sciendo

Transport and Telecommunication, 2021, volume 22, no. 4, 383–391
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2021-0029

OBJECT AND LANE DETECTION TECHNIQUE FOR AUTONOMOUS CAR USING MACHINE LEARNING APPROACH

Raja Muthalagu[1], Anudeep Sekhar Bolimera[2], Dhruv Duseja[3], Shaun Fernandes[4]

[1,3,4]Department of Computer Science, Birla Institute of Technology and Science Pilani
Dubai Campus, Dubai, UAE
[2]Electrical and Computer Engineering, Carnegie Mellon University
Pittsburgh, PA, USA
[1]raja.m@dubai.bits-pilani.ac.in, [2]abolimer@andrew.cmu.edu, [3]f20160904@dubai.bits-pilani.ac.in and [4]f20160290@dubai.bits-pilani.ac.in

The main objective of this work is to develop a perception algorithm for self-driving cars which is based on pure vision data or camera data. The work is divided into two major parts. In part one of the work, we develop a powerful and robust lane detection algorithm which can determine the safely drive-able region in front of the car. In part two we develop and end to end driving model based on CNNs to learn from the drivers driving data and can drive the car with only the camera data from on-board cameras. Performance of the proposed system is observed by the implementation of the autonomous car that can be able to detect and classify the stop signs and other vehicles.

Keywords: Image Processing, Machine Learning, Autonomous Cars, Self-Driving, Object Detection.

1. Introduction

In the present-day world, we spend a significant amount of time in our vehicles or interacting with other vehicles as a pedestrian on the roads. Each year, 1.35 million lives are killed on roadways around the world and almost 3,700 people are killed around the world per day in crashes involving Buses, trucks, car, motorcycles, bicycles, or pedestrians (Singh, 2015). The number of automobiles on the road around the world is also increasing rapidly (Bellis *et al*., 2008). Researchers shows that the human error is a main cause for the greatest number of road accidents.

The Advanced Driver Assistance Systems (ADAS) is developed in recent years for the purpose of improving the safety and comfort of the passengers (Lu *et al*., 2005; Bengler *et al*., 2014; Liyong *et al*., 2020) which has motivated to develop a fully autonomous self-driving car. The self-driving car technology can be broken down into three major building blocks namely perception, planning and control. A considerable amount of work has been done by researchers to develop a various technique to improving the driving safety and reducing traffic accidents. Most of the ADAS systems are using variety of sensors for lane and object detection on the highway. Various camera vision approaches are proposed to detect the lane and object (Gayko *et al*., 2012; Simon *et al*., 2017).

This kind of driver assistance systems are expected to be more perfect under any traffic conditions (Eskandarian *et al.*, 2012). However, predicting the road traffic is very difficult due the dynamic nature of the road traffic environment and that is a bottleneck for the development of this kind of driver assistance systems (Eskandarian *et al.*, 2012). In a complex traffic situation like heavy traffic, a greater number of high-speed vehicles, the possibility of accident is much greater that usual. Under these traffic conditions, road boundary, object detection, lane marking, road texture detection and colour extraction are the main perceptual clues of human driving.

The vision-based lane and object detection is proposed by many researchers around the (Xing *et al*., 2018; Kunz *et al*., 2017; Yan *et al*., 2017). The image features like intensity difference between the lane marking and road information is used to detect lane (Borkar *et al*., 2012; Audibert *et al*., 2010). The straight lanes are detected using Hough transform and edge information (Raja *et al*., 2020; He *et al*., 2012). To improve the speed of the detection, the modified Hough transform based lane detection also proposed (Kuk *et al*., 2010). The mathematical model of the lane marking is developed with the help of obtained road information that is called as B-spline (XU *et al*., 2009). For object detection, the deep convolutional networks (CNNs) is used to produce very good results. CNNs has been used for classification by using the image feature learning capability of the CNNs (Zhao *et al*., 2018). A multi-stage CNN architecture proposed that can perform both the semantic segmentation for lane markings and also the localization of each lane in the form of key-points (Muthalagu *et al*., 2021). A system for image processing in autonomous cars has been implemented (Shaun *et al*., 2021).

