Ameliorated End-to-End Deep Learning System for Self Learning Cars

R.Thirumahal[1], BalajiMuthazhaganT[2]
{ trk.cse@psgtech.ac.in[1], balajimuthazhagan@gmail.com[2]}

Department Of Computer Science & Engineering PSGCollegeof Technology, Coimbatore – 641004[1,2]

Abstract. The introduction of self-driving automobiles in today's society necessitates the development of high-quality algorithms to guide them. Convolutional neural networks are extensively utilised because of their ability to classify images based on observed attributes. The proposed solution entails steering a car autonomously using just the windshield view as input. This is accomplished by using Convolutional neural networks in an end-to-end deep learning strategy, as proposed by NVIDIA for self-driving automobiles. To improve the accuracy of existing models, the neural network will be supported with inputs from image processing modules that identify lane markers and cars.Reduced costs, increased safety, and increased mobility are all advantages of providing such an algorithm for retrofitting existing cars.

Keywords: Autonomous Vehicles, Self-driving car, End-to-End Deep learning, Convolutional Neural Networks,Feature Extraction,Classification Tools.

1 Introduction

This Autonomous automobiles, often known as self-driving cars, are vehicles that can steer and drive themselves. For more than a decade, academics and industrial R&D departments have been researching emerging autonomous driving solutions. In the not-too-distant future, Level-3/4 autonomous vehicles may become a reality. A combination of interlocking trends, including the renaissance of deep learning [17], [18], the rapid progression of sensing and in-vehicle computing devices, the accumulation of data with annotations, and technical breakthroughs in related research fields, are the primary reasons for drastic technical achievement in recent years. In research, the levels of automation pertaining to vehicle steering vary from Level 0(zero autonomous features)toLevel5(operate entirely without the presence of a driver) with the current multitude of cars operating with an autonomy of Level 1 (driver assistance is mandatory). Various automobile manufacturers and software giants have shown interest in increasing this autonomy for safer travel, lower operational costs, and increased mobility. The goal of this research is to improve an end-to-end deep learning system with assisting image processing modules in order to attain Level 3 autonomy. (automated driving with optional driver engagement). The image processing modules will focus on identifying lane markings and vehicles to suggest the applicable steering angle. Just like in the case of a human driver, the system will receive only the windshield view of the vehicle as input. The system will then predict the direction and a quantified measure of steering. The project is largely inspired by a paper published by NVIDIA towards autonomous steering

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314534

which prescribes a pure end-to-end deep learning approach. By implementing a better pipeline and having aiding modules on top of it will lead to better prediction of steering angles.

I. LITERATUREREVIEW

A.Autonomous Vehicles

Waymo, Uber, Tesla, Nissan, and other companies are devoting a significant amount of engineering resources to developing self-driving vehicles. Figure 1 depicts the similarities between autonomous cars and human-driven automobiles.

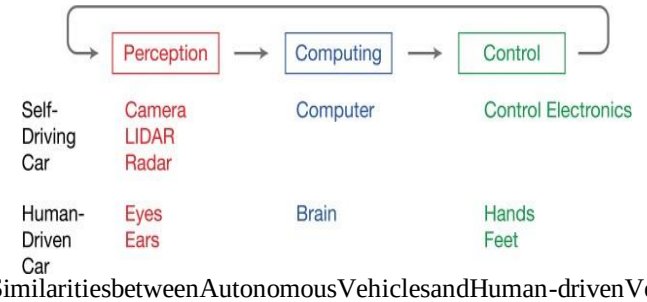
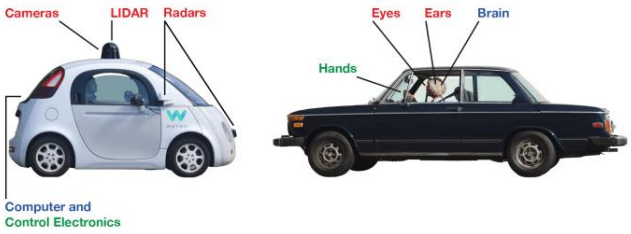


Fig.1.SimilaritiesbetweenAutonomousVehiclesandHuman-drivenVehicles.

