CHAPTER 9

Automated Guided Autonomous Car Using Deep Learning and Computer Vision

Anubha Parashar and Vidyadhar Aski

Manipal University Jaipur
Jaipur, India

Apoorva Parashar

Maharshi Dayanand University
Rohtak, India

CONTENTS

DOI: 10.1201/9781003048381-9

9.1 INTRODUCTION

A smart car generally has two controlling units that dictate its actions, the controllers of low magnitude and high magnitude. The controller of high magnitude takes input from its components—i.e., the driver (in this case the camera), the surroundings such as the traffic, and the sensors in place. Then, after deducing the correct actions to be taken, it sends signals to the low-level controller, which controls the brakes, steering, engine, and throttle [1]. To do this successfully, controllers needs to understand driver psychology—e.g., how and when a specific maneuver is necessary, which changes with the terrain and the area in which the car is driving—since driver temperament and driving style cannot be universally generalized. To understand and correctly predict such outcomes is risky. Studies show that if the driver is given even an extra half a second before a collision, 60% of accidents can be avoided, and this percentage increases to 90% if one second of warning time can be provided [2]. Such results stress the importance of timely and correct decisions, but the conventional architecture of a smart car system faces problems in this area. Also, when cars rely on so many sensors, it is difficult for them to adapt to new surroundings and to reconfigure the system to achieve a different goal based on learning that occurs on the move, especially when noisy sensor data is received [3]. The foundation of the end-to-end approach was laid almost 14 years ago when a project sponsored by the Defense Advanced Research Projects Agency (DARPA) created DAVE) [4], an autonomous car that ran along an alley filled with junk [5]. The project demonstrated that this approach was viable and adequate for the functioning of a smart car. The end-to-end approach thus provided alternatives that were uncomplicated and also easier to test [6]. With the advancement in technology, the approach achieved similar functionality more skillfully and with limited resources. At the center of this end-to-end learning is the convolutional neural network (CNN), which performs the driver's actions based on the training. The data was not hard to collect, and the system itself could be reproduced efficiently at a low cost, which made it possible to do rigorous experimentation. The data from the system could also be shared with another system of the same kind to improve performance.

One of the breakthroughs in navigating autonomous vehicles happened because of using CNN [4, 6–16]. In order to navigate the vehicle while providing complete passenger safety, the CNN model has been deployed and trained as well as tested. This model maps the complete driving model in real time by collecting information about driving behavior and taking images from the angle of the steering wheel and camera [6, 10, 17–29]. Vehicle performance is heavily dependent on the data set; i.e., if the vehicle encounters a new hazardous situation (one on which the model has not been trained), the system will not provide a good analysis, and an accident might happen. There are a lot of reasons that the system might malfunction and crash, such as software failure or camera sensor impairment.

The rest of the chapter proceeds as follows. In the literature review, we explore the ultramodern models used for self-driving vehicles. We also touch on the limitations of models based on conventional deep neural networks (DNNs). Then we gave introduce the approach preferred in this study and describe the CNN architecture used for autonomous

driving. In order to validate our method, we provide the testing and training results along with the real-time simulation of the vehicle drive.

9.2 LITERATURE REVIEW

This journey to make a smart autonomous vehicle began with the invention of modern cruise control in 1948 [30]. Many dedicated steps have been taken since then to make self-driving cars a reality. The autonomous land vehicle in a neural network (ALVINN diversified the field by combining end-to-end learning [31] with a neural network to enable it to solve any issue [10]. DARPA then gave perspective to what can be achieved with the technical know-how of the time [4, 32]. In the recent past, when CNN was trained to map camera pixels so they could be interpreted directly as guiding orders [6], it demonstrated the reliability of this approach in contrast to the more modular alternatives [33].

One of the most used techniques for self-driving vehicles consists of the deep neural network (DNN) [34]. A lot of car companies like Ford, Tesla, BMW, and Volvo swear by DNN. Many other firms such as Uber, Lyft, and Google are also deploying DNNs in their autonomous vehicles [35]. The working of such systems is based on the accurate transfer of information from RADAR, cameras, and LIDAR as input to the DNN, such as a recurrent neural network (RNN) or CNN [36]. The information is processed by the system in order to control the angle of the steering wheel and the velocity of the vehicle being driven. For instance, the models used by Udacity make use of both RNN- and CNN-based technique, whereas the models created by NVEDIA make use of only a CNN model for processing the input data used to take command of the steering wheel [37]. Our study has made use of a model based on a CNN just like that of DAVE-2 (made by NVEDIA).

A DNN-based system can sometimes misread the given data and cause treacherous consequences [38]. A lot of studies have been done on DNN-based models and they have determined that these systems are vulnerable to erroneous data and can malfunction easily [39]. For instance, we can disrupt a DNN-based system by adding a small inaccuracy in the image, causing the system to categorize it in a different group [40]. It has also been found that objects can be morphed physically by attackers in order to cause glitches in DNN-based self-driving systems [41]. These low-cost techniques used by attackers can cause inaccuracy in measuring distances, resolution, and angles. Likewise, barricades or debris present on the road can lead the system to produce incorrect outputs (Figure 9.1) [42–47].

