Implementation of an autonomous car in a simulated environment

Atharva Kunte
Department Of Computer Engineering
D.J Sanghvi College of Engineering
Mumbai-56, India
askunte13@gmail.com

Avasyu Bhatia
Department Of Computer Engineering
D.J Sanghvi College of Engineering
Mumbai-56, India
avasyubhatia@gmail.com

Ziyad Dhuka
Department Of Computer Engineering
D.J Sanghvi College of Engineering
Mumbai-56, India
ziyaddhuka7@gmail.com

***Abstract*— Our system is an implementation of driverless cars in a simulated environment such as a car driving simulator. The chief reason a simulated environment was chosen over real world is to mitigate the risk to human life. Also, a simulated environment would reduce the material losses as well as the overall cost of the implementation process. Once the system has attained a high proficiency in the simulated environment, the system can be migrated and perfected for the real world with some modifications. The software will take the first person view of the car and feed the image to our system. We train our system to map raw pixels from a single front-facing camera directly to steering commands in our simulation.**

Keywords—neural networks, deep learning, image processing

I. Introduction

Transportation is one of the most influential aspects of our modern society. Vehicular transportation has transformed the world as we know it. Travelling from one point to another at far off distances has become very easy because of vehicular transportation. But what we're unable to ensure is safety. In 2015, 35,092 people died in car accidents. Someone dies once every 88 million miles driven. That gives you about a 0.011% chance of dying in a car accident in any given year, or 0.88% in your lifetime. 2.6 million people are injured in vehicles every year. This doesn't just cause damage to human life but also causes property damage on a massive scale. The leading cause of most automobile accidents today is driver error. Alcohol, drugs, speeding, aggressive driving, over-compensation, inexperience, slow reaction time, inattentiveness, and ignoring road conditions are all contributing factors.

The top four causes of accidents are:

- Distraction
- Speeding
- Drunk Driving
- Recklessness

The causes can be mitigated and eliminated by self-driving cars. The possibility of distraction is negligible in a self-driving car as the car will constantly have its detection system examining its environment. By properly training the algorithm to control its speed based on the conditions, the possibility of accidents due to speeding can be reduced. Drunk driving and other human-related issues don't exist for a self-driving car. Also, by properly training the algorithm reckless driving can be avoided.

Accidents in case of self-driving cars are possible only from car failures (if the wheel falls off it isn't going to matter much who is driving), sensor failures (the computer can't see the road or other cars), or software bugs. The latter will only happen once because it gets added to a regression test and all the cars running that software automatically learn how to avoid that mistake.

Apart from reduced accidents, another major benefit of self-driving cars is improved efficiency. One of the leading causes of traffic jams is selfish behavior among drivers. It has been shown when drivers space out and allow each other to move freely between lanes on the highway, traffic continues to flow smoothly, regardless of the number of cars on the road. In fact, we have the capability of pretty much eliminating traffic jams right now. All we'd have to do is allow three to four car lengths of space between our car and the car in front of us, even in slow-moving traffic. The way we drive now, when traffic gets heavy, if someone needs to change lanes to exit the freeway, or if someone needs to enter the freeway, everybody has to stop to let it happen because we drive packed so tightly together. And, there’s no other way to say it, we do this out of selfishness. Every time we have ever proffered this theory to a group of drivers, their first response is if we drove that way, everybody would get in front of us. Which, of course is exactly the idea, if we allowed cars to get in front of us and freely change lanes, traffic would continue to flow. When you don’t allow cars to get in front of you traffic has to stop, your frustration level increases, and you become determined to let even fewer cars get in front of you—thus exacerbating the problem. Self-driving cars can be programmed to space out automatically, thereby eliminating the problem.

II. LITERATURE SURVEY

A. *Resizing*

In this operation the image collected needs to be processed by the neural network. Since neural networks takes more processing time for an image, to reduce the complexity we resize the obtained image [1].

B. *Data Balancing:*

In this step the data of the key-press is balanced. While training the neural network we need to make sure that we don’t feed biased data to the neural network as giving a biased data would decrease the accuracy of the network. We make sure that the key-press input given is balanced i.e. number of left, right, up (acceleration), down (deceleration) key-press data fed to the network is equal. Hence, we delete the input data such that the number of key-press for a key will be equal to the minimum number of key-press of a key. This will prevent the neural network from getting a biased data [2].

C. *Convolutional neural networks:*

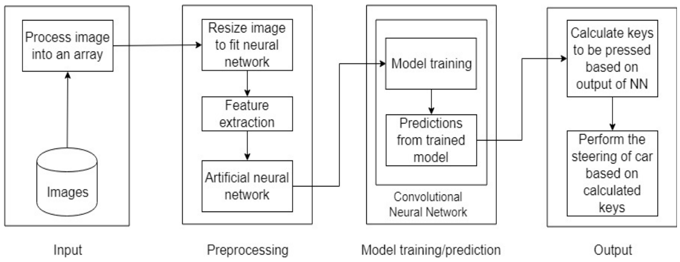
The important breakthrough of CNNs is that features are now learned automatically from training examples. The CNN approach is especially powerful when applied to image recognition tasks because the convolution operation captures the 2D nature of images. By using the convolution kernels to scan an entire image, relatively few parameters need to be learned compared to the total number of operations. The CNNs that we describe here go beyond basic pattern recognition. We developed a system that learns the entire processing pipeline needed to steer an automobile. The first step to training a neural network is selecting the frames to use. Our collected data is labelled with road type, weather condition, and the driver’s activity (staying in a lane, switching lanes, turning, and so forth). To train a CNN to do lane following we only select data where the driver was staying in a lane and discard the rest. We then sample that video at 10 FPS. A higher sampling rate would result in including images that are highly similar and thus not provide much useful information.