The entire process of Vehicle steering can be broken upinto3parts[1]:

Perception – This refers to the vehicle's ability to receive visual and other important information.

The self-driving car's perception can be achieved through various methods. 360º vision cameras are good in identifying shapes and the corresponding colour of the environment but are not useful in determining the distance to the objects. LIDAR (Light Detection and Ranging) contains a constantly rotating beam which helps in judging the distance, location of objects and shapes. Radar is used in addition to 360º vision cameras and LIDAR for fine tuning.



Control – Deals with the mechanical component of driving the vehicle after the computing stage.

Computing – Deals with how the visual data must be processed in order to acquire meaningful information.

The steering wheel will receive input from the Computing stage in order to steer the vehicle. The three processes can be combined in order to emulate an autonomous vehicle.

B.Convolutional Neural Network Methods

The arduous nature of pattern recognition has been simplified using Convolutional Neural Networks. Before the use of CNNs, the entire process used to be split into two parts: Feature extraction and Classification [2].Through CNN, features can now be automatically learned through training examples.

1) ALVINN

The first approach to using a neural network for autonomous vehicle navigation was ALVINN (Autonomous Land Vehicle in a Neural Network). Pomerleau was the one who came up with it (1989). The model structure was simple, and the network predicted actions from pixel inputs. The predicted actions were then mapped on to typical driving circumstances. It showed that neural networks can be used to create an end-to-end navigation system [3].

2) DAVE

The Defense Advanced Research Projects Agency (DARPA) began an experiment in which a radio-controlled automobile was driven through several locations. The car was also steered with the help of a human being to record the steering angle. Two cameras placed in the front of the vehicle recordedthevideoinput.Thisalongwiththesteeringanglewaspopulatedtoformthetrainingdata [4].

3) DAVE-2 System

The DAVE-2 system was proposed by NVIDIA in 2016. The collection system for the training data is shown in Fig 1

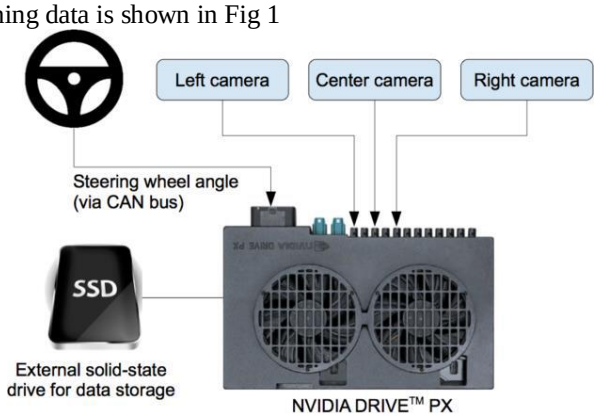


Fig 1. High-level view of the data collection system

Behind the windshield, three cameras are mounted, and time stamped video from the cameras is captured simultaneously with the steering angle applied by the human driver. 1/r, where r is the turning radius in meters, is used as the steering component to make the calculations independent of the vehicle's geometry [2] [13].

The training data contained images coupled with the respective steering component (1/r). Human driven data alone is not enough, the network must be able to recover from the mistakes. Additional images with different steering actions are recorded and are added to the

training data in order to make it more reliable [2]. By introducing a varied training data, the model trained will be less prone to errors.