9.3 PROPOSED METHODOLOGY

We proposed to use a toy car and mounted a Raspberry Pi on top of it. An 8 MP camera was inserted in the raspberry pi and attached in a way that gave it a clear view of the road ahead. An ultrasonic sensor was attached on the hood of the car to detect any obstructions on the road.

The car could be driven by a remote control (the one that came with the toy car) or could be driven autonomously (by applying DNNs). An Arduino Uno microcontroller board was used in this experiment to collect the data from the remote (the data here was input for the Arduino).

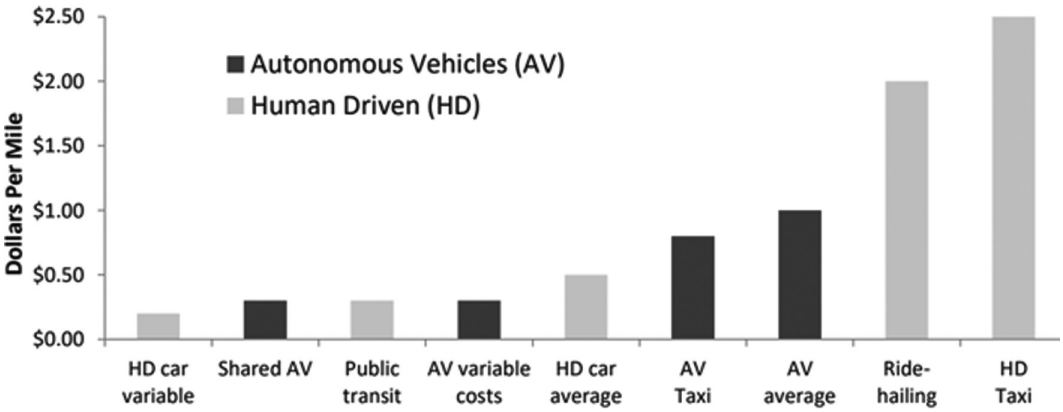

FIGURE 9.1 Comparison of driving costs.

A Wi-Fi module was also latched onto the Arduino in order to connect it to the built-in Wi-Fi of the Raspberry Pi. This way the Raspberry Pi would have access to the data coming from the remote control (data from the Arduino through Wi-Fi) and from the ultrasonic sensor (which would detect any road debris) as well as input data on the road (from the camera mounted on the Raspberry Pi).

The road was made with a sheet of black paper, and details were painted on it. Some barricades and debris were used to create obstructions. The model was trained on various road paths—straight, curved left, curved right, and circular. Testing was done in phases, and after completion, the model was given a final test path with a broad curvature.

9.3.1 Data Collection

Initially, Udacity's Self-Driving Car Simulator was used to generate the training data needed to test the equipment and the plan of action [48]. It gives the user the liberty to drive a car on preset tracks, and the data corresponding to how you drive is collected frame by frame.

The data also contains steering angle and acceleration on each frame. This data helped actualize the direction of this study and formulate the kind of data the project needed so that it could be mapped onto directions. Thus, the data had to be generated independently and was specific to the need of the project [49]. There were two major reasons for this. First, the remote-control car was not taking directions in the form of steering angle but rather only left and right. Second, the camera angle played an important role in how the car understood the road ahead, which greatly affected its decision-making capabilities.

Data was collected from the next set of tests by transmitting direction commands to car's remote and thus to the car via a laptop computer. The camera view was visible on the screen of a remote device using the VNC viewer to stream the camera's input [7]. The car was steered on self-laid tracks of different orientations that were similar to but not the same as those on which the car had been tested. The tracks mimicked unmarked roads and the environment where the car would be tested, and at times the car had to stop or turn to avoid having a collision or going off-road [50].

