOR

4.1 Importing Data

We can gather data in the Udacity simulator by repeatedly running the model around the track.
The information gathered consists of steering angle, throttle, and picture data from the left, right, and center cameras.
Images and log files are two ways that data is saved.
After recording the data, this log file and images are imported in VS code editor.

4.2 Visualization and Balancing of Data

Data is balanced by removing the images from the left and right cameras and only using the data from the center camera.

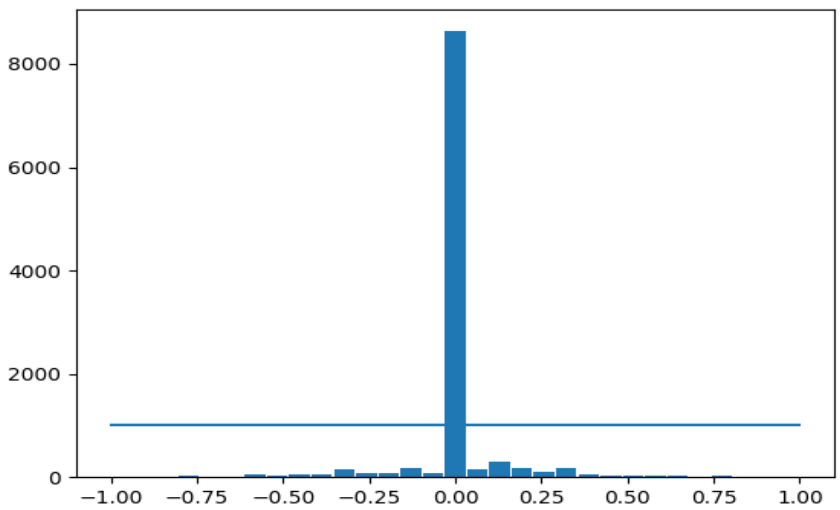

Fig 1. Graphical representation of dataset

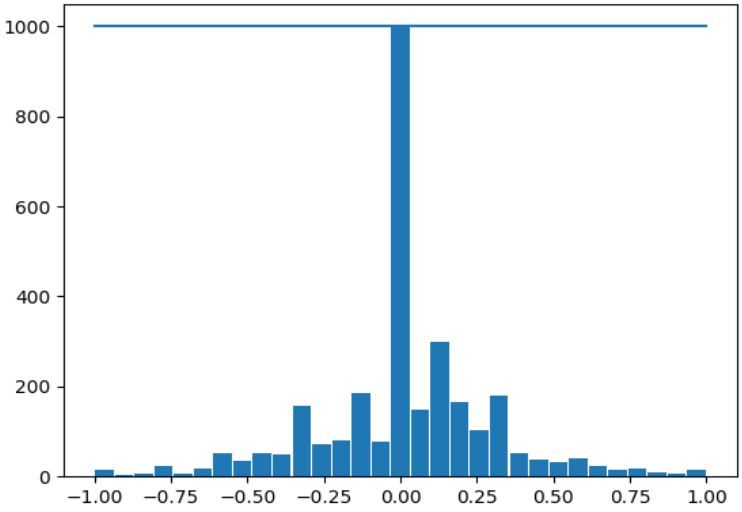

Fig 2. Refined Dataset

4.3 Data Augmentation

Once the data has been refined, the refined data set is supplemented with various elements such as pan, zoom, brightness, and others so that the model may demonstrate greater accuracy during training.

