Deep Learning Based Lateral Control System

Tarun Tiwari
Department of Computer Science and Engineering
MVJ College of Engineering
Bengaluru, India
tarun12191998@gmail.com

Nishanth Shastry
Department of Electronics and communication
MVJ College of Engineering
Bengaluru, India
n18shastry@gmail.com

Aparajita Nandi
Department of Computer Science and Engineering
MVJ College of Engineering
Bengaluru, India
aparajita.nandi7178@gmail.com

2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC) | 978-1-7281-8880-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/iSSSC50941.2020.9358854

***Abstract*—Autonomous vehicles have seen advancement in technology over a while. Even then, it still lacks a robust system for the lateral control system of vehicles. Autonomous vehicles are considered the future due to the increasing population and limited infrastructure. But modern innovations in the industry along with research in the deep learning domain have shown promising results. We tried to demonstrate a lateral control system trained on a conventional neural network algorithm. An end-to-end deep neural network is discussed in this paper. The trained model of the proposed network is the re-implementation of the existing model AlexNet and PilotNet. The complexity and size of the novel network are much less than existing models. The model has been tested on a virtual environment trained on a small dataset collected from a simulated environment. Despite the small dataset, model attained an accuracy of 97.42% on known tracks and an accuracy of 86.09% on unknown tracks. The result indicates behavioral cloning and reinforcement learning can be used for driving vehicles in an unknown environment.**

***Keywords*— Machine Learning, Autonomous Vehicles, CNN, Reinforcement Learning, Data Analysis, Simulation**

I. INTRODUCTION

As of 2020, we have 1.2 billion vehicles running on the road and have an annual death count of 1.35 million people due to road accidents. According to the Navigant Research report, we would have 2 billion vehicles on road by 2035 which can lead to an annual death count to reach 1.8 million by 2035. Driving in such conditions would be beyond human capabilities, thus we need a self-driving car system that can drive without or less intervention from a human.

Autonomous vehicles are seen as the future of automobiles. Efforts in autonomous vehicle research [1][2] can be traced from the early 21st century. More recent studies [3]– [5] have targeted various challenges on controlling, vision, and decision-making capabilities. In recent times, development in deep learning methods using Convolutional Neural Networks (CNNs), shows a positive side for feature extraction from surrounding and especially decision-making capability of vehicles. General approaches that have been proposed for autonomous driving based on how the environment is analyzed by vehicle. First in which the environment is unknown to the vehicle and various algorithms are deployed to extract important features in the environment such as lanes, cars, signs, and pedestrians. These methods usually assume that there is a central system that collects this data and makes real-time decisions [3]– [6]. The other approach is in which a neural network model is trained to make driving decisions from monitoring the driving behavior of a human driver in reaction to different driving scenarios [1], [3], [7].

We propose a method to combine both these techniques which takes reinforcement learning into consideration for analyzing extracted features of the environment as well and help the central system of the vehicle in increasing the efficiency of its decision making. Burleigh. N [8] proposed a system similar to Audi Autonomous Driving Cup and Carolo Cup using deep learning for independently handling lane keeping and traffic sign recognition. Zanchin [9] through their research discusses the improvement of comfort, safety, and performance of the autonomous vehicle for potentially reducing traffic and congested traffics. Through their paper, they also discuss sensor fusion for an autonomous vehicle. Rajat Kumar Sinha [10] discussed different deep learning algorithms that are used in solving conventional artificial intelligence problems. It also discusses various approaches followed by image classification and object detection, image extraction, and segmentation in presence of noise and disturbance.

Locktev [11] proposed an algorithm to estimate the distance between objects based on geometric and kinematic parameters. It also discusses the use of methods and procedures of statistical analysis and probabilistic approach. Xia. W and Li. H [12] proposed a system for autonomous vehicles that works on principles of reinforcement learning, where the system learns from the experience of a professional driver and a Q-learning algorithm with filtered experienced and trained and implemented in the central system. Mnih and K. Kavukcuoglu [13] through their research explain the usage of reinforcement techniques in real-world complex situations and how can machine learn and implement how humans solve such complex situations. All models and research works were considered and analyzed while building our system from scratch.

The task is divided into planning paths and control logic. Data collected from the simulator is used to train re-implementations of existing models. Image Processing techniques are applied to the images collected as data for the model to increase the accuracy of the model by decreasing the chances of false detection.

Through our paper, we discussed a neural network based on the re-implementation of AlexNet and PilotNet after modification. AlexNet is traditionally used for object detection, but modification discussed helps model in an autonomous vehicle. The trained models are tested on a simulator based on the Unity environment.