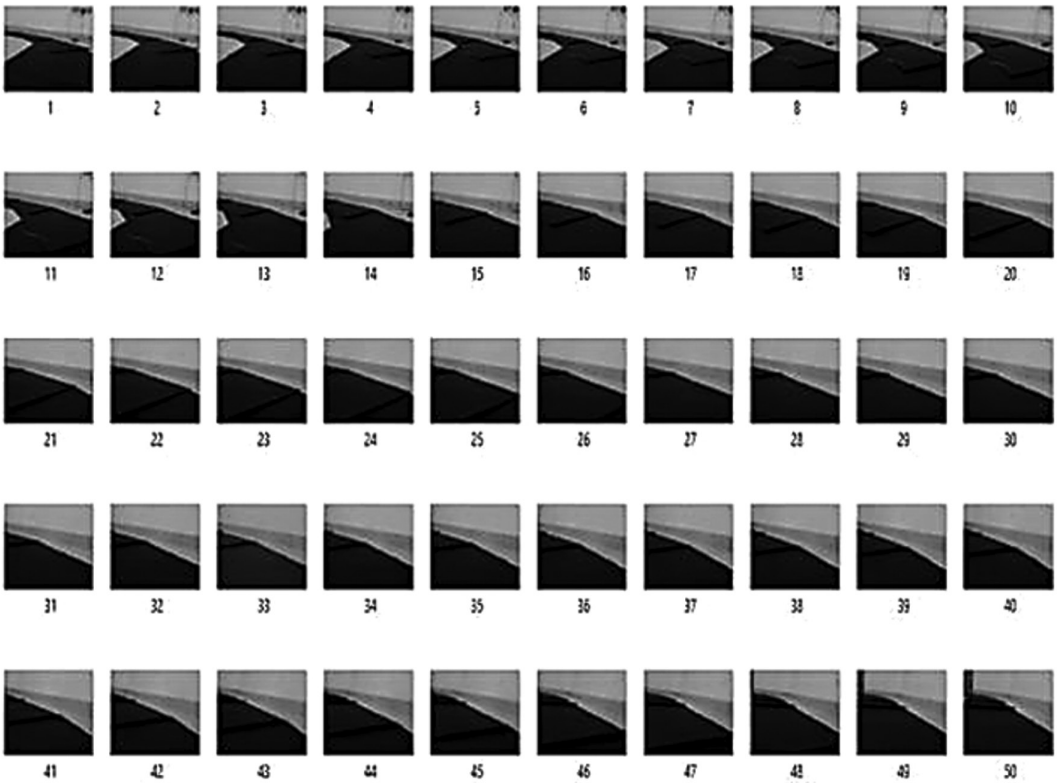

FIGURE 9.2 Sample section of the collected data set.

The frames were captured by the Pi camera staged on a remote-control vehicle, which also documented directions at every moment that were time-synchronized and embedded in the final data set. This was repeated on diverse tracks until a useful amount of data was collected. Thus, the data set had frames containing the tracks as viewed by the Pi camera along with the directions, as shown in Figure 9.2 [51].

9.3.2 Feature Extraction

In each image, unnecessary data was cropped out; i.e., the height of the images was reduced in order to increase the contributing information in the frame, and the images were resized so that less memory was occupied.

The images were then converted into the YUV color system, as it allows a systematic decrease in the quantity of information necessary to constitute images of equivalent quality [52]. To increase the sample data, images containing turns were flipped with the corresponding directions (Figure 9.3).

The brightness of undecided frames was increased to random degrees to account for the changes in lighting conditions, and the same procedure was followed to reduce the brightness of random images in order to make the model more robust.

Furthermore, Gaussian blur was applied to each frame to smooth the image and to reduce noise in the captured frame. Also, it was observed that a disproportionately large

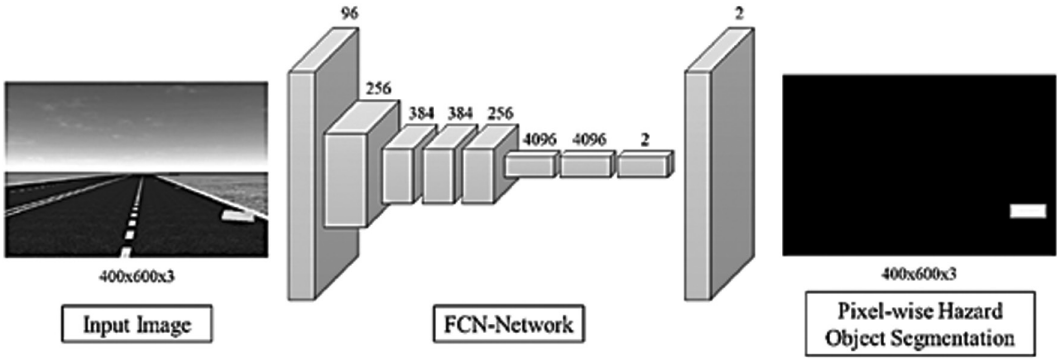

FIGURE 9.3 Feature extraction model using object detection and segmentation.

amount of the recorded data corresponded to the car driving in a straight direction as compared to that recorded while the car was making a right or left turn. Thus, the data had to be normalized to make the scales even and avoid the possibility of a biased model [53].

9.3.3 CNN Architecture

The CNN was trained on our collected data following the VGG16 model. It had 16 layers and used the softmax function as the classification layer, as shown in Figures 9.4 and 9.5. Transfer learning was implemented by freezing the top layer in order to give the desired output.