In this paper, we propose a two different lane detection and object detection method that is suitable for all kinds of complex traffic situations, especially as driving speed in roads is too fast. The images taken from the model car is processed, and the processed data can be used to control the autonomous vehicle. First one is a minimalistic lane detection approach which can detect only the straight lane lines. Second one is to develop a CNNs based model which can learn to drive a car from the drivers driving data. This approach can drive the car that is learned by cloning the behaviour of a human driver. The shortcomings of the minimalistic approached were overcome with the advance lane finding approach which does an excellent job at detecting the lanes and is robust and less susceptible to weather changes or shadows and different road conditions. The simple Remote-Controlled car is equipped with the cameras in the front that record the video as well the detecting the object. In this paper, our contributions are summarized as follows.

1. A minimalistic lane detection approach is proposed to detect the straight lane lines.
2. A CNNs based model is developed which can learn to drive a car from the drivers driving data. This approach can drive the car that is learned by cloning the behaviour of a human driver.
3. The simple remote-controlled car is equipped with the cameras in the front that record the video and it uses the proposed late detection approach to detect the lane and object.

The remainder of the paper is organized as follows. Section 2 describes minimalistic approach-based lane-detection. Section 3 describes the machine learning approach-based lane detection. Section 4 describes the hardware implementation of remote-controlled car. Experimental results of the object detection and lane detection are presented in Section 5. Finally, Section 6 concludes our work.

2. Minimalistic Approach-based Lane-detection

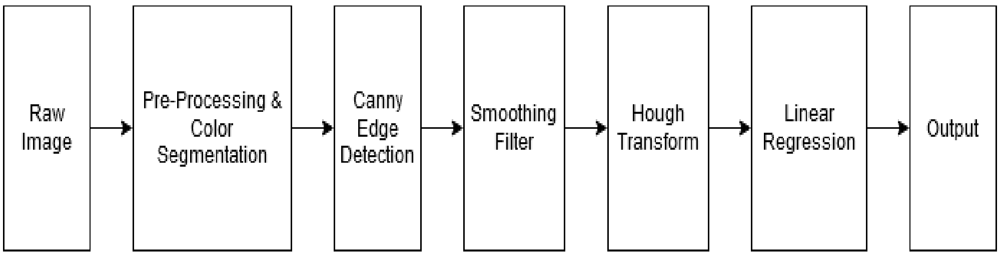

Figure 1. Minimalistic approach-based lane detection

The proposed minimalistic approach for lane detection is shown in Figure 1 is used to keeping the vehicle in the middle of the lane al all the time. In this work, the frames read from the video are given as an input to the system and the raw images are go through the following processing steps. Pre-processing and color segmentation on the image that includes loading the raw image, converted the raw image from its original RGB color space to HSL (Hue Saturation Lightness) and split into its individual color bands of H, L and S. It is used to detect objects or parts of the image that are in a specific color. At the end this process, the algorithm processes only part of the image that are in specified colors. This algorithm is very much useful for extracting information about lane markings. Figure 2 depicts the input to and output from the pre-processing and color segmentation block which clearly extract the yellow and white line.