C.Object Localization for Vehicle Detection

Though computer vision is expected to play a large role in self-driving cars, it has low precision, and hence it is coupled with road models, which uses multiple sensors for object detection. The aim is to get the detections using sensors in conjunction with a Kalman filter, and then filter out the false positives using road models. A neural network is used to make predictions and it is trained to predict the depth value of each object based on the labels. Further, a convolutional neural network (CNN) is used to extract the features. Initially, R-CNN used selective search for proposing regions and AlexNet to classify them. But it remained impractical for real-time implementations. Instead, Overfeat architecture has been used due to its high efficacy and ability to make a single forward pass by reusing the layers of the CNN efficiently [5].

While there has been a significant improvement in image recognition systems, the real-time part of it has not been considered. In [5] it has been applied to a laptop GPU which operates at a frequency greater than 10Hz and high image resolutions of 640 x 480. Overfeat CNN detector is used to make a single forward pass by reusing layers of the CNN. It converts an image recognition CNN into a sliding window, which converts the fully connected layers to convolutional layers. A grid of final feature vectors is produced, which is used to predict the presence of an object. After the object has been detected, a single bounding box is predicted using regression, using the same features. The network used in [5] has a context-view of 355 x 355 pixels. There is a problem of ambiguity where the prediction of two bounding box locations for two objects, is incorrectly handled since it predicts that there should be a third box in between [5].

D.End-to-end deeplearningsystem

The task is divided into lane detection [9][15], path planning [15][16], and control logic [7] in the primitive method. Lane keeping is a fundamental feature for self-driving cars. Despite many sensors installed on autonomous cars such as radar, LiDAR, ultrasonic sensor and infrared cameras, the ordinary color cameras are still very important for their low cost and ability to obtain rich information. Given an image captured by camera, one of the most important tasks for a self-driving car is to find the proper vehicle control input to maintain it in lane [6]. The lane markers can be recognised using various image processing techniques such as Hough transform, canny edge detection, and so on. Following that, path planning and control logic can be implemented. The quality of feature extraction and its interpretation of

visual data is crucial to the accuracy of such an approach. The rules obtained manually are not perfect and are prone to errors.

Only Convolutional neural networks are used in an end-to-end deep learning strategy to take in images as input and produce the control signal. There is no manual intervention in the model's optimization because it is solely reliant on the training data. This eliminates the need to identify and categorise pre-defined object categories [8].

Before training the CNN model, the data need to be further processed. First of all, to simplify the problem, driving at night is not considered in this paper and all four clips recorded at night are not considered. Second, the data contains many scenarios such as driving forward, changing lanes, making turns, driving on straight or curved roads, driving in normal speed or moving slowly in a traffic jam, etc. To train a lane keeping model, the data that meet the following criteria are selected: driving in normal speed, no lane changes or turns, and both straight and curved roads. End to end deep leaning largely reduced the human error which is involved.

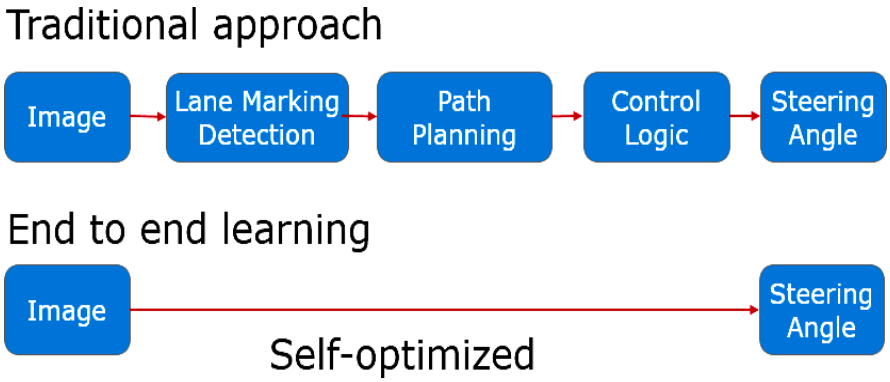


Fig.2.Traditional approach vs End to end deep learning.