9.4 DISCUSSION AND RESULTS ANALYSIS

9.4.1 Experimental Setup

For this experiment, we used a toy car and mounted a Raspberry Pi on top of it. An 8 MP camera was inserted in the Raspberry Pi and attached in a way that it gave a clear view of the road ahead. An ultrasonic sensor was attached on the hood of the car to detect any obstructions on the road. The car could be driven by a remote control (the one that came with the toy car) or could be driven autonomously (by applying DNNs). An Arduino Uno microcontroller board was used in this experiment to collect the data from the remote (the data here was input for the Arduino). A Wi-Fi module was also latched onto the Arduino in order to connect it to the built-in Wi-Fi of the Raspberry Pi. This way the Raspberry Pi had access to the data coming from the remote control (data from the Arduino through Wi-Fi)

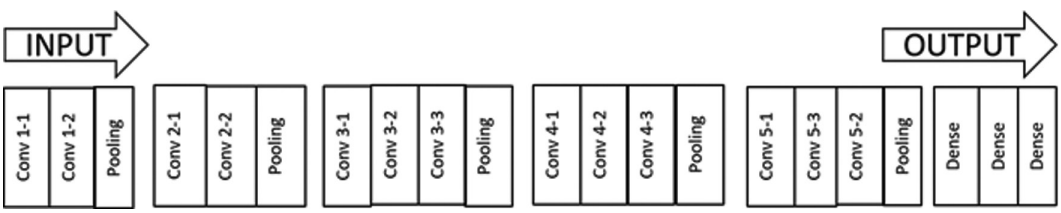

FIGURE 9.4 Deep learning layers used in the project.

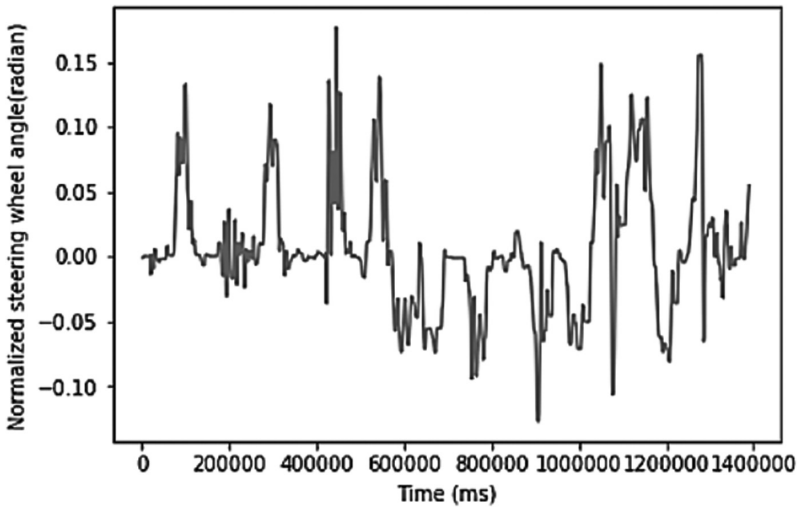

FIGURE 9.5 Results from the training data set captured the curvature of the steering wheel after normalization.

and from the ultrasonic sensor (for detecting any road lumber) as well as input data on the road (from the camera mounted on Raspberry Pi). The road was made with a sheet of black paper, and details were painted on it. Some barricades and debris were used to create obstructions. The model was trained on various road paths—straight, curved left, curved right, and circular. Testing was done in phases, and after completion, the model was given a final test path with a broad curvature, shown in Figure 9.6

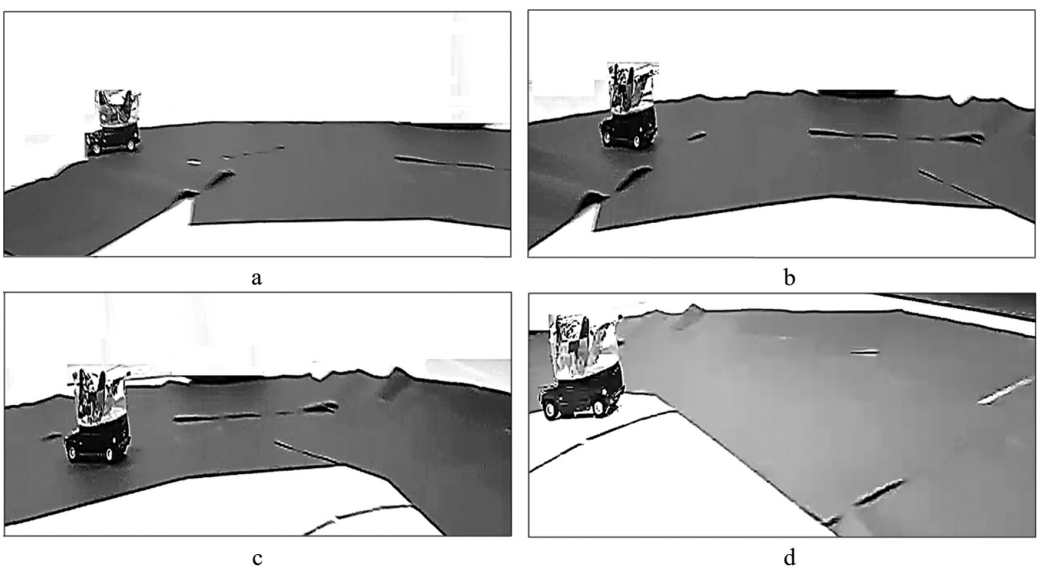

FIGURE 9.6 (a) The car is unable to detect the road on the right-hand side so it makes a left turn. (b) The car is on a circular curvature. (c) The car makes a sharp left turn. (d) The car is shown in a mismatched scenario.

