Autonomous Driving on Nvidia Dave-2

Document revision of “End to End Learning for Self-Driving Cars ” and adaptation on Udacity Simulator

Bruckner Calderon, Patrick*
Universidad Privada Boliviana
La Paz, Bolivia
patbruckner69@gmail.com

Hervas Okabe, Ken*
Universidad Privada Boliviana
La Paz, Bolivia
knhrvs@gmail.com

Martínez Moreno, Fernanda*
Universidad Privada Boliviana
La Paz, Bolivia
f.l.m.m.0027@gmail.com

Ugarte Yaffar, Jordi Martin*
Universidad Privada Boliviana
La Paz, Bolivia
jordi93@gmail.com

***Abstract*—The present document synthesizes structure, scope, applications and experimentation on the End-to-End Dave-2 system by Nvidia in 2016 for autonomous, optimal and safe driving [1].**

Keywords—Behavioural Cloning, Data Augmentation, Flattening, Ground Truth, Self-Driving Cars.

I. Introduction

Nowadays, there is a wide variety of Computer Vision-based systems for autonomous driving. Most of these models train through videos and sensors to perceive objects and people in order to avoid crashes against these. Other projects interpret traffic signs and identify streets to know which path to follow. Nvidia's Dave-2 system was developed to learn in which direction to steer according to images of different roads.

II. Base Model and Concepts

Nvidia introduces a model that trains with each frame of more than 70 hours of video captured by three cameras relative to a driver's perspective. It associates the images with a steering wheel angle and a Ground Truth: a position relative to the center of the track [1].
Reviewing some of the concepts on which the model is based upon we have:

A. Convolutional Neural Networks

These are Deep Learning models designed to work with images as inputs and learning weights to certain elements to differentiate them from each other. They can detect simple contours as well as recognize complex details [2].

B. Flattening

Matrix transformation resulting from the convolutions to a one-dimensional vector connected directly to the final layer of neuronal activation classifying the inputs [2].

C. Standardization

Modification of input images with padding techniques, saturation, brightness, and such, thus speeding up training and improving the model’s generalization capacity [4].

D. Data Augmentation

Technique used to augment data by making slightly modified copies of elements on the dataset. It acts as a regulator reducing overfitting when training a model [4].

Elucidating more about the variations of the present implementation in comparison with the original, it is shown that the DAVE-2 base model is almost entirely kept, but there are some changes regarding the collection system and the image augmentation as well [3]. Due to the way the images are collected in this implementation (using only one camera instead of the original three) the image augmentation is performed in a more conventional way using zoom in and zoom out in some areas in combination with shifting and flipping the image. Although the original three-camera system is arguably more elegant in some form, the present implementation is using a simulation for data collection and only one image must be manipulated [5]. Nevertheless, the results obtained through this method are, at least, satisfying, as they are discussed next.

III. Experimentation and Results

Before altering the Dave-2 system, it is worth noting that Nvidia has already optimized the model and obtained the exact values that guarantee a formidable performance [3]. For this reason, experimentation maintains the pre-established values and hyper-parameters. Moreover, since the implementation is developed in the Udacity environment, the system has a tendency towards underfitting.

The experimentation was focused on modifying data used for training and validation which is composed of images and the wheel’s angle captured at continuous stages of a lap [5]. Initially, the experiment was solely directed on increasing the number of laps that were recorded as data for the model. However, the most autonomous model on this phase was only trained with 3 laps. Although it was able to navigate the course on its own, the agent showed a certain abruptness that could be mistaken as a sign for a human counterpart to interfere with its driving. Therefore, other variables were employed to smoothen the agent’s driving.

First, human intervention was implemented through a mouse instead of a keyboard, which helped to smooth the turning angle and ensured better turning. Then the car was trained using “forward and backward” laps; being driven clockwise through the course and another lap was driven

*All authors have contributed equally

2021 Selected Topics in Artificial Intelligence. Professor Gissel Velarde

counterclockwise. This allowed more variety of data to appear during training and validation, making the car generalize better relative to escenarios with strange textures. The last modification was slowing down on tricky regions during the drive. Due to the nature of the simulator, textures were prone to suddenly pop in as the engine rendered the scene or shadows could be confused with the road depending on the graphics quality of the simulator. Slowing down allowed more data to be captured and turning tricky regions on images represented on the dataset.

Nvidia's model was given an autonomy grade through the following equation:

$$autonomy = (1 - \frac{\#\ of\ interventions * 6\ [seconds]}{elapsed\ time\ [seconds]}) * 100$$

FORMULA I: Autonomy math calculus for the original end-to-end system [3].

