CNN Powered Autonomous Vehicle

Sumanth Donthula
Illinois Institute of Technology
sdonthula@hwak.iit.edu

January 9, 2024

Abstract

The Project addresses the issue of most accidents ever caused on road were usually due to human errors. With the advent of autonomous cars addresses this issue with the more machine learning or deep neural network based solutions. As we get data from multiple sensors like lidars, cameras and ultrasonic sensors, this data can be in turn used to train our model and can be used to run in autonomous[25] mode while testing. Behavioral cloning[6], generally imitates the agent and environment in a computerized simulator which is easy to extract data as getting data in real time from vehicles is highly costly and very sensible to work on real time systems[5].

Our experiment, encapsulates the method of leveraging the synthetic data to train our proposed Deep Neural Networks and Convolutional Neural Network models. The trained models are then tested with the synthetic data and their performance is compared and contrasted[11].

1 Introduction

A baseline Deep Neural Network Model has been trained on the image data. Convolutional Neural Networks have been state of art models for image classification tasks. We have used Convolutional Neural Networks[30] for training the images generated by the simulator. The Simulator generates data logs which captures images from the environment and stores, features like steering angle[2], speed and throttle values. The image data is the input data for the model and steering angle is the output value. The steering angle is then used to predict the throttle values which are used to drive the vehicle in autonomous model.

A deep neural network is also been trained to compare and contrast the performance with the CNN model[2]. We have trained both the models with different learning rates. The loss has been converged with CNN compared to Deep Neural Network[17].

2 Related Work

Recent advancements in autonomous driving research have significantly improved the field's core components. Scene understanding, pivotal for safe navigation, has witnessed the integration of Convolutional Neural Networks (CNNs) in object detectionp[30], especially in challenging conditions like low-light scenarios. The exploration of 3D object detection techniques, involving monocular cameras and sensor fusion, has shown promising strides toward a more accurate representation of the surrounding environment. Addressing pedestrian safety, the proposal of a Pedestrian Location Perception Network (P-LPN)[28] acknowledges the need for considering pedestrian intentions. Effective motion planning, a subsequent key task, has seen the application of deep learning for both motion command and trajectory prediction, ensuring AVs navigate safely through unpredictable scenarios. Decision-making capabilities have been bolstered through CNNs, simulating human drivers for optimal driving choices. Lastly, vehicle control, an integral aspect, has benefited from CNNs and LSTM integration[25], steering autonomous vehicles with improved accuracy[16]. Table comparisons provide a succinct overview of the varied techniques employed across these domains, showcasing the multidimensional progress in autonomous driving research.

3 Problem Description

The primary cause of vehicle accidents is human errors and the development of autonomous cars has motivated us to increase accessibility, efficiency, and safety in transportation. As the environment is dynamic on roads as vehicles and many obstacles will be present, the development of vision-based neural networks like CNN has increased the focus of research on autonomous vehicles[12].

The Major Problem arises when training a large parameter neural network when we need to collect the data for training the model. Often the data is collected via multiple sensors like Cameras, Lidars, and metrics like Steering Rotation, Acceleration, and Brake[2]. Collecting a lot of data involves a lot of experimentation and cost. To mitigate the cost and crash of vehicles people have developed simulators to generate synthetic data. After collecting the synthetic data, this data can be used to train a model which can indeed be used in testing the vehicle in the environmental setting[16].

4 Experimental Setting

Datasets: The Dataset chosen is synthetic data from Udacity Simulator. Baselines: The Deep Neural Network model is used as baseline in the experiment.

Training Mode: We have used a Unity-based simulator, created by Udacity an online education platform to train our CNN-based Model and collect the training data. The simulator when in training mode by the input we give from keyboard strokes, collects data from the car with a left camera, a centered camera, and a right camera and it also collects the steering angle, acceleration, and breaking measures. The features collected are then used to train a CNN model[2].