9.4.1.1 Input Component (Camera)

A Pi camera attached to the Model B+ board of the Raspberry Pi, which draws its power from an external power bank, was used to collect input data. The camera was fitted on top of the car in such a way that it provided vision to the car and was overlooking the road and the immediate surroundings. The client program that runs on Raspberry Pi was used to send frames to the neural network, which was also stored in the Pi.

9.4.1.2 Processing Unit (Raspberry Pi)

A computer's processing unit manages numerous chores: inputting the data, training the neural network, projecting the data via commands, and sending the resultant commands to the remote control for the final movement, as shown in Figure 9.1.

9.4.1.3 Unit of Autonomous Vehicle Control

The controller of the car used contained a switch for turning it on and off. The Raspberry Pi was used to simulate button-press actions by soldering the jumper cables to the remote and then connecting them to four general-purpose input/output (GPIO) pins that were selected to collect the pin chips of the remote, communicating actions like turn left, move ahead, turn right, and move back. When the GPIO pins emitted a low signal, it meant that the pins had been given a supply from the ground, whereas a high signal denoted that chip resistance had no effect on the ground [54]. The model returned the resultant commands to the Pi using the serial interface, and the Pi then output the signal as high or low after reading the input command, mimicking the act of driving the autonomous vehicle by pressing buttons.

9.4.2 Simulation

Before testing the hardware on the actual field, the working of the model and the systems in place had to be tested. To achieve this, a video was fed to the training model frame by frame that corresponded to the directions generated by the CNN.

After obtaining these results, it was apparent that the basic functionality had reached the point of acceptable accuracy, as confirmed by the accuracy graphs shown in Figures 9.7 and 9.8. However, due to the hardware restrictions—i.e., having only left, right, forward, and backward controls—no concrete conclusions could be drawn from this effort, and field testing was the only way to be able to understand the working and the shortcomings in detail.

9.4.3 Results Analysis

After rigorous test runs on tracks different from those on which the car was trained, it was concluded that the car was able to function competently in a controlled environment. The neural network was also working correctly and gave better results than those obtained by previous studies, as shown in Table 9.1. The only complications faced pertained to the type of car used, which it was later determined could be comfortably resolved.

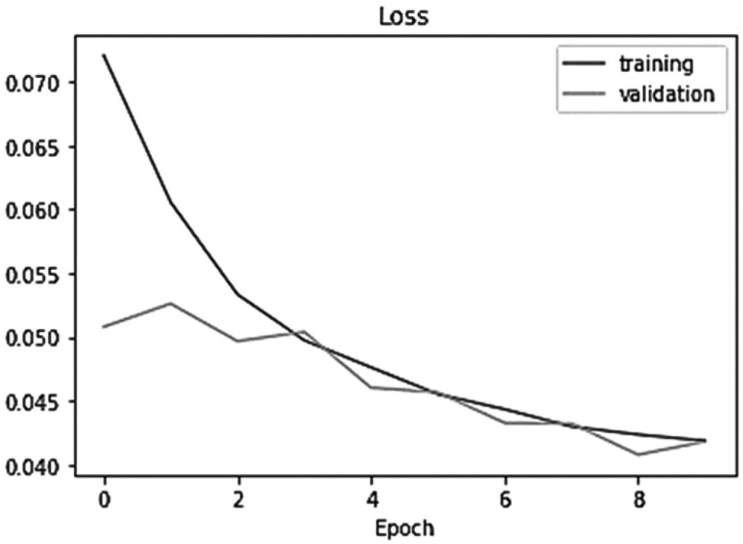

FIGURE 9.7 The result given by the CNN for every frame.

After testing and training the data, a data set of images was obtained from the first trial of our self-driving vehicle. The data set consisted of videos and images with similar structure to the tracks we used in driving vehicle. The test data was used to evaluate the accuracy of the model. A new curvature of the road (on a sheet of black paper) was introduced for each test case. After several tests, we noticed that the accuracy increased and that with every tryout, the car started to make more precise turns. The car drove effortlessly and dodged the obstructions in the lane with apt precision.