The paper discusses is organized as follows: proposed methodology of the algorithm in section 2 followed by data acquisition discussion in section 3. Architecture is discussed in section 4 and sections 6, 7, and 8 discuss results, conclusion, and future scope.

II. Proposed Methodology

In this section, we present a model based on reinforcement learning that predicts the path mapping on road by deciding the vehicle's steering angle. Different vehicles on road are also taken into consideration. A conventional neural network is trained and used for extracting angle from captured images.

The methodology is divided into two parts; one focuses on collecting images and gathering data from it. The second focuses on the reinforcement learning technique to train a model and handles the vehicle controls like steering, throttle, and brakes thus enabling an autonomous system. The image is collected when the vehicle is manually driven and different sensors capture images and extract angles, which enables the system to clone its behavior the same way a human would in certain scenarios. A deep neural network is trained on the data collected from manual driving to create a lateral control system for the autonomous vehicle.

III. DATA ACQUISITION

This is accomplished is three stages.

A. Data Collection:

The data collection procedure begins with a manually driven vehicle equipped with sensors in a known environment on a testing track. At end of manual driving, images are collected from cameras mounted in front of the vehicle as it traveled through different terrains and situations. Data collection system software collects those images and extracts information from it. The images that are collected are later stored along with data collected about steering angle, throttle, and brakes in a table.

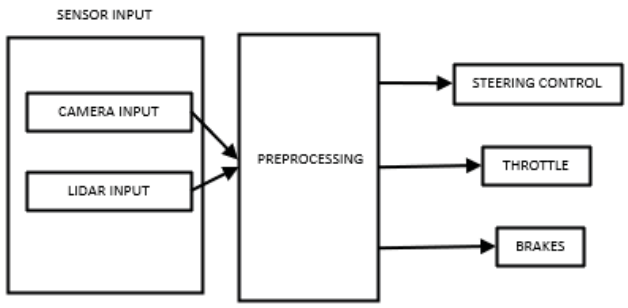

Fig1. Data Acquisition work flow

During data collection, the aim is to drive the vehicle in the center of the road as much as possible to teach the system how to behave or curves

B. Data Augmentation

Several laps are required to collect a large dataset. To maintain accuracy, a small dataset is collected and data augmentation techniques are applied. The size of the dataset is doubled and images were flipped in vertical access. The steering angle is multiplied by -1. By flipping images, we get added value to the final value. Also, techniques like image panning and zooming are applied to provide variety in the dataset. Data augmentation in this manner provides a balanced dataset for the model.

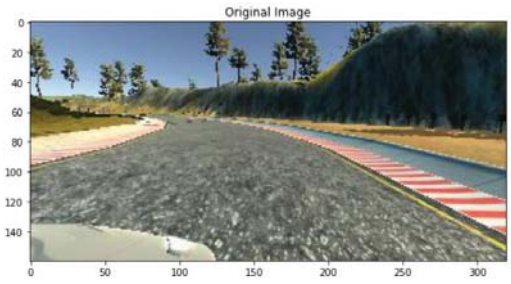

Fig2. Original image captured while collecting data

C. Data Preprocessing

To train the neural network, the images collected from sensors undergo few processing steps to ensure accuracy is maintained. Few steps are:

- Image is cropped to remove car fronts covered in the image and converted to YUV because YUV takes into consideration human perception.
- Image is grouped along with data extracted from it and stored in batches so that it can be used as data for the neural network.
- Morphological operations are applied to images to teach recovery of data from poorly orientated/positioned images.

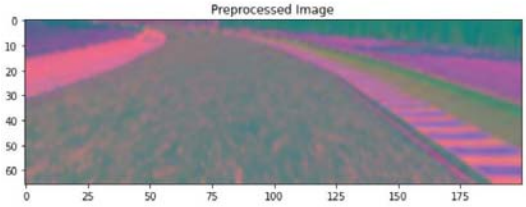

Fig3. Pre-processed image

The images from the data set are also normalized. To normalize, every pixel of the image is divided by 255. The result obtained is between 0 and 1. Later 0.5 is subtracted from the result to centralize the data.

$$x_{norm} = \frac{x}{255} - 0.5 \quad \ldots\ldots\ldots\ldots\ldots (1)$$

TABLE.I. Sample of Acquired Data

Image	Steering Angle	Throttle	Brake
2020-06-13-19-08-21-44-248	0.1785	0.0841	0.0611
2020-06-13-19-08-21-44-287	0.1656	0.0745	0.0632
2020-06-13-19-08-21-44-315	0.1721	0.0765	0.0689
2020-06-13-19-08-21-44-339	0.1882	0.0711	0.0609

The steering angle is the behavior that the model needs to learn and predict based on data that will be normalized between values 1 and -1. By plotting the data on the graph, we notice segregation of 0 angle data, this can lead to biased data set resulting in less accuracy.