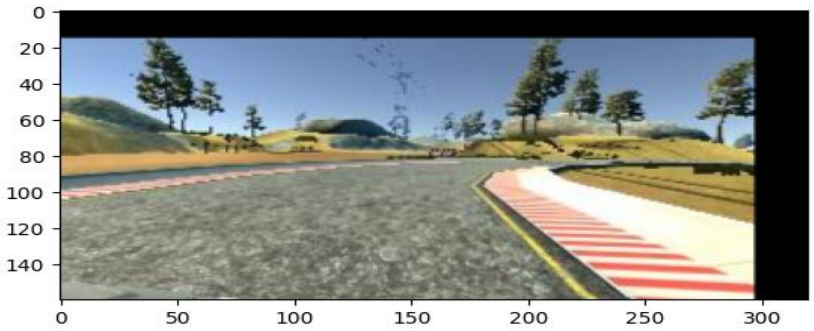

Fig 3. Pan Image

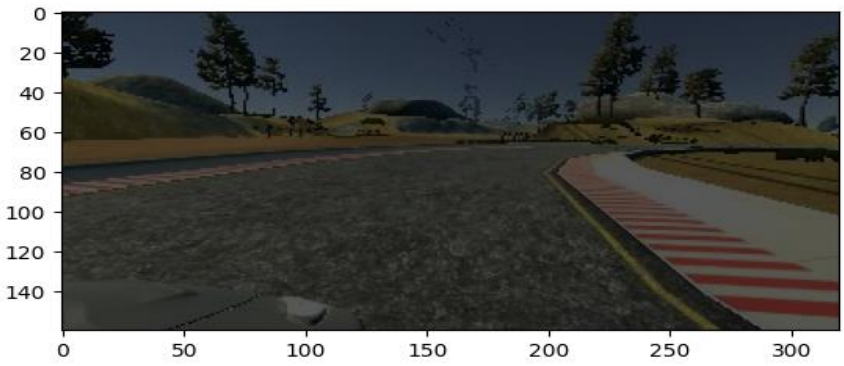

Fig 4. Brightness Image

Data is transformed into training data and validation data arrays in addition to data augmentation.When training a model, validation data is an array that has never been seen before by the model.

4.4 Preprocessing

Preprocessing is the process of transforming input into a form that a machine learning model, in this case, the CNN model, can use. It is an essential step for raising our Model's accuracy and effectiveness. Preprocessing of augmented data involves converting photos from RGB to YUV format. When preprocessing, photos are also scaled and blurred.

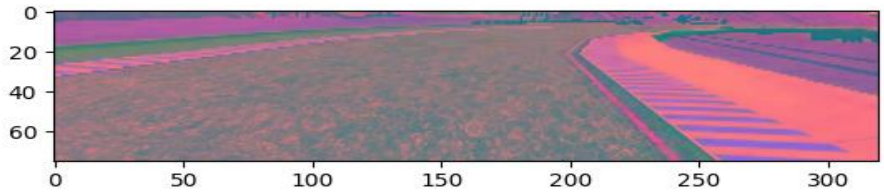

Fig 5. RGB to YUV

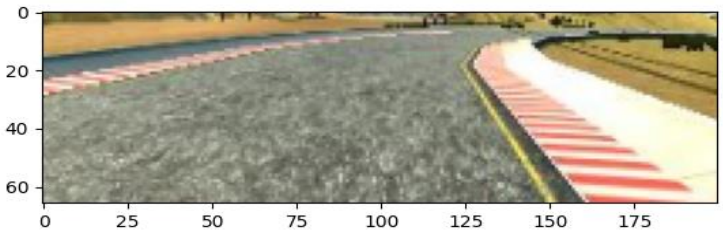

Fig 6. Resize Image

4.5 Model Creation

With reference to NVIDIA'S model for self driving simulation we built the model resembling nVidia's model.

The network consists of 9 layers , including a normalization layer, 5 convolutional layers and 3 fully connected layers. The input image is split into YUV planes and passed to the network.

The network has about 27 million connections and 250 thousand parameters.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 31, 98, 24)	1824
conv2d_1 (Conv2D)	(None, 14, 47, 36)	21636
conv2d_2 (Conv2D)	(None, 5, 22, 48)	43248
conv2d_3 (Conv2D)	(None, 3, 20, 64)	27712
conv2d_4 (Conv2D)	(None, 1, 18, 64)	36928
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 100)	115300
dense_1 (Dense)	(None, 50)	5050
dense_2 (Dense)	(None, 10)	510
dense_3 (Dense)	(None, 1)	11

Total params: 252,219
Trainable params: 252,219
Non-trainable params: 0

Fig 7. CNN Model based on Nvidia's architecture.

4.6 Model Training

After the model has been created and saved, it is trained using a real dataset that has been enhanced and preprocessed.
The training loss and validation loss are represented graphically after the training data and validation data have been trained.