In second stage, the Canny edge detector (Yan *et al*., 2017) is used to detect sharp intensity changes and to find lane boundaries in a color segmented image. The Canny edge detector classifies a pixel as an edge if the gradient magnitude of the pixel is larger than those of pixels at both its sides in the direction of maximum intensity change, Figure 3 shows the result after applied the ‘Canny’ edge detector over the gray scaled image. As shown in Figure 3 the Canny edge detection algorithm yields good edge detection results. The detection of the edges for the lane markings may be heavily affected by present noise in the image, so it is required to remove the noise after the edge detection by filtering. A smoothing filter is used to smoothen the gradient, and only the prevailing gradients are detected in this edge detection algorithm. Figure 4 shows the noise removed edge detected image. After the edge detection, the lines can be extracted from these edges. The Hough transform (Rui *et al*., 2013) is used in the third stage to extract the lanes from the detected edge. The Hough transform is used to convert the lines into a parameter space as a point, it helps to detect the lines by associate a line in the image as a single point in the parameter space. The slope for the lanes is determined with linear regression after the most dominant lines are found. But this approach can detect only the straight line not capable of detecting the curved line. Figure 5 shows the line extracted output by Hough transform.

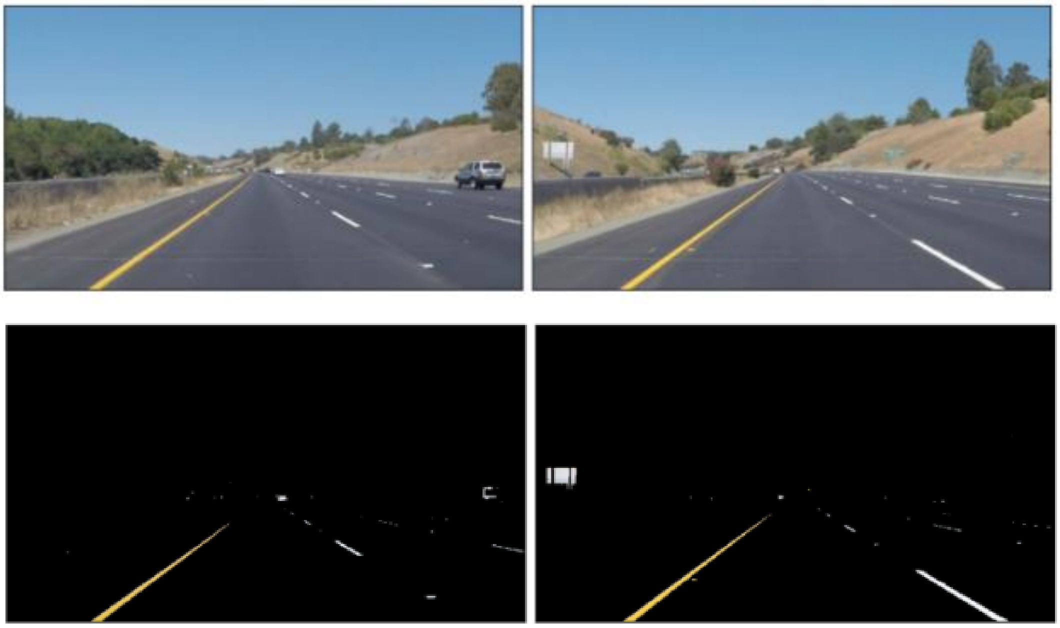

Figure 2. Input raw image and output color segmented image from pre-processing and color segmentation block