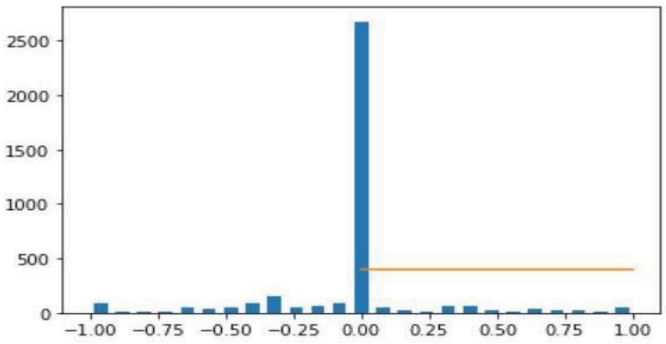

Fig.4 Biased dataset with excess 0 steering angle

To tackle this, excess of 0-angle data is removed and remaining values are flipped as well to make sure that the model is not biased towards any specific condition and is overall unbiased.

About 55% to 60% of the 0-angle data is removed from our collected dataset. As shown in the figure we have taken into consideration, randomly selected 400 0-angle data rows out of the total of more than 2500 data rows. The following histogram plot shows the final dataset distribution.

```
total data 3754
removed 2278
remaining 1476
[<matplotlib.lines.Line2D at 0x7f5bbbcdb438>,
 <matplotlib.lines.Line2D at 0x7f5bbbd7df60>]
```

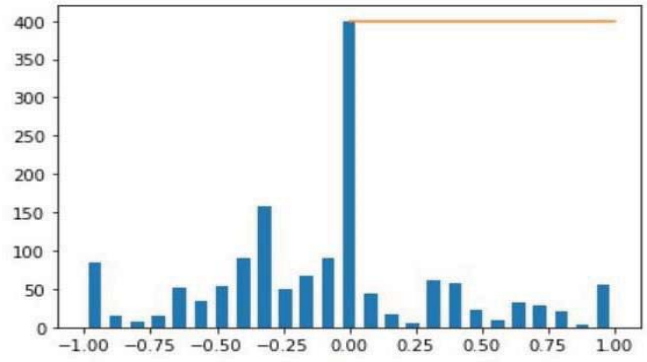

Fig.5. Unbiased dataset after removing steering angle with 0

Further, the data will be preprocessed by data augmentation and cleaning. This is done with pre-defined libraries in languages like the conversion from RGB to YUV of collected images.

IV. ARCHITECTURE

The two-dimensional convolution operation is applied on raw data of input images, where 2x2 size of the kernel is applied to extract patches of the image where weights w, is shared on convolution operation output S (I, j) to detect particular representation on the image.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)k(m,n) \quad \text{…….. (2)}$$

Relu function is applied later on the output to decrease gradient vanishing. The linear unit generated output 0 if the input is 0 else 1.

$$\mathrm{Re}\,LU(x) = \max(0,x) \text{ …… (3)}$$

Pre-existing neural network models AlexNet and PilotNet are modified and implemented in the end-to-end model for autonomous vehicle's lateral control system. This would reduce the attributes in the model thus reducing processing speed.

AlexNet is proposed to be re-implemented and be adapted. The traditional architecture of AlexNet has two pipelines as it is trained on 2 GPUs. New AlexNet would be trained on one GPU, so the simplified architecture of AlexNet contains one pipeline by combining the numbers of filters available in the proposed original solution. To deal with data accuracy, two of the three pooling layers in the original model are removed.

Following re-implementation helps in reducing trainable parameters. The size of the input image is 320x160 pixels (before cropping) as compared to the traditional input size of 224x224 pixels. The difference in input images, between the traditional and proposed models, helps in reducing trainable parameters. In the proposed model, only one output node is required i.e., steering angle prediction for a lateral control system.

Hyper Parameter Tuning: Architectures are implemented in Python-based Keras which is built based on the Tensorflow library. After images are collected from the camera, the correction parameter is applied to steering angle measurements. After fine-tuning, a correction factor of 0.22 is applied for images from side cameras. For normalization, the Lambda layer is applied. Both models are trained separately on same data.