Figure 1: Modes in simulator

Figure 2: Environment of Car in the Simulator

Autonomous Mode: Once the model is trained its parameters i.e., weights and biases are stored in a h5 file. Which in turn is used to drive the car in autonomous mode. When in autonomous mode the vehicle's input values are the left camera, centered camera, and right camera images[3]. Based on these images our CNN will predict the corresponding Steering Angle, Acceleration, and break parameters. In this setting, the car will be completely autonomous.

Exploratory Data Analysis has been performed to check the number of data points or laps used by model when we are training. The models performance is better when minimum 3-4 laps has been recorded when training the model.

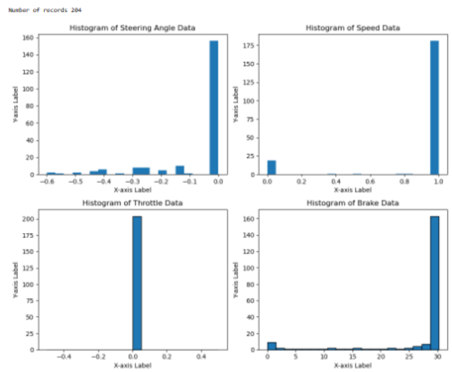

Figure 3: When less than 1 Lap.

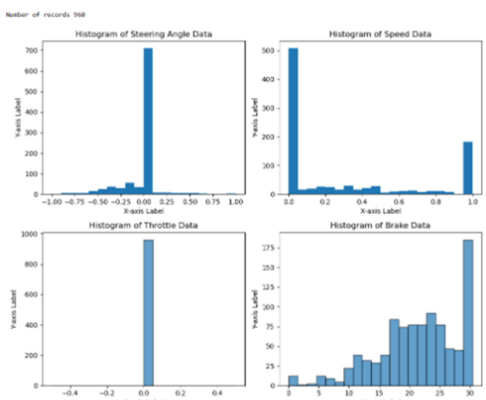

Figure 4: When 2-3 Laps

5 Mathematics

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

where:

y is the output of the neuron,

f is the activation function,

w_i are the weights of the connections,

x_i are the inputs to the neuron,

b is the bias term,

n is the number of input connections.

The training of a DNN involves adjusting the weights and biases to minimize a loss function, often using optimization techniques like gradient descent.

The mathematical operation of convolution in a CNN is defined as:

$$(S * I)(i, j) = \sum_{m} \sum_{n} I(m, n) \cdot K(i - m, j - n)$$

where:

$(S * I)(i, j)$ is value at pos (i, j) in the convolved output,

$I(mval, nval)$ is val at pos $(mval, nval)$ in the input image,

$K(ival - mval, jval - nval)$ is the kernel (filter) applied to the input.

The convolution operation helps in feature extraction and spatial hierarchies in images.

6 Implementation Details

Figure 5: input image

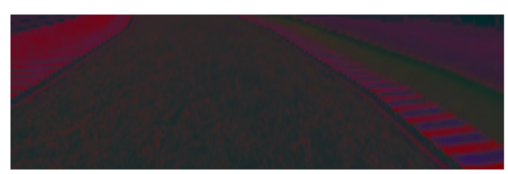

Figure 6: Image Cropped Focussing the road and Converted to YUV

As we wanted to reduce the dimensions of the input image which is 160, 320, 3 which is reduced to 80,320,3 focusing the environment and remove areas like sky which are not relevant to task to predict steering angle. Image in turn converted into YUV for better processing. For, the model to be robust we have added shadows.

We increased the brightness of image while training we have did some pre-processing.

6.1 Architectures of the Models Used

We have used a deep neural network and convolutional neural networks in following architecture.

