Mitigating the Zero Biased Steering Angles in Self-driving Simulator Datasets

Muhammad Ammar Khan[a], Khawaja Ghulam Alamdar[b], Aiman Junaid[c] and Muhammad Farhan[d]

DSSE-Dhanani School of Science & Engineering, Habib University, Pakistan

Keywords: Autonomous Cars, Self Driving, Udacity Simulator, Zero-bias Steering Angle.

Abstract: Autonomous or self-driving systems require rigorous training before making it to the roads. Deep learning is at the forefront of the training, testing, and validation of such systems. Self-driving simulators play a vital role in this process not only due to the data-intensiveness of the deep learning algorithm but also due to several parameters involved in the system. The data generated from self-driving car simulators have an inherent problem of large zero-bias due to the discrete nature of computation arising from computer input devices. In this paper, we analyze this problem and propose filtering to make the steering angles in the dataset smoother and to remove random fluctuations that make our model learn better. After such processing, the test run on simulators showed promising results using a significantly small dataset and a relatively shallow network.

1 INTRODUCTION

With the growing field of autonomous learning, self-driving cars are emerging as a center of focus for automobile industries. The current trend of the automotive industry combined with research by the major tech companies e.g., Apple, Ford, NVIDIA, proves that self-driving cars are the future (Greenblatt, 2016). Google is one of the leaders in self-driving cars, based on its strong foundation in artificial intelligence. It already tested two self-driving cars on the road in June 2015. The current development is that Google vehicles have accumulated more than 3.2 million km of tests, becoming the closest to the actual use. Tesla is another company that has made significant progress in this field. It was the first company to devote self-driving technology to production. Followed by the Tesla models series, its "auto-pilot" technology has made breakthroughs in recent years. There are several other Car and Internet companies like Zenuity, established by the collaboration of Sweden, Volvo, and Autoliv, committed to the security of self-driving cars and have devoted their research towards this growing field (Coelingh et al., 2018).

One of the major aspect in success of self-driving cars hinges on learning human sub-cognitive skills. Behavioral cloning is a method through which it is done and in recent years, several deep learning-based behavioral cloning approaches have been developed in the context of self-driving cars. Training such deep networks requires a large dataset which is difficult to gather. Moreover, extensive testing in a real setting is very hard since a lot of conditions are needed to be validated by the model i.e., different weathers, climates, and topographies, which makes it hard for the researchers to collect dataset for such scenarios and also test it physically (Hars, 2010). The research groups trying to build their autonomous car prototypes run into the same problems, however, this could pave the way for more productive research including new sensor setups (i.e. LiDARS) and command and control systems.

Another challenge in the development of autonomous vehicles is that every day, test fleets around the world create petabytes of data. Multiple teams working simultaneously must process, sample, and utilize this data. Additional design iterations may be generated by every update during the development cycle. Therefore, simulation remains the best method to tackle these challenges. Moreover, with the simulator, the benefits and drawbacks of various algorithms could be thoroughly examined conveniently and without imposing risk to any life in case of model failure

[a] https://orcid.org/0000-0002-7591-7548
[b] https://orcid.org/0000-0002-2137-7208
[c] https://orcid.org/0000-0002-4394-6133
[d] https://orcid.org/0000-0002-8244-8313

Khan, M., Alamdar, K., Junaid, A. and Farhan, M.
Mitigating the Zero Biased Steering Angles in Self-driving Simulator Datasets.
DOI: 10.5220/0010839900003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP*, pages 470-475
ISBN: 978-989-758-555-5; ISSN: 2184-4321

(Sharma et al., 2020).

To aid the current research on self-driving cars, it is necessary to make simulation test-beds for the trained models which proves to be faster, less expensive, and much more insightful than a regular and physical prototype. Developing a real-time simulation environment for automatic cars however poses interesting computing challenges that will enhance research on efficient algorithm building as well.

In this paper, we analyze the inherent problem with the simulator datasets and propose the methodology to deal with it. Utilizing the processed dataset with the existing architectures for end-to-end learning yields better results. The rest of the paper is organized as follows. Section 2 highlights the problem with the simulator datasets and presents a literature review on existing techniques to solve the problem. Section 3 discusses in detail the rationale of the problem and presents the proposed methodology to solve it. Section 4 discusses the obtained results and Section 5 concludes the paper.

2 LITERATURE REVIEW

When driving a car, we usually experience a straight road that is being followed for a longer period and thus, the driver conditionally takes a turn, enabling the steering wheel to rotate in either direction. This phenomenon causes the car to usually stay at zero turn and therefore, during data collection, we get highly zero biased data, which affects the learning process. Therefore, the procured dataset needs few pre-processing steps in order to minimize its negative effect.