The deviation between ground and steering prediction is reduced by mean squared error. The benefit of applying mean squared error (MSE) is that it provides an easy gradient computation. During network training, Adam optimizer is applied to both models. It is an optimization algorithm for deep learning models. It can combination Descent of Stochastic Gradient and RMSprop. Like RMSprop, it applies a squared gradient of scale learning rate and like Stochastic Gradient, it uses momentum by applying an average of gradients.

The usage of a small dataset increases the chances of overfitting. Therefore, early stopping regulation methods are applied to decrease the chances of over-fitting. Based on different combinations, AlexNet gave good results with 30 epochs and PilotNet gave good results on 4 epochs. Final implementations of the model are finalized by all hyperparameter tuning techniques. Models after being trained are saved used in the simulator for the self-driving vehicle.

V. TRAINING

For checking the compatibility of the model in a real-time scenario of the environment, the trained model is simulated

on a virtual simulator. A self-driving car simulator allows the developer to create a lateral control system and test in real time scenario. The simulator provides an inbuilt model to extract steering angle, brake, and throttle while manually driving car in the simulator. For testing, a server patch is required to provide a surface for the manual controller of the vehicle in the simulator. It helps in developing a connection between simulation client and server. A standard install of the simulator can only help in data collection of vehicles

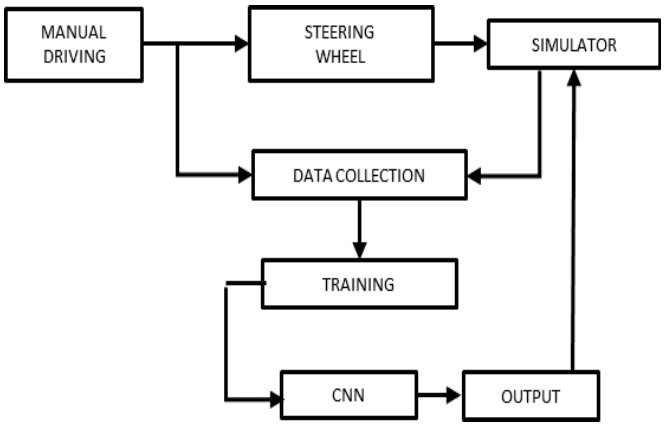

Fig6. Training cycle of the model

A. *Software Requirement*

1. *Unity Simulator:* Cross-platform game engine used for developing graphics applications for PC and consoles. Main purpose to use Unity is that base used is built on Unity so simulator can provide better compatibility.

2. *TensorFlow:* Open source deep learning framework based on mathematical operations on data arrays.

3. *Keras:* Neural Network API that runs on TensorFlow Backend.

B. *Model Training*

A batch of a random set of data is imported from the dataset, the sets are random batches imported while training. After every epoch, the network is evaluated. Colab then starts the training process again. The batch generator then provides input batches which are later used to start the training, and model having random weights, the output is then compared with labels. Using the backpropagation process, weights are adjusted after the mean square error is calculated for the output. It is achieved by applying Adam optimizer with a learning rate of 0.0001.

VI. RESULTS

The trained model was close enough in learning from the behavior of drivers with an accuracy of 97.42% in a known environment and accuracy of 86.09% in an unknown environment, with data collected from manually driving a car in the simulator for approx. 1 hour. The model was able to predict steering angle and throttle depending on different terrains. Image preprocessing was useful to train models about extracting data from poorly positioned data. The experiment cannot be seen as an industry-ready feature in the vehicle but it can be seen as steps towards building a lateral control system.

The experiment also provides an alternative for hardware. Deep learning algorithms can help in reducing dependency on hardware, which can help in reducing the cost of production. A major development in GPU has acted as a catalyst by reducing the period of training data. Overall, the main aim to implement a neural network was achieved. To achieve this several existing algorithms were tested. Though it helped in reducing implementation it increases dependency on in-built libraries and frameworks.