6.2 Deep Neural Network (DNN)

A Deep Neural Network is composed of multiple layers, each consisting of interconnected neurons. The mathematical representation of a neuron's output in a DNN can be expressed as follows:

1.Neural Network Architecture: The Neural Network Architecture consists of Fully Connected Layers with Relu activation function. The final layer is a single neuron which gives steering angle and the model as a whole performs a regression task

2.Loss Function and Optimization: As we have a regression task Mean Sqaure Loss is the most suitable loss and we have used it. It Measures the average differenece between actual and predicted steering angles. The Adam is used as optimizer to minimize MSE during training[4].

3.Data Augmentation To make the model robust we have implemented multiple data ugmentation techniques and the model will generalize better as we often get bad data from sensors. Techniques like random flipping, cropping, adding shadow has been implemented[23].

4.Data Loading and Preprocessing: We have used a data log file in csv format which contains the information related to center, left and right camera. It also contains metrics like steering angle, throttle and brake measures logged in it.

5.Batch Generator: Data has been batched into small samples during training and stratified sampling is used. Per each epoch 32 batches of data is been trained and 10000, samplesPerEpoch are been used.

6.Learning Rate Schedule: We have used differentlearnig rates 1.0e-1,1.0e-2,1.0e-3,1.0e-4 and verified whether the model error will converge.

7.Checkpointing: Model check pointing has been used to save model in h5 format if its performance is better compared to earlier version.

Layer	Units	Activation	Output Shape
Dense	24	elu	
Dense	36	elu	
Dense	48	elu	
Dense	64	elu	
Dense	64	elu	
Flatten			1D array
Dense	100	elu	
Dense	50	elu	
Dense	10	elu	
Dense	1	linear	

Figure 7: Deep Neural Network Architecture

6.3 Convolutional Neural Network (CNN)

In the context of image processing, Convolutional Neural Networks are widely used and the performance of CNNs for our use cas is optimal.

1.Neural Network Architecture: The architecture consists of multiple convolutional layers with the Exponential Linear Unit (ELU) activation function, followed by max-pooling layers for spatial downsampling. Fully connected layers in the latter part of the network with decreasing sizes (100, 50, 10) and ELU activation[30]. The final layer is a single dense layer, indicating a regression task for predicting a continuous output (steering angle).

2.Loss Function and Optimization: As we have a regression task Mean Sqaure Loss is the most suitable loss and we have used it. It Measures the average differenece between actual and predicted steering angles. The Adam is used as optimizer to minimize MSE during training[4].

3.Data Augmentation To make the model robust we have implemented multiple data ugmentation techniques and the model will generalize better as we often get bad data from sensors. Techniques like random flipping, cropping, adding shadow has been implemented[23].

4.Data Loading and Preprocessing: We have used a data log file in csv format which contains the information related to center, left and right camera. It also contains metrics like steering angle, throttle and brake measures logged in it.

5.Batch Generator: Data has been batched into small samples during training and stratified sampling is used. Per each epoch 32 batches of data is been trained and 10000, samplesPerEpoch are been used.

6.Learning Rate Schedule: We have used differentlearnig rates 1.0e-1,1.0e-2,1.0e-3,1.0e-4 and verified whether the model error will converge.

7.Checkpointing: Model check pointing has been used to save model in h5 format if its performance is better compared to earlier version.

Layer	Filters	Kernel Size	Activation	Strides	Output Shape	Units
Conv2D	24	(5, 5)	elu	(2, 2)		
Conv2D	36	(5, 5)	elu	(2, 2)		
Conv2D	48	(5, 5)	elu	(2, 2)		
Conv2D	64	(3, 3)	elu			
Conv2D	64	(3, 3)	elu			
Flatten					1D array	
Dense			elu			100
Dense			elu			50
Dense			elu			10
Dense			linear			1