A very simple approach to deal with the over-represented zero steering angles uses a threshold which results in elimination of excess zero steering angles (Upamanyu and I., 2021; Tripathi et al., 2019; Sokipriala, 2021). Different thresholds are experimented and one that performs the best is selected. A possible problem in this approach is that there is a very large bias in simulator datasets which means a significant reduction in dataset. Lokhande et al. eliminated 15000 out of 20000 of its data-set. (Lokhande et al., 2021). Samak et all eliminated seventy percent of the zero-steering data. (Samak et al., 2021) Due to significant reduction in dataset, most of the authors that used this thresholding method have used data-augmentation to increase their dataset.

Data augmentation techniques without thresholding method are also used to tackle the issue of zero bias. Since more turns are desired to be present within the data, we can simply flip the images with turns and multiply the steering angle by -1. This would also ensure that we get same amount of left and right turns(Koci´c et al., 2019; Bojarski et al., 2018). Furthermore, the images of road turns can be further augmented by adding shadows, dark-spots, or adjusting brightness to avoid over-fitting of the model while reusing the same images thus allowing larger dataset with turns and not just zero angles.

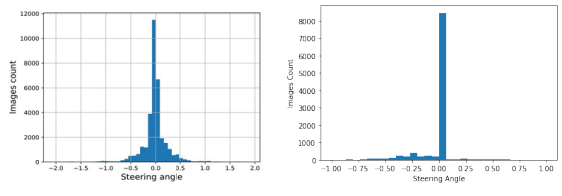
Another method used for eliminating zero bias is using probabilistic dropping of over-represented steering angles. In this approach bins of steering angles are created and the average number of samples per bin is computed. Then a keep probability is created for each bin. Keep probability is 1 for bins that have less samples than the average number of samples and for others it is number of samples for each bin divided by average samples per bin(Farag and Saleh, 2017). This way the over-represented steering angles are dropped proportionally with their over-representation. This is a contrasting approach to the data augmentation approach. A major advantage here is that the dataset does not increase which makes learning faster as unnecessary over-represented angles that make learning hard for the model are decreased.

A unique technique was also used where during the dataset collection mode of the simulator, the driver forcefully kept the car on the edge of the road which enabled the dataset to be generated with fewer steep angles and a smooth steering trend overall. This also minimized the zero bias as mostly the car was taking small turns instead of being kept in its steady state (Koci´c et al., 2019). However, this would negatively impact the overall learning as the model would be over-fitted on taking turns only at the edge of road and thus, it would rarely run at the center of the road and this creates a higher chance of the car getting off the road.

Each of the above-stated methods have different disadvantages. In the following Section, we discuss the inherent problem that causes zero bias in the dataset that is not addressed by the methods used in literature.

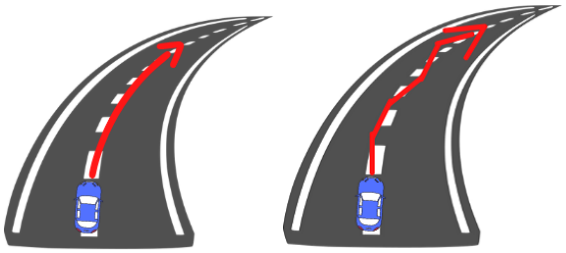
3 METHODOLOGY

In the first subsection we discuss the problem inherent in self driving simulator datasets and the reason why the earlier described methods from the literature are unable to cater to them. The subsequent section focuses on the methodologies to tackle the problem.



(a) Challenge Dataset (b) Simulator Dataset

Figure 1: Udacity Dataset Steering Angle Histogram.



(a) Reality (b) Simulator

Figure 2: Driving a Car in Reality vs Simulator.

3.1 The Issues with the Simulator Datasets

There are two datasets for Udacity self driving cars: one is the Udacity Challenge Dataset, that comprises real life driving images, and the second is the Udacity simulator dataset where data is collected while driving in the simulator. When the histogram of steering angles for the two datasets is plotted, the zero-bias problem seems a bit more exaggerated in the simulator dataset. However, when we increase the quantization bins of the histogram, the unique issue with the simulator dataset is quite visible as shown in Figure 1.