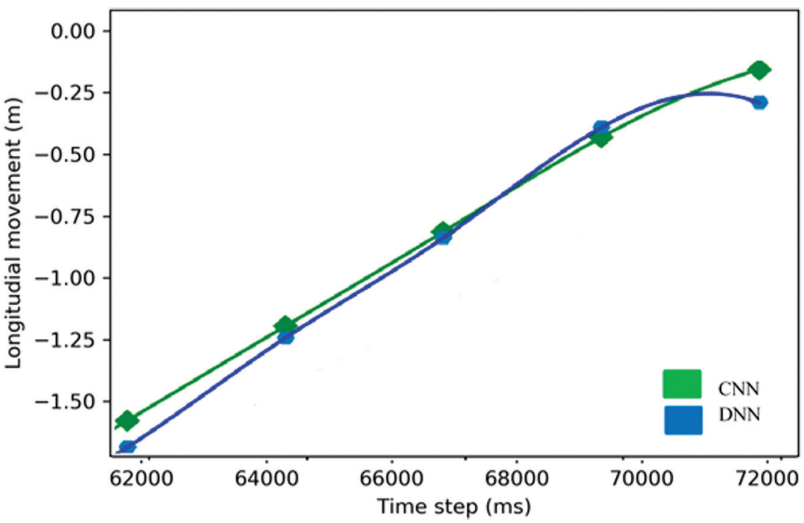

FIGURE 9.8 Accuracy comparison between CNN and DNN.

TABLE 9.1 A Comparison of the Accuracy of the Current Study with That of Others

	Title of Paper	Accuracy
Muller et al. (2006) [55]	Off-Road Obstacle Avoidance through End-to-End Learning	75%
Mori et al. (2019) [56]	Visual Explanation by Attention Branch Network for End-to-End Learning-Based Self-Driving	79%
Xu et al. (2017) [57]	End-to-End Learning of Driving Models from Large- Scale Video Datasets	84.6%
Bojarski et al. (2016) [6]	End to End Learning for Self-Driving Cars	88%
Ours	Automated Guided Autonomous Car Using Deep Learning and Computer Vision	98%

9.5 CONCLUSION

The learning approach used here in building a self-driving vehicle from beginning to end is an effective alternative to the traditional one. The car learned driving mannerisms and could detect the road without the need for explicit labels, and it gave viable results in a limited time frame, making the approach also very cost-effective [58]. The project thus developed should be viewed as independent of the vehicle, like a compressed natural gas kit, as the vision was to develop a setup that could be incorporated into any vehicle, given that its acceleration and direction could be controlled. More work needs to be done on how to make the hardware implement the product of the neural network more precisely.

Some researchers think that the future of vehicles is electric energy-saving vehicles. Today we rely heavily on fossil fuels to run our vehicles, but because these fuels are nonrenewable, they should be used judiciously. By consuming fossil fuels, we are not only exhausting our resources but also adding to air pollution. With electric vehicles, we may think we will have a one less problem to worry about, but believing that vehicles that run on electricity will not cause any harm is incorrect. An automobile running on electric current produces particulate discharge from use of the breaks and wear and tear on the tires, and this emission is also hazardous.

Analysts indicate that it will be about 25 years before autonomous vehicles make up half the automobile market and probably 40 years before there is an autonomous cars. In the coming years, we might see drivers getting rid of vehicles that are not smart and self-driving. It is obvious that at first, autonomous vehicles will be very costly, and only the affluent will have the luxury of using them. There are numerous upsides of self-driving vehicles, but first and foremost, with less reliance on drivers, there will be fewer to no chances of accidents occurring, as the algorithm will have peak precision. Also, this will bring a sense of relief for nondrivers. Nevertheless, unless autonomous vehicles become a common thing, they will not be affordable to moderate-income people/organizations. For these vehicles to be feasible, they need to be flooding the auto market.