To remove a bias towards driving straight the training data includes a higher proportion of frames that represent road curves. The convolutional neural network will train itself to identify the lanes, traffic lights, obstacles like pedestrians and the cars besides it. The more the data for training, greater the accuracy of the neural network [3] [4].

D. *Feature extraction:*

The features obtained from the input images are the size and positions of obstacles like pedestrians, other vehicles, etc. After obtaining the positions, the car will try to maneuver away from the obstacles and avoid colliding into them. The output from the feature extraction phase will be passed to an artificial neural network which will do the job of predicting the proper movement of the car [5] [6].

III. PROPOSED ARCHITECTURE



The architecture of the system is divided into 4 major parts- input, preprocessing, model training/ prediction and output. In the first part, the input images are captured. Images are stored in the form of a 2D array. In the next phase, the images are resized so as to make the next steps easier. Features are extracted from the images like pedestrians, other vehicles, obstacles, etc. The positions of these objects are identified and the car avoids colliding with them. Next, a convolutional neural network is used to calculate the maneuvers for the car. The neural network predicts the steering angle and throttle values which will take the car to the selected location while avoiding any collisions. In the final phase, the output of the neural network is mapped to the steering commands (key presses) for the car.

IV. EXPECTED OUTCOME

Our system will detect its surroundings and based on past experiences and training provided to algorithm, make decisions about driving. The system will detect the road, the lanes drawn on it to classify areas as drivable. After detecting drivable area the system will maintain in its path in the drivable area. The system will also detect other objects such as vehicles, humans, animals, obstructions, etc.; identify their speed relative to the vehicle and take decisions based on those external factors. Thus, the system will try to eliminate the risk to human life and improve the efficiency of road travel.

Fig. 1. Simulator shows first person view from the vehicle

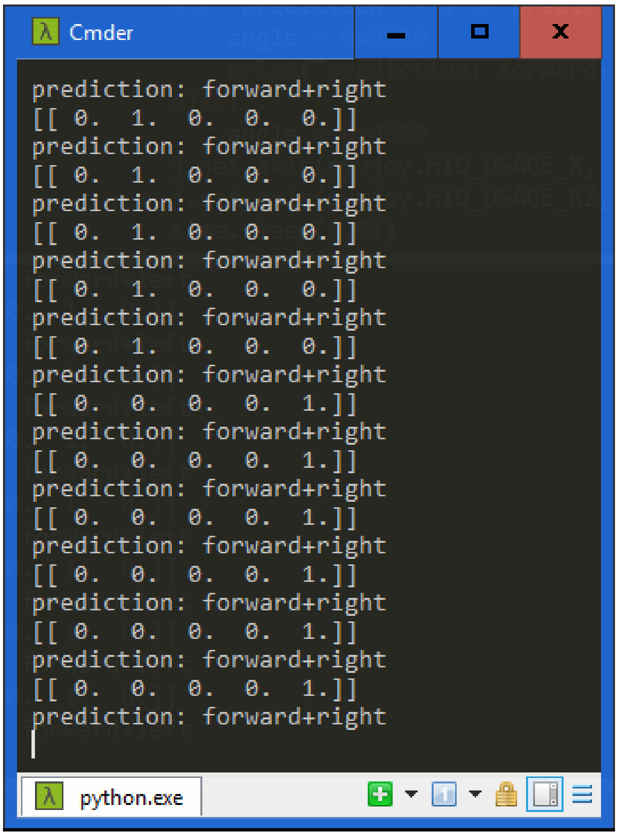


Fig. 2. Predicted direction and keys to press

ACKNOWLEDGMENT

The authors would like to thank the Department of Computer Engineering at D J Sanghvi College of Engineering and in particular Prof. Ruhina Karani for her assistance and Head of Department Dr. N.M. Shekokar. We would further like to thank our Principal Dr. Hari Vasudevan for making the appropriate resources available.

REFERENCES

[1] Torralba, Antonio, Rob Fergus, and William T. Freeman. "80 million tiny images: A large data set for nonparametric object and scene recognition." *IEEE transactions on pattern analysis and machine intelligence* 30.11 (2008): 1958-1970.

[2] He, Haibo, and Edwardo A. Garcia. "Learning from imbalanced data." *IEEE Transactions on knowledge and data engineering* 21.9 (2009): 1263-1284.

[3] Bojarski, Mariusz, et al. "End to end learning for self-driving cars." *arXiv preprint arXiv:1604.07316* (2016).

[4] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." *Neural computation* 1.4 (1989): 541-551.

[5] Glasius, Roy, Andrzej Komoda, and Stan CAM Gielen. "Neural network dynamics for path planning and obstacle avoidance." *Neural Networks* 8.1 (1995): 125-133.

[6] Gavrila, Dariu M., and Vasanth Philomin. "Real-time object detection for" smart" vehicles." *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 1. IEEE, 1999.

[7] Waldrop, M. Mitchell. "No drivers required." *Nature* 518.7537 (2015): 20-20.

[8] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *Nature* 521.7553 (2015): 436-444.

[9] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[10] Huang, Bing-Qiang, Guang-Yi Cao, and Min Guo. "Reinforcement learning neural network to the problem of autonomous mobile robot obstacle avoidance." *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*. Vol. 1. IEEE, 2005.