There is a huge zero bias in the simulator dataset which suggests there must be another reason for it in the simulator dataset. The zero bias in Udacity challenge dataset is because of majority of turns being very mild, whereas the zero bias in simulator dataset is because of how the data is collected. While driving on a real road, the driver adjusts the steering angle continuously and swiftly as shown in Figure 2a. Here the steering wheel is not rotated drastically but makes small swift changes. However, driving in a simulator pressing the turn right/left keys discretely creates a lot of zero bias as shown in Figure 2b.

As can be seen in the Figure 3, the road has a right turn. In the simulator the driver presses the turning keys 4 times in total and the car seems to be correctly navigating the turn. This is causing the corruption of our dataset. e.g., if we collect 20 images, they will all be of a right turn. However, a majority of the corresponding steering angles collected will be 0. This means that the correlation between images and steering angle is corrupted. Now even if we augment the images, or drop the images probabilistically to decrease zero bias, this problem will be inherent in our data where many images that are visibly turning right, will have zero steering angle.

The problem with the simulator dataset was identified by (Farag and Saleh, 2017) as well but the actual problem that causes some data points to make the learning harder for the model was not discussed. They created a subroutine to display those frames from the dataset on which the model does not perform well. These data-points were found to be miss-labelled and were manually corrected. Despite this technique being tedious, it helped improve the results to some extent. It is apparent that the problem faced by them was because of the zero bias problems with the simulator dataset as can be seen in Figure 3.

In Figure 4 a lot of random fluctuations to 0 can be seen. Although the two techniques of data augmentation and probabilistic dropping work well to reduce zero bias in Udacity challenge dataset, they fail to address the main problem in the Udacity simulator dataset as stated above. Apart from this, another issue is the frame sequence preservation. Both the preprocessing methods disrupt the temporal frame sequence of the data. A growing number of papers have used LSTMs recently to capture the temporal information in the data as well since it consists of frames of a temporal video. They have also shown that LSTMs outperforms its non-LSTM counterpart models. So there is a need for preserving temporal sequence of data while addressing the zero bias problem in simulator datasets.

3.2 Filtering to Smoothen Steering Angles

We propose smoothing the steering angles as an alternative to data augmentation and probabilistic dropping of data to remove zero bias. We achieve this by

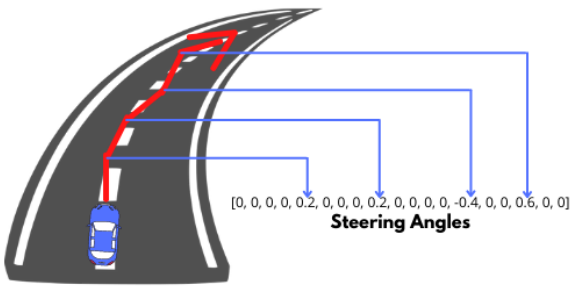


Figure 3: Discrete Turns in Simulator Dataset Creates Zero Bias.

a moving average filter that acts as a low pass filter and removes the high fluctuations of angles. We have used a filter with n length and all filter coefficients as 1.

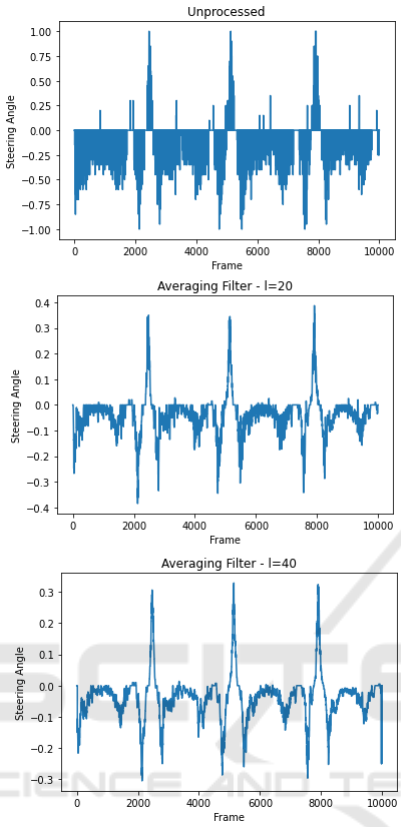


Figure 4: Affect of Moving Average Filter.

Since we are changing y-labels (steering angles), when we change our length of filter the input is not constant hence we can not compare the loss of different models with filters of different n. So we had to analytically observe how well the car is driving using the Simulator.