REFERENCES

1. Pan, Y., C. A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots. 2017. Agile Autonomous Driving Using End-to-End Deep Imitation Learning. arXiv preprint, arXiv:1709.07174.
2. Desai, S., and S. Desai. 2017. Smart vehicle automation. *International Journal of Computer Science and Mobile Computing*, *6*(9), pp. 46–50.
3. Hussain, R., and S. Zeadally. 2018. Autonomous cars: Research results, issues, and future challenges. *IEEE Communications Surveys & Tutorials*, *21*(2), pp. 1275–1313.
4. Lecun, Y., E. Cosatto, J. Ben, U. Muller, and B. Flepp. 2004. DAVE: Autonomous Off-Road Vehicle Control Using End-to-End Learning. Final Technical Report, Net-Scale Technologies http://www.cs.nyu.edu/yann/research/dave/dave-final-report.pdf.
5. Jackel, L. D., E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan. 2006. The DARPA LAGR program: Goals, challenges, methodology, and phase I results. *Journal of Field Robotics*, *23*(11–12), pp. 945–973.
6. Bojarski, M., D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, and X. Zhang. 2016. End to End Learning for Self-Driving Cars. arXiv preprint, arXiv:1604.07316.
7. Gurghian, A., T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali. 2016. DeepLanes: End-to-end lane position estimation using deep neural networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 38–45.
8. Huang J., et al. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 3296–3297.
9. Billones, R. K. C., A. A. Bandala, E. Sybingco, L. A. G. Lim, A. D. Fillone, and E. P. Dadios. 2017. Vehicle detection and tracking using corner feature points and artificial neural networks for a vision-based contactless apprehension system. 2017 Computing Conference, London, pp. 688–691.
10. Pomerleau, D. A. 1988. ALVINN: An autonomous land vehicle in a neural network. Advances in Neural Information Processing Systems 1 (NIPS 1988), pp. 305–313.
11. Rowley, H. A., S. Baluja, and T. Kanade. 1998. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(1), pp. 23–38.
12. Sharifara, A., M. S. Mohd Rahim, and Y. Anisi. 2014. A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection. 2014 International Symposium on Biometrics and Security Technologies (ISBAST), pp. 73–78.
13. Baykal, S. I., D. Bulut, and O. K. Sahingoz. 2018. Comparing deep learning performance on BigData by using CPUs and GPUs. 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), Istanbul.
14. Mo, Y. J., J. Kim, J. K. Kim, A. Mohaisen, and W. Lee. 2017. Performance of deep learning computation with TensorFlow software library in GPU-capable multi-core computing platforms. 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 240–242.
15. Gong, T., T. Fan, J. Guo, and Z. Cai. 2017. GPU-based parallel optimization of immune convolutional neural network and embedded system. *Engineering Applications of Artificial Intelligence*, *62*, pp. 384–395.
16. Ertugrul, E., U. Kocaman, and O. K. Sahingoz. 2018. Autonomous aerial navigation and mapping for security of smart buildings. 2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG), Istanbul, pp. 168–172.
17. Turker, T., O. K. Sahingoz, and G. Yilmaz. 2015. 2D path planning for UAVs in RADAR threatening environment using simulated annealing algorithm. 2015 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 56–61.

18. Bresson, G., Z. Alsayed, L. Yu, and S. Glaser. 2017. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, *2*(3), pp. 194–220.
19. Choudhury, S., S. P. Chattopadhyay, and T. K. Hazra. 2017. Vehicle detection and counting using Haar feature-based classifier. 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), Bangkok, pp. 106–109.
20. Tefft, B. C. 2016. The prevalence of motor vehicle crashes involving road debris, United States, 2011–2014. *AAA Foundation for Traffic Safety*, (24), pp. 1–10.
21. Bertozzi, M., A. Broggi, and A. Fascioli. 2000. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, *32*(1), pp. 1–16. https://doi.org/10.1016/S0921-8890(99)00125-6.
22. Dagan, E., O. Mano, G. P. Stein, and A. Shashua. 2004. Forward collision warning with a single camera. *IEEE Intelligent Vehicles Symposium, 2004*, pp. 37–42. https://doi.org/10.1109/IVS.2004.1336352.
23. Tatarek, T., J. Kronenberger, and U. Handmann. 2018. Functionality, advantages and limits of the Tesla Autopilot. https://www.semanticscholar.org/paper/Functionality-%2C-Advantages-and-Limits-of-the-Tesla-Tatarek-Kronenberger/3f6d7ec8f532aeabb8a0cb0c88201266483980 93
24. Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (NIPS 2012), pp. 1–9. https://doi.org/http://dx.doi.org/10.1016/j.protcy.2014.09.007.
25. Yang, Z., Y. Zhang, J. Yu, J. Cai, and J. Luo. 2018. End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perception. arXiv:1801.06734.
26. Bhavsar, P., P. Das, M. Paugh, K. Dey, and M. Chowdhury. 2017. Risk analysis of autonomous vehicles in mixed traffic streams. *Transportation Research Record: Journal of the Transportation Research Board*, *2625*(1), pp. 51–61. https://doi.org/10.3141/2625-06.
27. Yao, J., S. Fidler, and R. Urtasun. 2012. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 702–709. https://doi.org/10.1109/CVPR.2012.6247739.
28. Zhang, M., Y. Zhang, L. Zhang, C. Liu, and S. Khurshid. 2018. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp. 132–142. https://doi.org/10.1145/3238147.3238187.
29. Sak, H., A. Senior, and F. Beaufays. 2014, Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 15th Annual Conference of the International Speech Communication, pp. 338–342. arXiv:1402.1128.
30. Udacity, Inc. n.d. Self driving car. https://github.com/udacity/self-driving-car.
31. Takahashi, N., Shibata, N., and Nonaka, K. 2020. Optimal configuration control of planar leg/wheel mobile robots for flexible obstacle avoidance. *Control Engineering Practice*, *101*, p. 104503.
32. Carlini, N., and D. Wagner. 2017. Towards evaluating the robustness of neural networks. Proceedings—IEEE Symposium on Security and Privacy, pp. 39–57. https://doi.org/10.1109/SP.2017.49.
33. Papernot, N., P. McDaniel, X. Wu, S. Jha, and A. Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. Proceedings—2016 IEEE Symposium on Security and Privacy (SP 2016), pp. 582–597. https://doi.org/10.1109/SP.2016.41.
34. Athalye, A., N. Carlini, and D. Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv:1802.00420v4.