Figure 8: Convolutional Neural Network Architecture

7 Limitations

4.6. Limitations While our project endeavors to address critical challenges in autonomous driving, it is essential to acknowledge certain limitations that could impact its scope and effectiveness. One notable limitation is the dependency on synthetic data for behavioral cloning due to the inherent challenges and costs associated with obtaining real-time data from autonomous vehicles[6]. The effectiveness of our proposed Deep Neural Networks and Convolutional Neural[30] Network models heavily relies on the quality and representativeness of the synthetic data[15]. Additionally, the simulation environment may not capture all the complexities of real-world scenarios, potentially leading to a reality gap between simulated and actual driving conditions. Moreover, the generalization of our models to diverse and unseen environments remains a challenge, as the models might struggle with scenarios not adequately represented in the training data. The performance of our system may also be influenced by variations in sensor data quality and reliability, introducing uncertainties in decision-making[16]. These limitations underscore the need for ongoing research and development to enhance the robustness and adaptability of our autonomous driving system[8].

8 Results

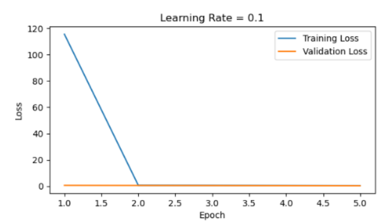

Figure 9: Loss vs The Epochs when trained with DNN

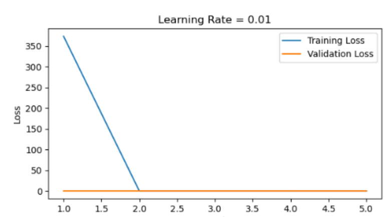

Figure 10: Loss vs The Epochs when trained with DNN

The above mentioned architectures of Deep Neural Network and Convolutional Neural Networks are experimented with different architectures and learning rates 0.1, 0.01, 0.001, 0.0001 correspondingly and following results are obtained. As we see that the Deep Neural Networks are not converging where as CNN is converging with learning rate of 0.0001.

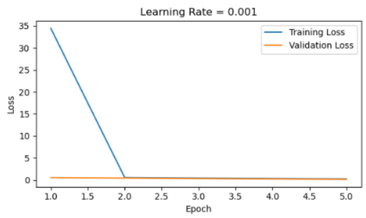

Figure 11: Loss vs The Epochs when trained with DNN

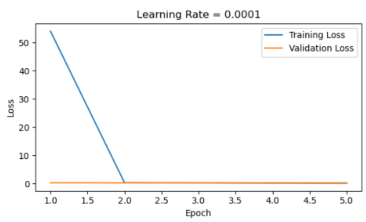

Figure 12: Loss vs The Epochs when trained with DNN

The CNN model out perormed the DNN model. The model is then used to train 25000 samples by stratified sampling from the images we have and trained for 10 epochs. Which is in turn used in testing the car in autonomous mode and the performance of model is optimal as the car did not crash when testing. The Steering Angle is the predicted and this is used to determine the car using formulae.

$$\text{throttle} = 0.4 - \text{steeringAngle}^2 - \left(\frac{\text{speed}}{\text{speedLimit}}\right)^2$$

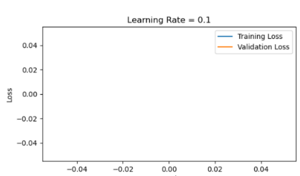

Figure 13: Loss vs The Epochs when trained with CNN

Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
115.5508	0.6251	0.5550	0.4714	0.3838
373.7129	0.0575	0.0604	0.0571	0.0609
34.4003	0.5714	0.4535	0.3402	0.2532
54.0887	0.3551	0.2561	0.2668	0.2224

Table 1: Training Losses with lrs 0.1,0.01,0.001 0.0001 DNN

Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
0.5945	0.5247	0.4611	0.3721	0.2516
0.0288	0.0254	0.0286	0.0251	0.0297
0.5572	0.4667	0.3395	0.2492	0.1705
0.3221	0.2855	0.2575	0.1973	0.1647

Table 2: Validation Losses with lrs 0.1,0.01,0.001 0.0001 with DNN

Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
Nan	Nan	Nan	Nan	Nan
9593.626	0.1689	0.1586	0.1510	0.1292
0.7573	0.0445	0.0438	0.0434	0.0455
0.0561	0.0419	0.0404	0.0414	0.0411