2 System Design

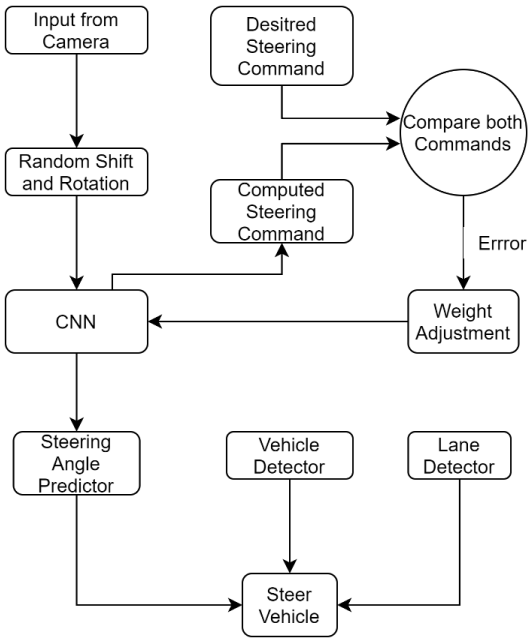


Fig 3. Training the neural network.

Our training system is depicted in Figure 3 as a block diagram. Images are sent into a CNN, which then calculates a steering command. The proposed command for the image is compared to the desired command, and the CNN weights are changed to get the CNN output closer to the desired output. Back propagation is used to make the weight change.

The agent can either be an intelligent agent which is trying to invoke the application backend or an actual user who is trying to interact with the application. Steer Vehicle functionality incorporates inputs from three different functionalities - Steering Angle Predictor, Lane Detector and Vehicle Detector and display the final output. Steering Angle Predictor is responsible for predicting the steering angle from the given input frame. Lane Detector is responsible for identifying the lanes from the given input image. Vehicle Detector functionality is responsible for identifying the vehicles from the given input images using bounding boxes. Then train the model using the neural network and export it. Finally, load the model along with the pre-trained weights

4 Implementation

The entire system can be broadly split up into two major modules.

1. Steering angle prediction using end-to-end networks
2. Straight line lane tracking using image processing techniques

A. Steering angle predictor using end to end deep learning neural networks

An end-to-end convolutional neural network was used to create the steering angle predictor. The windshield view is taken as the input and the neural network analyzes and outputs a steering angle. The datasets used for to deploy and test the Steering Angle Predictor modules are as follows:

TABLEI.DATASETSFORTRAININGEND-TO-ENDCNN

SNo	DatasetName	Alias	Contents
1	Rancho Palos Verdes Dataset	D1	45,500images,2.2GB Data format :filename.jpg angle
2	SanPedro Dataset	D2	63,000images,3.1GB Data format:filename.jpg angle,year-mm-ddhr:min:sec:millisec

Two CNN Models were considered to implement the end-to-end deep learning system

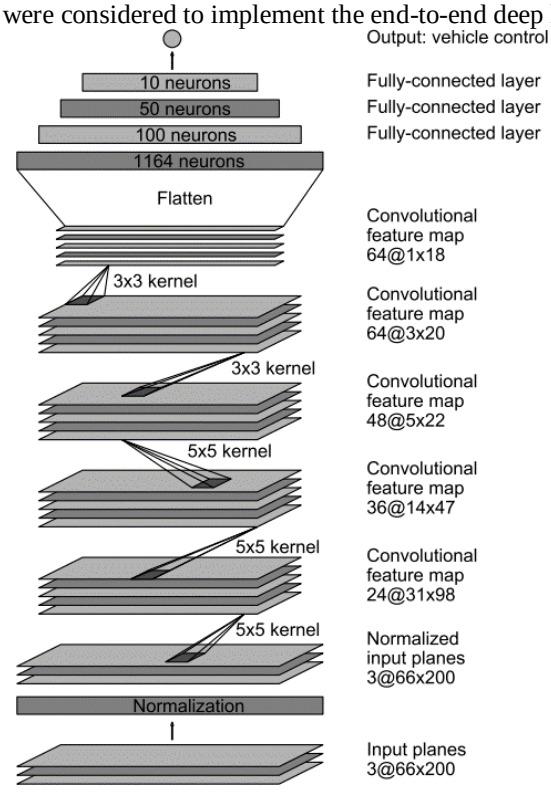


Fig.4.CNN Model1,Architectureproposedin[2]