35. Papernot, N., P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. 2016. The limitations of deep learning in adversarial settings. Proceedings—2016 IEEE European Symposium on Security and Privacy (EURO S and P 2016), pp. 372–387. https://doi.org/10.1109/EuroSP.2016.36.
36. Eykholt, K., I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. 2018. Robust physical-world attacks on deep learning models. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1625–1634 https://doi.org/10.1109/CVPR.2018.00175.
37. Geiger, A., P. Lenz, C. Stiller, and R. Urtasun. 2013. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, *32*(11), pp. 1231–1237. https://doi.org/10.1177/0278364913491297.
38. Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. 2016. The Cityscapes dataset for semantic urban scene understanding. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3213–3223. https://doi.org/10.1109/CVPR.2016.350.
39. Guo, Y., Y. Liu, T. Georgiou, and M. S. Lew. 2018. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, *7*(2), pp. 87–93. https://doi.org/10.1007/s13735-017-0141-z.
40. Caesar, H., J. Uijlings, and V. Ferrari. 2016. Region-based semantic segmentation with end-to-end training. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9905, pp. 381–397. https://doi.org/10.1007/978-3-319-46448-0_23.
41. Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 580–587. https://doi.org/10.1109/CVPR.2014.81.
42. Long, J., E. Shelhamer, and T. Darrell. 2015. Fully convolutional networks for semantic segmentation. Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965.
43. Eigen, D., and R. Fergus. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2650–2658. https://doi.org/10.1109/ICCV.2015.304.
44. Liu, Y., Y. Guo, and M. S. Lew. 2017. On the exploration of convolutional fusion networks for visual recognition. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10132, no. 1, pp. 277–289. https://doi.org/10.1007/978-3-319-51811-4_23.
45. Dai, J., K. He, and J. Sun. 2015. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. Proceedings of the 2015 IEEE International Conference on Computer Vision, pp. 1635–1643. https://doi.org/10.1109/ICCV.2015.191.
46. Papandreou, G., L.-C. Chen, K. P. Murphy, and A. L. Yuille. 2015. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1742–1750. https://doi.org/10.1109/ICCV.2015.203.
47. Khoreva, A., R. Benenson, J. Hosang, M. Hein, and B. Schiele. 2017. Simple does it: Weakly supervised instance and semantic segmentation. Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), pp. 1665–1674. https://doi.org/10.1109/CVPR.2017.181.
48. Udacity, Inc. n.d. Self-Driving Car Simulator. https://github.com/udacity/self-driving-car-sim. Accessed on 5 Feb. 2019.

49. Eraqi, H. M., M. N. Moustafa, and J. Honer. 2017. End-to-end deep learning for steering autonomous vehicles considering temporal dependencies. 31st Conference on Neural Information Processing Systems (NIPS), pp. 1–8. arXiv:1710.03804.
50. Teichmann, M., M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. 2018. MultiNet: Real-time joint semantic reasoning for autonomous driving. 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1013–1020. https://doi.org/10.1109/IVS.2018.8500504.
51. Siam, M., S. Elkerdawy, M. Jagersand, and S. Yogamani. 2018. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. Proceedings of the 2018 IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1–8. https://doi.org/10.1109/ITSC.2017.8317714.
52. Podpora, M., G. P. Korbas, and A. Kawala-Janik. 2014. YUV vs RGB: Choosing a color space for human-machine interaction. *FedCSIS Position Papers*, *18*, pp. 29–34.
53. Zhang, Jiakai. 2019. *End-to-end learning for autonomous driving.* Dissertation, New York University.
54. Wang, Zheng. 2018. Self driving RC car. https://zhengludwig.wordpress.com/projects/self-driving-rc-car/.
55. Muller, U., Ben, J., Cosatto, E., Flepp, B., and Cun, Y. L. 2006. Off-road obstacle avoidance through end-to-end learning. In Advances in neural information processing systems, pp. 739–746.
56. Mori, Keisuke, et al. 2019. Visual explanation by attention branch network for end-to-end learning-based self-driving. 2019 IEEE intelligent vehicles symposium (IV). IEEE.
57. Xu, H., Gao, Y., Yu, F., and Darrell, T. 2017. End-to-end learning of driving models from large-scale video datasets. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2174–2182).
58. Rajasekhar, M. V., and A. K. Jaswal. 2015. Autonomous vehicles: The future of automobiles. 2015 IEEE International Transportation Electrification Conference (ITEC), pp. 1–6.