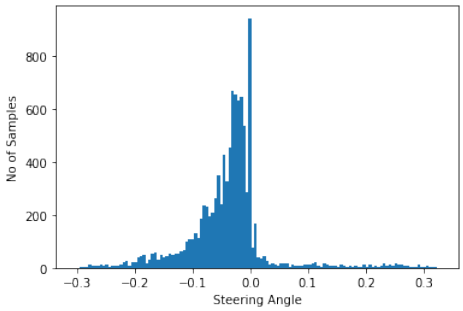
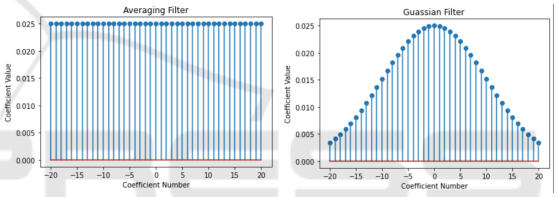


Figure 5: Steering Angle Histogram After Moving Average Filter.

As can be seen in the histogram in Figure 5, the steering angle distribution becomes unbiased. However, our samples still have very few right turns which could be improved by having larger dataset.

As can be seen in Section 4, the results from the averaging filters were acceptable so we also explored a Gaussian filter with weighted coefficients. It decreases the impact of neighbouring steering angles on the computation of the current steering angle. Gaussian filters have been used in preprocessing of self-driving datasets (Kim and Canny, 2017), but they are only used to remove sensor noise and mild driver steering angle fluctuations and did not target the zero bias issue in simulator datasets that is addressed in this paper.

The coefficients of simple averaging filter and Gaussian filter coefficients, for a variance of 10 and length of 40, are shown in Figure 6. The length as well as the variance of the Gaussian filter was varied to observe the performance on the Udacity Simulator and the results are summarized in Section 4.



(a) Averaging Filter (b) Gaussian Filter

Figure 6: Simple Averaging vs Gaussian Filter Coefficients for n = 1/40, Variance = 10.

3.3 NVIDIA Architecture

Here we outline the network architecture, used to test the proposed filtering methodology, which is a minor revision of the architecture released by NVIDIA for self-driving cars as shown in Figure 7. 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. Here we omit the first layer of the actual network which performs image normalization because our preprocessing method already normalizes the images before feeding it into the architecture. The five convolutional layers were designed to perform feature extraction. We use strided convolutions in the first three convolutional layers with a 2×2 stride and a 5×5 kernel and a non-strided convolution with a 3×3 kernel size in the last two convolutional layers.

The five convolutional layers were followed by three fully connected layers which then output the steering angles. In order to reduce over-fitting, dropout (0.2) layer was used. Adam optimizer was

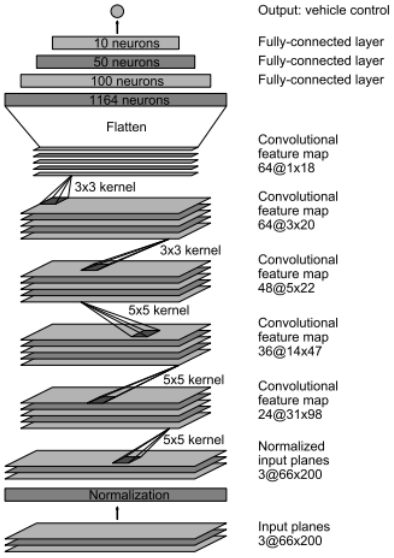


Figure 7: NVIDIA Architecture (Bojarski et al., 2018).

used for optimization which requires little or no tuning as the learning rate is adaptive(Bojarski et al., 2018).

3.4 Post-processing

Since the dataset was initially zero biased, the applied filter reduced the proportional gain of the steering angles, causing the car to take accurate but smaller turns. As a result, the model's output needed to be tuned so that the predicted value could be given an appropriate gain and bias to make it as effective as it was in its original state.

The results were obtained after such post-processing as the simulator variables were tuned to make the car steer near perfectly. Different filters required unique tuning. The equation below maps the acceleration of the car based on steering angle and speed of the car.

$$T = 1.0 - (SA)^2 - (\frac{S}{SL})^2.$$

Here, T is throttle, SA is steering angle, S is speed, and SL is speed limit. Since the filter was applied onto the steering angles and is an irreversible operation, simple manual tuning could be done to attenuate and bias the prediction from the model towards a much better result. Therefore, the Steering angle was calculated using following equation:

$$SA = (predict(image) * M) + B,$$

where predict(image) function sends the captured image to the model and outputs the steering angle, M is an attenuating constant, and B is a bias. This equation can be manually tuned to get better results as by default, M is 1 and B is 0.

4 RESULTS

Since the input to different lengths of averaging filters are different, we can not effectively compare losses between them to determine which works best. The losses only give us the idea that which filtered data, the particular value of n, is being learned well by model. The losses at different values of n filter length are shown in Figure 8.