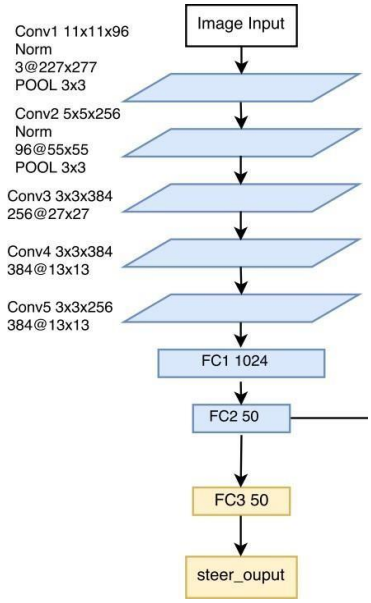


Fig.5.CNNModel2,Architectureproposedin[10][11]

The pipeline was trained, and the model was saved in a local repository. The summary of the results are tabulated as follows:

TABLE II. RMS EVALUES FOR CNN MODEL Vs DATASET

SNo.	**CNNModel**	**RMSEonD1**	**RMSEonD2**
1	CNNModel1	0.0567	0.2034
2	CNNModel2	0.0789	0.3023

Fig 5 shows that by providing the windshield view as the input,the steering angle is successfully predicted and represented in the form of a steering wheel. The two different datasets covered varied types of lanes ranging from straight lanes to curved lanes. However the RMSE values indicate that the model is not completely accurate and can be improved with the help of aiding modules.

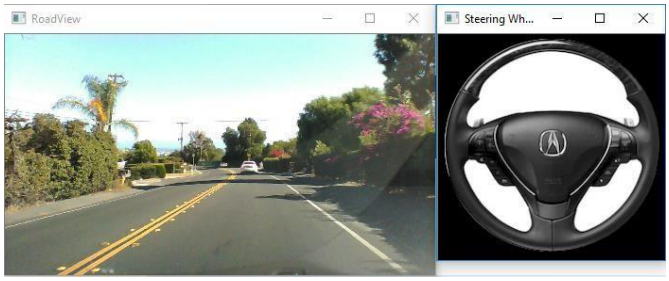


Fig. 6. Steering Angle predictor Implementation.

The model was able to predict good results, however from the tabulation it is imperative that the accuracy of the model should be increased.

B. Straight line lane detection

For lane tracking only image processing techniques were used to identify the yellow and white markings present on the road

Fig.7. Basic lane detection Implementation.

The straight lines are identified based on the yellow and white markings present on the road.

TABLEIII.IMAGE PROCESSING TECHNIQUES USED FOR LANE DETECTION

S. No.	**Method name**	**Description**
1	RGB to HLS	The first step is to select only the yellow and white colors of the image. The image is converted from RGB to HLS.
2	Grayscale	The image is thereafter converted into a grayscale format
3	GaussianBlur	Gaussian smoothing is applied to remove the noise or soften the edges since there will be a rapid change in the pixel intensity when a Lane is identified.
4	Canny EdgeDetection	The edge detection approach is Canny edge detection.
5	RegionsofInterest	A region of interest based on the edges of the lane are selected to be processed.
6	Hough LineTransform	Averaging and extrapolating multiple lines which appear in the image: The slopes of all the lines will be found and which evers lope is negative will be considered as right lane's line and whichever slope is positive will be considered as leftlane's line. Using poly fit function the combined slope and intercept value are found out, which will be used for finding out the x and y value