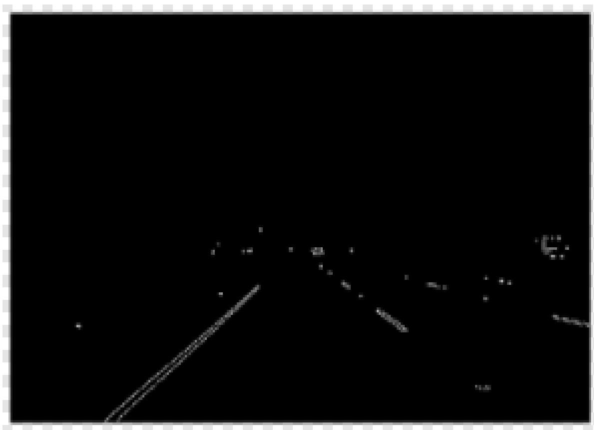

Figure 3. Edge detected output with noise from the Canny edge detector

Figure 4. Edge detected output from the Canny edge detector after noise removal

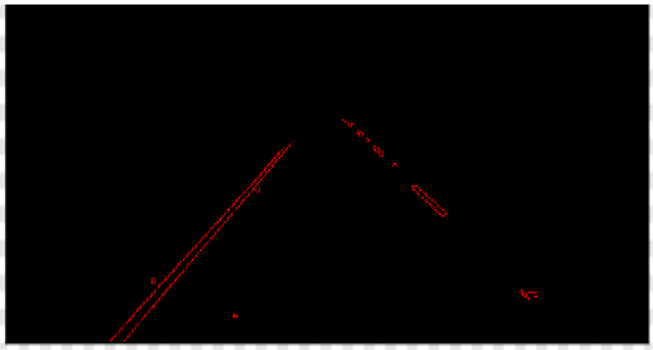

Figure 5. Line extracted output by Hough transform

Detection of false positive and removal of outlier is another important process that need to be employed in the proposed approach. The removal of outliers is done by employing more filtering based on their slopes and created two groups based on the left and right lane lines. In last stage the dominant lines are identified based on the maximum probability of being part of the road marking. The length and slope of the line are the two important metric those are used to identify the dominant lines. Due to the fact that lane always appear as a straight line from the vehicle, the length of the lane is used to identify the detected line is belonging to the road or not. Similarly, the slope of the line is used to filter the outlier. Linear regression is used to determine the slope of the line. The proposed minimalistic approach is detecting the straight line properly that can be verified from the result as shown in Figure 6. But only drawback is that it fails to detect the curved lane.

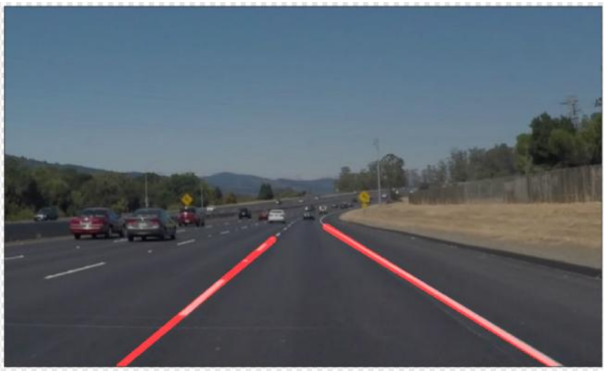

Figure 6. The result of the minimalistic lane detection approach

3. The Machine Learning Approach based Lane Detection

3.1. Behavioral-Cloning Using CNNs for Self Driving Cars

In the modern era of Deep learning, self-driving cars have become a reality. With most players like Google, Tesla, Mercedes, Ford etc. trying to build robust self-driving schemes which can perform in the unstructured environment, the focus has shifted a lot to the use of deep learning models. In particular, the convolutional neural networks are used commonly, to process raw RGB data and its ability to segment the scene and detecting various objects in the scene. CNNs also have the ability to generalize image recognition which leads to its ability to be used in a variety of driving environments. The main aim of this section is to develop a CNN based model which can learn to drive a car from the drivers driving data.

3.2. Model Architecture

The overall strategy for deriving model architecture was to drive the car in the center of the lane smoothly. The first step was to start with the most basic neural network and to continue with the addition of new layer and gauge the change in performance with the new addition. This allowed a better understanding of visualizing what each layer in a CNN does as in Table1.

1. The first layer was the normalization layer in order to mean center the training data.
2. The next part of observation was that the upper half of the frame was mostly trees and mountains which harm the networks performance more than improving it. Hence a cropping layer has been added to crop that part off.