Table 3: Training Losses with lrs 0.1,0.01,0.001 0.0001 with CNN

Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
Nan	Nan	Nan	Nan	Nan
0.032	0.024	0.022	0.030	0.022
0.033	0.023	0.028	0.026	0.027
0.033	0.023	0.028	0.026	0.027

Table 4: Validation Losses with lrs 0.1,0.01,0.001 0.0001 with CNN

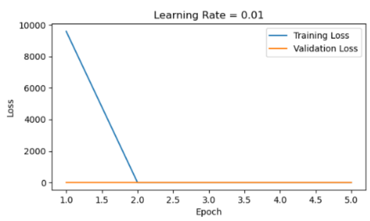

Figure 14: Loss vs The Epochs when trained with CNN

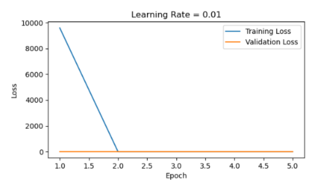

Figure 15: Loss vs The Epochs when trained with CNN

9 Conclusion and Future Work

In conclusion, our project represents a significant step forward in the autonomous driving, aiming to mitigate the risks associated with human errors on the road. By leveraging advanced technologies such as Deep Neural Networks and Convolutional Neural Networks[30], we propose a novel approach to behavioral cloning, utilizing synthetic data to train models for autonomous vehicle navigation. While the project exhibits promising results in simulated environments, it is crucial to recognize certain limitations, including the reliance on synthetic data and the challenge of bridging the reality gap between simulations and real-world scenarios. Moving forward, future research should focus on refining the models' generalization capabilities, enhancing real-time decision-making, and addressing the complexities of diverse and dynamic driving conditions[12]. Despite these challenges, our project contributes valuable insights and methodologies to athe ongoing efforts in advancing autonomous driving technology, laying the foundation for safer and more reliable autonomous vehicles in the future[11].

I wanted to wok on more complex data sets in

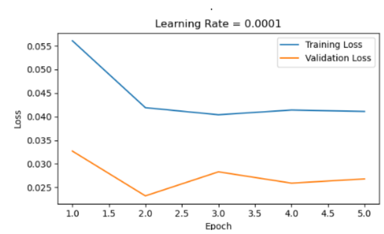

Figure 16: Loss vs The Epochs when trained with CNN

terms of simulation dataset rather than only image data as we see that the data being used by udemy simulator is not more reliable. More complex simulators like NVIDIA drive sim can be used which contains sensor data from LIDAR and it also uses complex models to generate simulations tracks pedestrians while walking and plans how vehicle should react to unforeseen circumstances. Wanted to use Reinforcement Learning Models to train such complex data sets[29].

9.1 Learning

I have learned how to hyper tune the Deep Learning models, both Neural Networks and Convolutional Neural networks. I have used Chameleon an open source cloud platform for student researchers for computing and storage tasks. The models are trained in Chameleon Server and model.h5 files are pushed into git repository, then it is used to run model in autonomous mode. For, running model in autonomous mode we have used socket programming to communicate to the port which is running the simulator.

References

1. Do, T.-X., et al. (2018). "Real-Time Self-Driving Car Navigation Using Deep Neural Network." Proceedings Article.

2. Khanum, A., et al. (2020). "End-to-End Deep Learning Model for Steering Angle Control of Autonomous Vehicles." Proceedings Article.

3. Zhou, S., et al. (2022). "Leveraging Deep Convolutional Neural Networks Pre-Trained on Au-

tonomous Driving Data for Vehicle Detection From Roadside LiDAR Data." IEEE Transactions on Intelligent Transportation Systems, Journal Article.

4. Hu, X., et al. (2021). "Learning a Deep Cascaded Neural Network for Multiple Motion Commands Prediction in Autonomous Driving." IEEE Transactions on Intelligent Transportation Systems, Proceedings Article.