		separately for both Left and right direction.
7	Removing the distortion	Since each camera lens will have some radial distortions it is necessary to remove them. For that, undistort function is used for removing out all the distortions from the image.
8	Image Pre processing	Pre-processing steps are done to the undistorted images and noise is removed.
9	Perspective Transform	When the image is given as input, the lanes will look as if it is going to converge at some point. Because of this, lanes can't be detected properly. So the image is changed to a different perspective, where the image is now seen from the top. This is much easier to isolate the lane lines and to fit the curve.
10	Sliding Window Algorithm	This algorithm is used to find out all the non-zero pixels that are present in the image. Using a histogram the starting point of the lane is identified in both the directions. A box is drawn over the lane and it is restructured based on the lane's direction. The non-zero pixels which are present inside the box are used to plot the curve. Then a curve is fitted in both the directions
11	Plotting it to the image.	A polygon is drawn based on the curve values. After this step the image is turned back to its original perspective: from top-down to front view.
12	Radius of curvature	Radius of curvature is found using a mathematical formula and the offset from the center of the road is calculated. This is used to account for a probabilistic steering angle.

C. Vehicle Object Identification

The feature vector is created by using the method of the Histogram of Gradients (HOG) which uses spatial binning and various parameters like orientation, number of bins, etc. Later the features are extracted, and the hog features of the image is displayed. Support Vector Machine classification algorithm is used for training the data set having 5967 vehicles and 5232 non vehicles. Classifier Accuracy obtained was 98.78%

Fig. 8.Vehicle Detector Module Output

The vehicle detection module draws bounding boxes around the vehicles present in the image. The bounding boxes give an account of whether an obstruction is present in the view or not.

D. Integration of steering angle and lane prediction

The Basic lane detection was successfully integrated with the Steering Angle Prediction module. The below formula from[12] can be used to better the accuracy of the steering angle and reduce the RMSE.

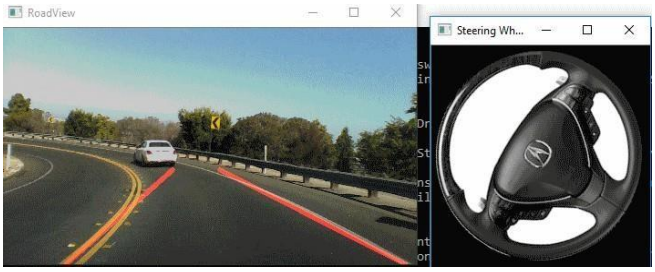
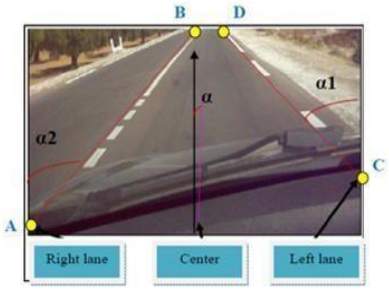


Fig. 9. Steering Angle predictor Implementation.

The model was able to predict good results, however from the tabulation it is imperative that the accuracy of the model should be increased.

Post integration the RMSE values decreased suggesting that the aiding module helped in improving the accuracy of the deep learning system.



$$\alpha = \frac{\alpha_1 + \alpha_2}{2}$$

With

$$\begin{cases} \alpha 1 = arctg\left(\frac{y_D - y_c}{x_D - x_c}\right) \\ \alpha 2 = arctg\left(\frac{y_A - y_B}{x_B - x_A}\right) \end{cases}$$

Fig.10. Integration of lane detection with steering angle prediction.

The accuracy of the model increased with the help of the aiding module.

TABLE IV. RMSEVALUESAFTERINTEGRATION

SNo.	CNNModel	RMSEonD1	RMSEonD2
1	CNNModel1	0.0540	0.2019
2	CNNModel2	0.0679	0.3018

E. Final Phase Integration

During the final phase all the three modules were successfully integrated and the steering angle was predicted based on the input of all the three resulting modules. The steering angle predictor backed by end-to-end deep learning network, the vehicle detection and curved lane tracking modules backed by image processing techniques has improved the overall prediction of the system.