3. In order to gauge how well the model was working, the image and steering angle data was split into a training and validation set.
4. The initial conclusion was that the first model had a low mean squared error on the training set but a high mean squared error on the validation set. This implied that the model was overfitting.
5. Next step was the addition of the convolution layer and fully connected layers. There was an increase in performance however the car still steered off the course majority of the times.
6. After that, we tried the Nvidia Autonomous Car Group (Bojarski *et al*., 2016) model, and the car drove the complete first track after just 5 training epochs.
7. To combat overfitting, initially the use of Dropout was made. However, the performance was much worse. So, to combat the overfitting, the number of training epochs was reduced. There was no major change in the neural network model.

3.3. Dataset

The data-set included 21,500 images of resolution which was utilized for the process of training (*Open Source Autonomous Driving Dataset- Udacity, 2017*). The steering angle data was also captured for each of these 21,500 images. The training set and the cross-validation data is split into the 80-20 % ratio respectively. The Adam's algorithm was used for optimizing the cost function which is the Mean squared error (MSE) in this case.

Table 1. Summary of the CNN architecture used

Sr no.	Layer	Properties
1	Lambda	Input: (160,320,3)
2	Cropping2D	Cropping: ((50,20),(0,0))
3	Convolution 0	Convolution2D
		Kernel: (5,5)
		Activation: Relu
		Stride: (2,2)
		Filters: (24)
4	Convolution 1	Convolution2D
		Kernel: (5,5)
		Activation: Relu
		Stride: (2,2)
		Filters: (36)
5	Convolution2	Convolution2D
		Kernel: (5,5)
		Activation: Relu
		Stride: (2,2)
		Filters: (48)
6	Convolution3	Convolution2D
		Kernel: (3,3)
		Activation: Relu
		Stride: (1,1)
		Filters: (64)
7	Convolution4	Convolution2D
		Kernel: (3,3)
		Activation: Relu
		Stride: (1,1)
		Filters: (64)
8	Flatten	
	Fully Connected 0	Dense
		Input: 8448
		Output: 100
		Activation: Linear
9	Fully Connected 1	Dense
		Input: 100
		Output: 50
		Activation: Linear
10	Fully Connected 2	Dense
		Input: 50
		Output: 10
		Activation: Linear
11	Fully Connected 3	Dense
		Input: 10
		Output: 1
		Activation: Linear

4. Hardware Implementation of Remote-Controlled Car

The proposed minimalistic approach is implemented in a remote-controlled car that is operated by Arduino as shown in Figure 7. The camera fixed at the front of the car is used to capture a live feed of images. The captured images are transmitted to the computer by using Bluetooth module. The Arduino give instruction for the movement based on the processed image by the computer as shown in figure.

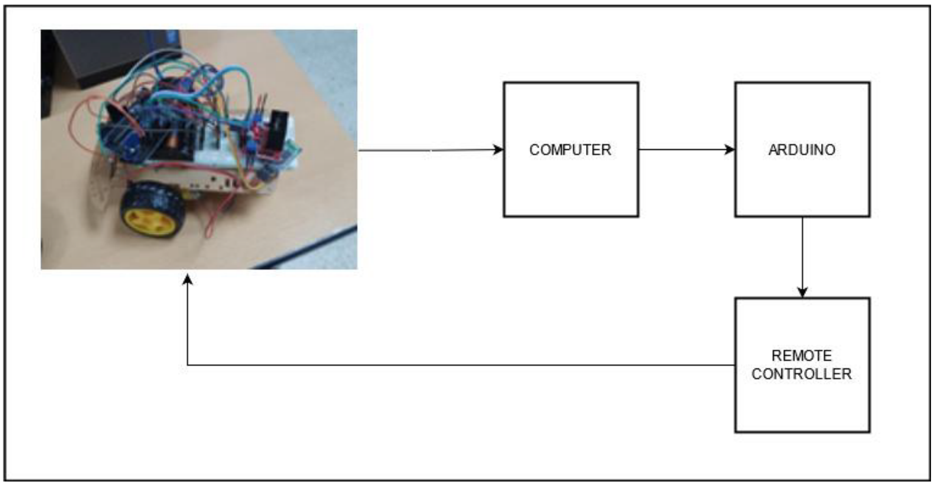

Figure 7. Remote-Controlled Car operated by Arduino

The Arduino was programmed to responding to certain input from the computer and performing certain task as response to the comment.