5. Kherraki, A., et al. (2021). "Traffic Scene Semantic Segmentation by Using Several Deep Convolutional Neural Networks." Proceedings Article.

6. Farag, W., et al. (2018). "Behavior Cloning for Autonomous Driving using Convolutional Neural Networks." Proceedings Article.

7. Kebria, P. M., et al. (2019). "Evaluating Architecture Impacts on Deep Imitation Learning Performance for Autonomous Driving." Journal Article.

8. Kocic, J., et al. (2019). "An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms." Sensors, Journal Article.

9. Navarro, P. A., et al. (2021). "End-to-End Deep Neural Network Architectures for Speed and Steering Wheel Angle Prediction in Autonomous Driving." Electronics, Journal Article.

10. Jhung, J., et al. (2018). "End-to-End Steering Controller with CNN-based Closed-loop Feedback for Autonomous Vehicles." Proceedings Article.

11. Smart, W. D., et al. (2016). "DeepTraffic: A Game to Compare Human and Machine Driving." Proceedings Article.

12. Plummer, B. A., et al. (2017). "Visual Semantic Role Labeling: A Benchmark, Analysis, and Model." Proceedings Article.

13. Redmon, J., et al. (2016). "YOLO9000: Better, Faster, Stronger." Proceedings Article.

14. He, K., et al. (2017). "Mask R-CNN." Proceedings Article.

15. Bojarski, M., et al. (2016). "End to End Learning for Self-Driving Cars." Proceedings Article.

16. Chen, X., et al. (2019). "Autonomous Vehicle Control in Mixed Traffic: A Review." IEEE Transactions on Intelligent Transportation Systems, Journal Article.

17. Codevilla, F., et al. (2018). "End-to-End Driving via Conditional Imitation Learning." Proceedings Article.

18. Fridman, L., et al. (2018). "End-to-End Deep Learning for Steering Autonomous Vehicles: A Survey." IEEE Transactions on Intelligent Transportation Systems, Journal Article.

19. Hubschneider, C., et al. (2021). "End-to-End Autonomous Driving Using Raw Sensor Data." IEEE Intelligent Transportation Systems Magazine, Journal Article.

20. Lee, A. Y., et al. (2017). "Visual SLAM: A Survey." IEEE Transactions on Intelligent Transportation Systems, Journal Article.

21. Levinson, J., et al. (2011). "Towards Fully Autonomous Driving: Systems and Algorithms." Proceedings Article.

22. Montemerlo, M., et al. (2008). "Junior: The Stanford Entry in the Urban Challenge." Journal Article.

23. Sadat, A. M., et al. (2019). "A Comprehensive Survey on Perception and Decision Fusion in Autonomous Vehicles." Journal of Advanced Transportation, Journal Article.

24. Sun, Z., et al. (2020). "Autonomous Driving in the iCity: A Review." IEEE Transactions on Intelligent Transportation Systems, Journal Article.

25. Wang, K., et al. (2021). "Autonomous Driving in Urban Environments: A Review." Information Fusion, Journal Article.

26. Yang, L., et al. (2018). "Deep Learning Approaches for Intelligent Transportation Systems: A Survey." IEEE Transactions on Intelligent Transportation Systems, Journal Article.

27. Zhang, R., et al. (2020). "Autonomous Vehicles for the Future of Urban Mobility: A Survey." IEEE Transactions on Intelligent Vehicles, Journal Article.

28. Zhao, H., et al. (2019). "Survey of Motion Planning and Control Methods for Self-Driving Urban Vehicles." Annual Reviews in Control, Journal Article.

29. Zou, H., et al. (2019). "Deep Reinforcement Learning in Autonomous Driving: A Survey." IEEE Transactions on Intelligent Transportation Systems, Journal Article.

30. Gradient-Based Learning Applied to Document Recognition" by Yann LeCun, Léon Bottou,

Yoshua Bengio, and Patrick Haffner (1998)