The model was able to achieve a 98% autonomy score (1 intervention in 300 seconds). The resulting and most autonomous model for this paper was trained for 5 laps using a mouse, forward and backward laps, and change of speed on tricky regions. This model is 98% autonomous and can complete several laps on its own, nonetheless it can rarely fail due to the texture issues and nature of the simulation that was mentioned during the experiment phase. It is worth noting that the model trained with 3 laps, without the additional variables, was more prone to commit mistakes due to the simulation despite the fact that it could also drive several laps autonomously.

TABLE I: RESULTS

Laps	Change of Speed on Tricky Regions	Collected Forward and Backwards Training Data	Controller	Completed at least 3 laps without interventions	Autonomy Rating
0	No	No	Keyboard	No	76%
1	No	No	Keyboard	No	80%
2	No	No	Keyboard	No	84%
3	No	No	Keyboard	Yes	86%
4	No	No	Keyboard	No	84%
5	No	No	Keyboard	No	84%
3	No	No	Mouse	No	80%
3	No	Yes	Mouse	No	84%
3	Yes	Yes	Mouse	Yes	92%
5	No	No	Mouse	No	84%
5	Yes	No	Mouse	Yes	96%
5	**Yes**	**Yes**	**Mouse**	**Yes**	**98%**

TABLE I: Result table obtained through different experimentation cases.

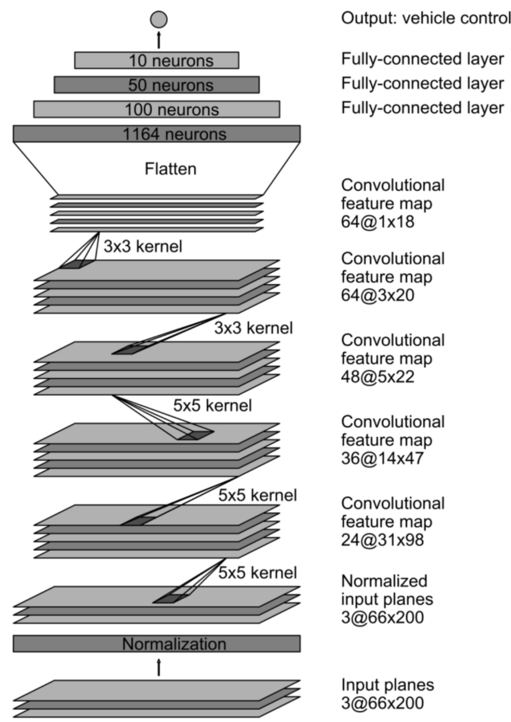
IV. CONCLUSION

Dave-2 by Nvidia is a model already implemented today for driving on tracks [1]. Although it needs improvement, it has inspired other models and allows more models to mimic human behavior. When it is implemented on a simulator with no real risk, it is perfect for smart driving on low resources than the model would need in real life. That is why the model implemented on this paper performs well on simple and virtual environments like the ones tested in the Unity Engine [5]. The model was successful in replicating human behavior, even though it was prone to some complications due to the nature of the environment it was trained on. Nonetheless with relatively low data, in comparison to 75 hours of footage [3], it successfully achieved around the same autonomy as the Nvidia model and was able to drive several laps without issues.

CODE: LOCAL RUNNING

https://drive.google.com/file/d/1geWOePnPXam_sbIoS2vS8XCViA9xUaJC/view?usp=sharing

ATTACHMENT A: NETWORK STRUCTURE



ATTACHMENT A: Visual representation for the CNN. Source: [3].

REFERENCES

[1] Kawakura, S., & Shibasaki, R. (2020). Deep Learning-Based Self-Driving Car: JetBot with NVIDIA AI Board to Deliver Items at Agricultural Workplace with Object-Finding and Avoidance Functions. *European Journal of Agriculture and Food Sciences*, *2*(3). https://doi.org/10.24018/ejfood.2020.2.3.45

[2] Bazan Yaranga, C. (2018). Transferencia de Aprendizaje Mediante Redes Neuronales Convolucionales para el Reconocimiento de Conductores Distraídos. *TECNIA*, *28*(2). https://doi.org/10.21754/tecnia.v28i2.549

[3] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zieba, K. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[4] del Egio, J., Bergasa, L. M., Romera, E., Huélamo, C. G., Araluce, J., & Barea, R. (2018, November). Self-driving a Car in Simulation Through a CNN. In *Workshop of physical agents* (pp. 31-43). Springer, Cham.

[5] Computer Vision Zone. (2021, July 15). Self Driving Simulation using NVIDIA's Model. Retrieved October 28, 2021, from https://www.computervision.zone/courses/self-driving-simulation-using-nvidias-model/