5. Results and discussion

In this section, we demonstrate the results of a CNN based model which can learn to drive a car from the drivers driving data using the Udacity's driving simulator (*Open Source Autonomous Driving Dataset- Udacity*, 2017). The dataset included 21,500 images of resolution which was utilized for the process of training. The steering angle data was also captured for each of these 21,500 images. The mean square error (MSE) is used as a loss function. From the result it may be observed that validation loss that is lower than the training loss as seen in Figure 8 and it indicates that the validation dataset may be easier for the model to predict than the training dataset. Figure 9 shows the car is centered on the road by using CNN based model.

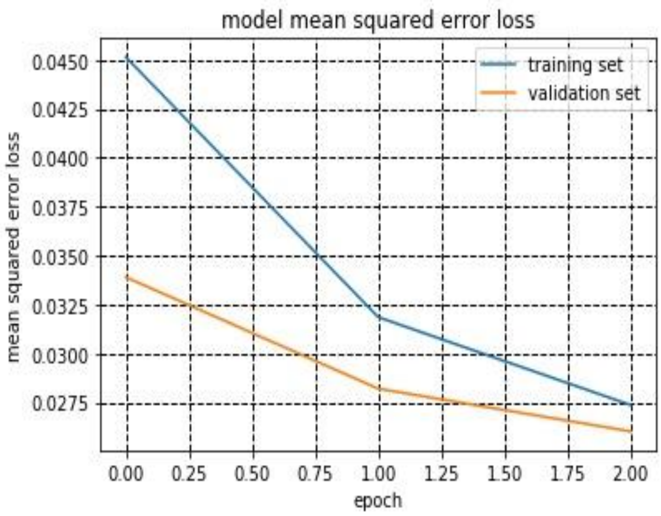

Figure 8. Change in training and validation error over the epochs

Figure 9. Car centered on the road

Similar to the late detection and identifying the street sign is also very much important feature for autonomous cars. As we know that street signs are provide useful information the driver, it can also provide very useful information to the driverless cars. As one can see from Figure 10, the stop sign is detected by our proposed minimalistic approach also it displays the distance between the car and stop sign. In addition to detect the stop sign, the algorithm was also trained to recognize vehicles and distance to the vehicle as shown in Figure 11. The distance 'd' between car camera and stop sign is calculated as follows.

$$d = \frac{f_l\, w}{W_p}, \qquad (1)$$

where f_l is focal length, w is width of the object and W_p is width in pixel.

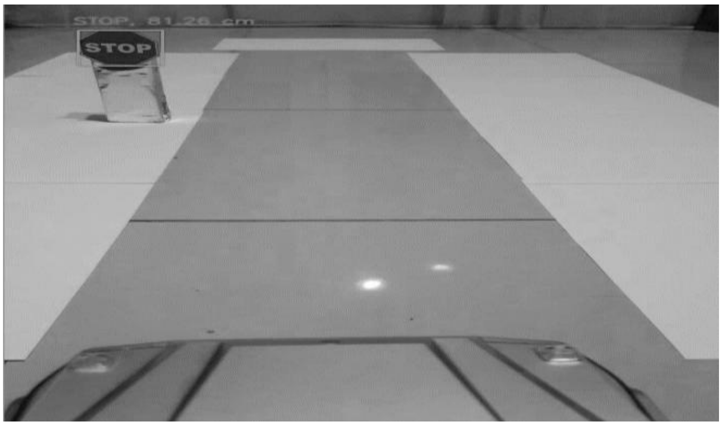

Figure 10. Output of the detection and distance measurement of road sign

Figure 11. Output of the detection and distance measurement of another vehicle

Table 2. Lane detection performance on the TUsimple (*tusimple benchmark ground truth*, 2018) test set

Method	*Accuracy*	*FP*	*FN*
PolyLaneNet (*Tabelini et al*., 2020)	93.36%	0.0942	0.0933
SCNN (*Pan et al*., 2018)	**96.53%**	**0.0617**	0.0180
Proposed CNNs approach	93.12%	0.0745	**0.0169**

The results and comparisons of our experiments in terms of accuracy, false positive and false negative are given in Table 2.