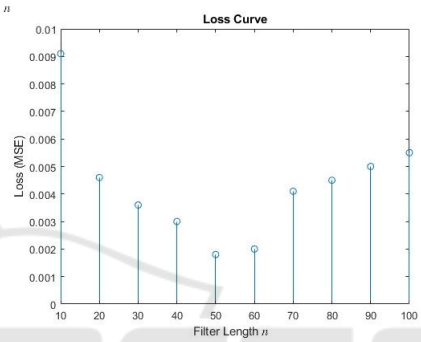


Figure 8: Loss with Different Filter lengths.

Analytically, we ran all the models on Udacity Simulator to observe which filter worked the best. We observed the simulations for the averaging filter length n between 10 and 100. The simulations were run multiple times for each n, and average was taken to calculate the average time a car stays on track without crashing. The results are summarized in the Figure 9.

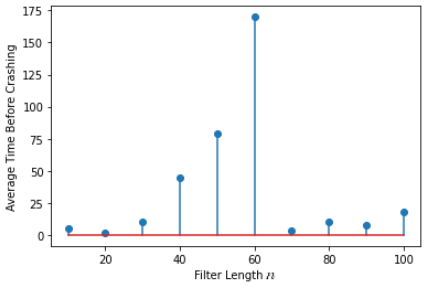


Figure 9: Performance of Car on Udacity Simulator with Different Filter Lengths.

Then we evaluated the performance of our Gaussian filter preprocessing method. Performance on the Udacity simulator was observed for different parameters (length and variance) and the length of 40 and variance of 10 were found optimal where the car took

multiple laps smoothly over the track. These results were obtained with a dataset of just 3500 images and was ran on the NVIDIA architecture for 10 epochs. This makes the performance even more commendable.

5 CONCLUSION

In this paper we analyzed the inherent problem of zero bias with self driving simulator datasets that was not adequately addressed before. We proposed filtering strategies for solving the issue and the results were found to be acceptable. The zero bias issue can be further reduced by creating input devices that has less latency and more frequent feedback from its sensor. With such devices, our proposed solution could further enhance the performance of the model as it participates in making the dataset more unbiased compared to its earlier form and therefore, assisting in simulator based training of such applications.

REFERENCES

Bojarski, M., Testa, D., and et al. (2018). End to end learning for self-driving cars. In *CoRR*. arXiv.

Coelingh, E., Nilsson, E., and Buffum, J. (2018). Driving tests for self-driving cars. In *IEEE Spectrum*. IEEE.

Farag, W. and Saleh, Z. (2017). Safe-driving cloning by deep learning for autonomous cars. In *International Journal of Advanced Mechatronic Systems*, page 390.

Greenblatt, N. (2016). Self-driving cars and the law. In *IEEE Spectrum*. IEEE.

Hars, A. (2010). Autonomous cars: The next revolution looms. In *Thinking outside the box: Inventivio Innovation Briefs*. Inventivio.

Kim, J. and Canny, J. (2017). Interpretable learning for self-driving cars by visualizing causal attention. pages 2961–2969.

Koci´c, J., Jovici'c, N., and Drndarevi´c, V. (2019). Adriver behavioral cloning using deep learning. In *Proceedings of the 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5.

Lokhande, S., Khan, M., and Pandey, G. (2021). Decision system for a self-driven car. In *Journal of Science and Technology, Vol. 06, Special Issue 01*, pages 365–370. Journal of Science and Technology.

Samak, T., Samak, C., and Kandhasamy, S. (2021). Robust behavioral cloning for autonomous vehicles usingend-to-end imitation learning. In *arXiv preprint*. arXiv.

Sharma, C., Bharathiraja, S., and Anusooya, G. (2020). Self driving car using deep learning technique. In *International Journal of Engineering Research & Technology*. IJERT.

Sokipriala, J. (2021). Prediction of steering angle for autonomous vehicles using pre-trained neural network. In *European Journal of Engineering and Technology Research*, pages 171–176.

Tripathi, R., Vyas, S., and Tewari, A. (2019). Behavioral cloning for self-driving carsusing deep learning. In *Proceedings of International Conference on Big Data, Machine Learning and their Applications (ICBMA)*, pages 197–209. Lecture Notes in Networks and Systems.

Upamanyu, K. and I., R. (2021). Effects of image augmentation on efficiency of a convolutional neural network of a self-driving car. In *Journal of Student Research*. Vol. 10 issue 2.