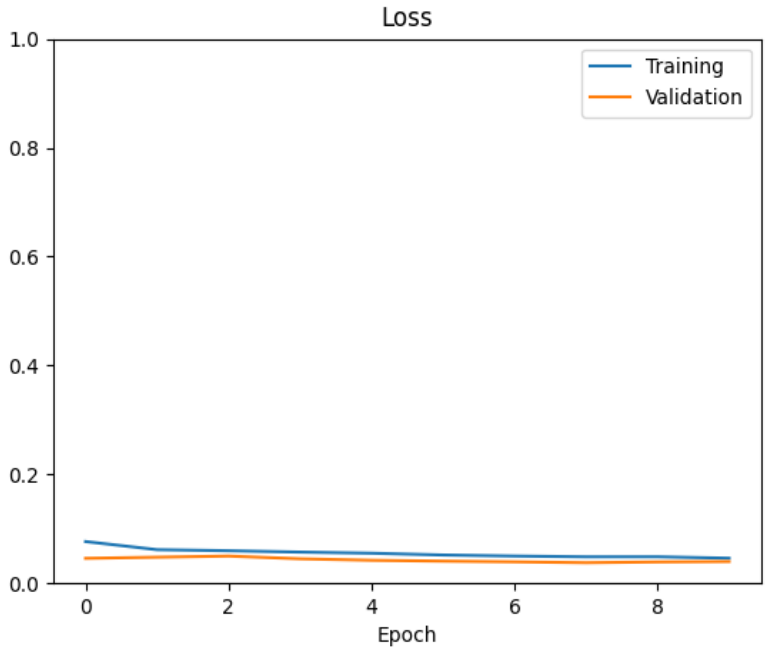

Fig 8. Graphical representation of data loss

V. Results and Conclusion

In order to simulate a real-world setting, Udacity published an open source simulator for self-driving cars. The task is to use a model built by deep neural networks to simulate human driving behavior on the simulator. Two tracks and two modes—training mode and autonomous mode—are included in the simulator. The user creates the dataset while operating the simulator while operating the vehicle in training mode. The "good" driving data is another name for this type of data. The deep learning model is then tested on the track to see how it does after being trained using the user data. Generalizing the performance across other tracks is another difficult task. That means, Using the dataset generated on one of the simulator's tracks as training data, the model is then tested on the other track.
The model was successfully able to detect lanes and maintain its position in the middle of the track when it was run on a simulator.

VI. Future Scope

Future design revisions of the project are possible, and we can make this system better by giving the model more functionality. such lane detection, vehicle recognition and crash avoidance, and increased computer vision use.

VII. Acknowledgment

We are pleased to express our deep sense of gratitude to our project guide Prof. Sudheer Hirolikar, who has opened floodgates of knowledge for us and his extended continuous cooperation, encouragement and his great insistence on research work to undertake.
We would also like to thank honorable Prof. Siddharaj Allurkar H.O.D. Automobile department for his constructive suggestions and giving us the opportunity to work on this project and present it.

References

[1]Sensors based Lane Keeping and Cruise Control of Self Driving Vehicles
Brook W. Abegaz Department of Engineering Science Loyola University Chicago Chicago, IL, 60626, USA babegaz@luc.edu
Naxi Shah Department of Engineering Science Loyola University Chicago Chicago, IL, 60626, USA nshah19@luc.edu

[2] Autonomous Driving Control Using End-to-End Deep Learning
Myoung-jae Lee Department of Computer Science and Engineering Konkuk University Seoul, South Korea dualespresso@naver.com Young-guk Ha Department of Computer Science and Engineering Konkuk University Seoul, South Korea ygha@konkuk.ac.kr

[3]End to End Learning for Self-Driving Cars
Mariusz Bojarski NVIDIA Corporation Holmdel, NJ 07735 Davide Del Testa NVIDIA Corporation Holmdel, NJ 07735 Daniel Dworakowski NVIDIA Corporation Holmdel, NJ 07735 Bernhard Firner NVIDIA Corporation Holmdel, NJ 07735 Beat Flepp NVIDIA Corporation Holmdel, NJ 07735 Prasoon Goyal NVIDIA Corporation Holmdel, NJ 07735 Lawrence D. Jackel NVIDIA Corporation Holmdel, NJ 07735 Mathew Monfort NVIDIA Corporation Holmdel, NJ 07735 Urs Muller NVIDIA Corporation Holmdel, NJ 07735 Jiakai Zhang NVIDIA Corporation Holmdel, NJ 07735 Xin Zhang NVIDIA Corporation

Holmdel, NJ 07735 Jake Zhao NVIDIA Corporation Holmdel, NJ 07735 Karol Zieba NVIDIA Corporation Holmdel, NJ 07735

[4] Vehicle to vehicle distance measurement for self-driving systems

Abdelmoghit ZAARANE, Ibtissam SLIMANI, Abdellatif HAMDOUN, Issam ATOUF LTI Laboratory, Physics Department, Faculty of Sciences Ben M'sik, University Hassan II, Casablanca, Morocco E-mail: z.abdelmoghit@gmail.com

[5] Traffic Light Detection for Self-Driving Vehicles Based on Deep Learning

WeiGuo Pan Beijing Key Laboratory of Information Service Engineering College of Robotics, Beijing Union University Beijing China ldtweiguo@email.com YingHao Chen College of Applied Science and Technology Beijing Union University Beijing China chenyinghao310@163.com Bo Liu College of Applied Science and Technology Beijing Union University Beijing China leilei12.1@foxmail.com