6. Conclusion & Future Works

The proposed minimalistic lane detection algorithm was successful in detecting lanes. The output images clearly shows that the lane lines are detected properly, and lines are very smoothly handled. Though it only detects the straight lane lines. It also helps to detect the road sign or another vehicle and measure the distance between road sign and the vehicle. Finally, we developed an end-to-end CNN based driving model which was able to drive the car based on the driver data with the accuracy of 93.23% and processing time of 26.83 ms. The shortcomings of the minimalistic approached were overcome with the advance lane finding approach which does an excellent job at detecting the lanes and is robust and less susceptible to weather changes or shadows and different road conditions. Our future work focuses on improving the lane proposal network's prediction accuracy using a generative adversarial training framework.

References

1. Audibert, H.K.Y. & Ponce, J. (2010) General road detection from a single image, *IEEE Trans. Image Process*. 19(8), 2211–2220, https://doi.org/10.1109/TIP.2010.2045715.
2. Bellis, Elizabeth, & Jim. (2008) *National motor vehicle crash causation survey (NMVCCS)*. (SAS analytical user's manual. No. HS-811 053.)
3. Bengler, K., Dietmayer, K., Farber, B., Maurer, M., Stiller, C., & Winner. H. (2014) Three Decades of Driver Assistance Systems: Review and Future Perspectives, *IEEE Intelligent Transportation Systems Magazine*, 6(4), 6–22. doi: 10.1109/MITS.2014.2336271.
4. Bojarski, M., Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016) End to End Learning for Self-Driving Cars. ArXiv, abs/1604.07316.
5. Borkar, A., Hayes, M., & Smith, M. (2012) A novel lane detection system with efficient ground truth generation, *IEEE Trans. Intell. Transp. Syst.* 13 (1), 365–374. https://doi.org/10.1109/TITS.2011.2173196.
6. Eskandarian A. (2012) Fundamentals of Driver Assistance. In: Eskandarian A. (eds) Handbook of Intelligent Vehicles. Springer, London. https://doi.org/10.1007/978-0-85729-085-4_19.
7. Gayko J.E. (2012). Lane Departure and Lane Keeping. In: Eskandarian A. (eds) Handbook of Intelligent Vehicles. Springer, London. https://doi.org/10.1007/978-0-85729-085-4_26
8. He Mao & Mei Xie. (2012) Lane detection based on though transform and endpoints classification, *International Conference on Wavelet Active Media Technology and Information Processing,* Chengdu, China, 125-127. doi:10.1109/ICWAMTIP.2012.
9. Kuk, G., An, H., Ki, H. & Cho, I. (2010) Fast lane detection and tracking based on hough transform with reduced memory requirement, *13th International IEEE Conference on Intelligent Transportation Systems.* Funchal, Portugal, 1344–1349, https://doi.org/10.1109/ITSC.2010.5625121
10. Kunz, N. Chase. (2017) *Vision-Based Control of a Full-Size Car by Lane Detection*. All Graduate Theses and Dissertations. 6534. Utah State University, United States.
11. Liyong Wang, Peng Sun, Min Xie, Shaobo Ma, Boxiong Li, Yuchen Shi & Qinghua Su. (2020). Advanced Driver-Assistance System (ADAS) for Intelligent Transportation Based on the Recognition of Traffic Cones, *Advances in Civil Engineering*, 2020 (8883639), 8 pages. DOI.org/10.1155/2020/8883639.
12. Lu, M., Wevers, K. & Heijden, R. V. D. (2005) Technical Feasibility of Advanced Driver Assistance Systems (ADAS) for Road Traffic Safety, *Transportation Planning and Technology*. 28(3), 167–187. https://doi.org/10.1080/03081060500120282.