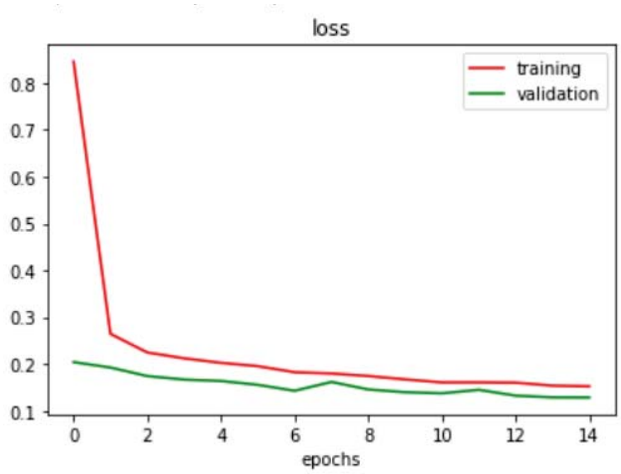

Fig.7.Loss Validation Function

VII. CONCLUSION

The experiment supports the usage of deep learning-based CNN in the development of the lateral control system in an autonomous vehicle. Trained model was trained on a simulator. Data was collected by manually driving the vehicle in a virtual environment. Collected data was used in the training model. 70-30 was the ratio of training and testing data. Experiment was successful as an accuracy of approx. 97% was achieved by testing the model. This demonstrates that reinforcement learning was successful in teaching the task of pathfinding by analyzing the steering angle and throttle. The experiment demonstrated how a combination of image processing and CNN can be used in semantic abstraction and path planning and reduce unsafe behavior of driving in public. This can help in solving the problem discussed above of increasing accidents and the need for a fully autonomous vehicle on roads.

The results ensure that a combination of image processing, deep learning, and computer vision is safe and accurate for fully autonomous vehicles and future development can benefit from combining the strengths of these technologies and increase the potential of vehicles.

VIII. FUTURE SCOPE

Further changes can be done in the future to make the system more accurate and increase its capability in an unknown environment. The performance of the system can be improved by:

• To modify the capabilities of the model, we can train the model on more training data on different tracks and different terrains. This can enhance their pathfinding accuracy.

• In the future models can be trained on Deep Q Learning algorithms to train vehicles about obstacle dodging capabilities. This can help in teaching cars to drive in the congested area.

• Trained model can be implemented in a physical car. This can create a huge impact by finding hardware compatibility and understanding realistic problems and can work to overcome them.

• Since the project experiment clearly shows software dependency, a system needs to be created to save the lateral control system from cyber-attacks as software is more prone to cyber-attacks.

REFERENCES

[1] D. a Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Adv. Neural Inf. Process. Syst. 1*, pp. 305–313, 1989.

[2] S. Oh, E. Kim, and J. Lee, "Autonomous Intelligent Cruise Control using Scanning Laser Sensor," pp. 1–7.

[3] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving," *2015 IEEE Int. Conf. Comput. Vis.*, pp. 2722–2730, 2015.

[4] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng, "An Empirical Evaluation of Deep Learning on Highway Driving," *arXiv*, pp. 1–7, 2015.

[5] M. Aly, "Real Time Lane Detection in Urban Streets," *Intell. Veh. Symp.*, pp. 1–3, 2008.

[6] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle Detection and Tracking in Car Video Based on Motion Model," *Intell. Transp. Syst. IEEE Trans.*, vol. 12, no. 99, pp. 1–13, 2011.

[7] M. Felsberg, A. Robinson, and K. Ofj, "Visual Autonomous Road Following by Symbiotic Online Learning," no. Iv, 2016.

[8] Burleigh, N., King, J., & Braunl, T. (2019). "Deep Learning for Autonomous Driving. 2019 Digital Image Computing: Techniques and Applications" (DICTA).

[9] B. C. Zanchin, R. Adamshuk, M. M. Santos and K. S. Collazos, "On the instrumentation and classification of autonomous cars," in IEEE Internation Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017.

[10] R. Sinha, R. Pandey and R. Pattnaik, "Deep Learning for Computer Vision Tasks: A review," in International Conference on Intelligent Computing and Control, Odisha, 2017.

[11] N. Burleigh, "Autonomous Driving on a Model Vehicle: Lane Detection and Control," University of Western Australia, Perth, 2019. NVIDIA Corporation, "Jetson Nano: Deep Learning Inference Benchmarks," 2019. [Online]

[12] Loktev, & Loktev, A. A. (2016). Estimation of measurement of distance to the object by analyzing the blur of its image series. 2016 International Siberian Conference on Control and Communications

[13] Xia, W., Li, H., & Li, B. (2016). A Control Strategy of Autonomous Vehicles Based on Deep Reinforcement Learning. 2016 9th International Symposium on Computational Intelligence and Design (ISCID).

[14] P. WawrzyĔSki and A. K. Tanwani, "Autonomous reinforcement learning with experience replay," Neural Networks, vol. 41, pp. 156- 167, 2013.