Fig 11. Final integration output

From the figure we can see that all the three modules were combined into a single pipeline. The pipeline executes all the three modules simultaneously and arrives at the steering angle.

3 Conclusion And Futurework

The end-to-end deep learning System was successfully integrated with the aiding module and was able to lower the RMSE values. This suggests that by coupling image processing modules with a deep learning model, we are able to achieve better results than existing models.

Reduced costs, increased safety, and increased mobility are all advantages of providing such an algorithm for retrofitting existing cars. The project focused on particular datasets with suitable lighting conditions and lower obstructions. However, by enforcing a longer training time on varied datasets this restriction can be removed.

Future work includes the integration of an advanced lane finding model along with vehicle detection strategies in order to further increasing the accuracy of existing modules.

References

[1] Science in the News. (2018). Self-driving Cars: The technology, risks and possibilities - Science in the News. [online] Available at:http://sitn.hms.harvard.edu/flash/2017/self-driving-cars-technology- risks-possibilities/ [Accessed 17 Dec. 2018.

[2] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J. and Zhang, X., 2016. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

[3] Pomerleau, D.A., 1989. Alvinn: An autonomous land vehicle in a neural network. In Advances in neural information processing systems (pp. 305- 313).

[4] Net scale Technologies. Autonomous off-road vehicle control using end- to-end learning, July 2004. [online] Available at: http://net- scale.com/doc/net-scale-dave-report.pdf [Accessed 19 Dec. 2018].

[5] Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R. and Mujica, F., 2015. An empirical evaluation of deep learning on highway driving. arXiv preprint arXiv:1504.01716.

[6] Chen, Z. and Huang, X., 2017, June. End-to-end learning for lane keeping of self-driving cars. In Intelligent Vehicles Symposium (IV), 2017 IEEE (pp. 1856-1860). IEEE.

[7] Humaidi, A.J. and Fadhel, M.A., 2016, May. Performance comparison for lane detection and tracking with two different techniques. In Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), Al-Sadeq International Conference on (pp. 1-6).IEEE.

[8] The comma.ai driving dataset. [Online]. Available: https://github.com/commaai/research [Accessed 22 Dec. 2018].

[9] A. J. Humaidi and M. A. Fadhel, "Performance comparison for lane detection and tracking with two different techniques," in 2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), May 2016, pp. 1–6

[10] Yang, Z., Zhang, Y., Yu, J., Cai, J. and Luo, J., 2018, August. End-to- end Multi-Modal Multi-Task Vehicle Control for Self-Driving Cars with Visual Perceptions. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 2289-2294). IEEE.

[11] Ferreira, A., Almeida, A. and Vidal, F. (2018). Autonomous Vehicle Steering Wheel Estimation from a Video using Multichannel Convolutional Neural Networks. Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics.

[12] Hechri, A. and Mtibaa, A., 2011. Lanes and road signs recognition for driver assistance system. IJCSI international journal of computer science issues, 8(6).

[13] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in CVPR, 2017.

[14] J. Zhao, B. Xie, and X. Huang, "Real-time lane departure and front collision warning system on an fpga," in 2014 IEEE High Performance Extreme Computing Conference (HPEC), Sept 2014, pp. 1–5.

[15] C. Li, J. Wang, X. Wang, and Y. Zhang, "A model based path planning algorithm for self-driving cars in dynamic environment," in 2015 Chinese Automation Congress (CAC), Nov 2015, pp. 1123–1128.

[16] S. Yoon, S. E. Yoon, U. Lee, and D. H. Shim, "Recursive path planning using reduced states for car-like vehicles on grid maps," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 5, pp. 2797–2813, Oct 2015.

[17] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” CoRR, vol. abs/1408.5093, 2014.