13. Muthalagu, R., Bolimera, A. & Kalaichelvi, V. (2021) Vehicle lane markings segmentation and keypoint determination using deep convolutional neural networks, *Multimed Tools Appl.* 80, 11201–11215. https://doi.org/10.1007/s11042-020-10248-2.
14. *Open-Source Autonomous Driving Dataset*. (2017) https://github.com/udacity/self-driving-car-sim.
15. Pan, X., Shi, J., Luo, P., Wang, X. & Tang, X. (2018) Spatial as deep: Spatial CNN for traffic scene understanding, *arXiv preprint arXiv:1712.06080*
16. Raja, M., Anudeepsekhar, B. & Kalaichelvi, V. (2020) Lane detection technique based on perspective transformation and histogram analysis for self-driving cars. *Computers & Electrical Engineering*, 85, 1-16. https://doi.org/10.1016/j.compeleceng.2020.106653.
17. Shaun, Fernandes, Dhruv, Duseja, & Muthalagu, R. (2021) Application of Image Processing Techniques for Autonomous Cars, *Proceedings of Engineering and Technology Innovation*, 17, 01-12. https://doi.org/10.46604/peti.2021.6074.
18. Simon Sternlund, Johan Strandroth, Matteo Rizzi, Anders Lie & Claes Tingvall (2017) The effectiveness of lane departure warning systems – A reduction in real-world passenger car injury crashes, *Traffic Injury Prevention*, 18(2), 225-229. DOI: 10.1080/15389588.2016.1230672.
19. Singh, S. (2015) *Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey*. (Traffic Safety Facts Crash•Stats. Report No. DOT HS 812 115). Washington, DC: National Highway Traffic Safety Administration.
20. Tabelini, L., Berriel R., Paixão T.M., Badue C., Alberto. F De Souza. & Oliveira-Santos, T. (2020) PolyLaneNet: Lane estimation via deep polynomial regression, *arXiv preprint arXiv:2004.10924*
21. Talib, M.L., Rui X., Ghazali K.H., Mohd. Zainudin N. & Ramli S. (2013) Comparison of Edge Detection Technique for Lane Analysis by Improved Hough Transform, *Advances in Visual Informatics (IVIC 2013).* Selangor, Malaysia, November, 176-183. https://doi.org/10.1007/978-3-319-02958-0_17.
22. *Tusimple benchmark ground truth* (2018), https://github.com/TuSimple/tusimple-benchmark/issues/3.
23. Xing, Y. *et al*. (2018) Advances in Vision-Based Lane Detection: Algorithms, Integration, Assessment, and Perspectives on ACP-Based Parallel Vision, *IEEE/CAA Journal of Automatica Sinica*. 5(3), 645-661, doi: 10.1109/JAS.2018.7511063.
24. Xu, H., Wang, X., Huang, H., Wu, K. & Fang, Q. (2009) A fast and stable lane detection method based on b-spline curve, *IEEE 10th International Conference on Computer-Aided Industrial Design and Conceptual Design*, Wenzhou, China, 1036–1040. https://doi.org/10.1109/CAIDCD.2009.5375392.
25. Yan, Xuqin and Yanqiang, Li. (2017) *A method of lane edge detection based on Canny algorithm. 2017 Chinese Automation Congress (CAC),* Jinan, China, 2120-2124.
26. Zhao, Z.Q., Zheng, P., Xu, S.T. & Wu, X. (2018) Object detection with deep learning: A review, *arXiv e-prints, arXiv:1807.05511.*