

Steering angle prediction via neural networks

Fayez Saeed Faizi^{1,2}, Ahmed Khorsheed Alsulaifanie¹

¹Department of Electrical and Computer Engineering, College of Engineering, University of Duhok, Duhok, Iraq

²Department of Energy Engineering, Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq

Article Info

Article history:

Received Sep 28, 2022

Revised Feb 28, 2023

Accepted Mar 12, 2023

Keywords:

Autonomous vehicle

Deep neural network

Lane line detection

Lane-keeping system

Steering angle prediction

ABSTRACT

Methods to calculate the correct steering angle are an important aspect of developing self-driving vehicles. Recently, steering angle prediction methods based on deep neural networks have achieved accurate results that outperform other methods over a range of road types. This paper investigates steering angle prediction for an autonomous vehicle as a regression problem and solves it using deep neural networks. The proposed approach obtains data in the form of the coordinates of lane lines and examines if they belong to one or more lines. Based on that, the method locates the path line that the vehicle is moving on. It then determines the equation that represents that path. Finally, the coefficients of this equation are fed into a trained neural network, which predicts the steering angle for that video frame. Extensive tests were performed on the Comma.ai and Udacity benchmarks to evaluate the performance of the approach. State-of-the-art results were achieved, with mean absolute error 0.64 and root mean square error 0.87 for the Comma.ai dataset, and mean absolute error 1.04 and root mean square error 2.33 for the Udacity dataset.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Fayez Saeed Faizi

Department of Energy Engineering, Technical College of Engineering, Duhok Polytechnic University

Zakho st., Duhok, Iraq

Email: fayez.faizi@dpu.edu.krd

1. INTRODUCTION

Until recently, autonomous vehicles were merely a dream outside research centers at a few universities and companies. Thanks to rapid developments in computer processing speed, image processing techniques, and miniaturization, this dream has become a reality that one can see in the street [1]. Vehicles are now able to sense the world around them and operate autonomously in different environments to perform several tasks in addition to driving, including firefighting, mine detection and clearance, and surveillance [2]. To perform such missions, reaching the destination point is an essential task. Determining the correct steering angle that will lead to the target point is key [3].

Most steering angle calculation methods can be categorized into two approaches: computer vision-based methods and learning-based methods [4]. The first approach depends on extracting the required features to define the lane lines and then performing trigonometric calculations to determine the steering angle. The second approach exploits the power of a neural network (NN) to determine the angle. The NN-based methods are mainly end-to-end, with the system receiving the input video frame with the related steering angle in each frame and feeding it into a convolutional neural network (CNN), which predicts the steering angle based on the weights previously obtained by training the NN [5].

In this work, we exploit the advantages of both approaches and propose an algorithm that has the accuracy and generalization of the learning-based methods and the low computational cost of the vision-based methods. The proposed system receives the coordinates of points on the lane lines and tests if one or more lines

were detected. Based on the number of detected lines, the system determines the path line that the vehicle is following, then calculates an equation that represents that line. The coefficients of the equation are fed into the trained NN, which predicts the steering angle accordingly. The main contributions of this study are as:

- Different types of lane detection algorithms are used to provide lane point coordinates to the system. This is not limited to NN-based lane detection approaches, but can also exploit the traditional lane and edge detection algorithms.
- The path line followed by the vehicle is determined efficiently and represented by a quadratic equation.
- An accurate steering angle is robustly predicted at a high speed, measured in frames per second (FPS).

We now review recent studies in the field of steering angle prediction, following the aforementioned classification from [4]. The recent studies on each of the two approaches are presented, and in section 3, their results are compared with the proposed method.

On the one hand, from a learning-based method point of view, most algorithms use the end-to-end approach for steering angle prediction. Some of them use simulators, as is the case in [6], [7], while others, such as [8], [9], use custom hardware that is trained and tested in special environments. The results of such methods depend on the driving style in the training phase, which makes it hard to compare their results with other studies. Using standard benchmarks to test a proposed algorithm can make it easier to evaluate against other available approaches.

Du *et al.* [10] proposed two different end-to-end algorithms, one based on a transfer learning model using ResNet50, and the other using a 3D CNN with long short-term memory (LSTM). They used the Udacity dataset [11] to test their algorithms. Acceptable results were recorded in this work, but substantial computational resources were required. Wu *et al.* [12] presented another method using the LSTM idea: an end-to-end system depending on a convolutional-LSTM NN with a multi-scale spatiotemporal integration (MSI) module. Their experiments achieved acceptable results on the Udacity benchmark, but the system failed to control the steering angle smoothly.

Three different systems were presented in [13], one based on RGB images, one using a segmentation scheme, and one using a depth image to predict the steering angle. A combination of these methods was used to improve the results and determine the features required to obtain accurate results. Promising results were registered at the cost of expensive computation in which the processed data were fed to the “super” network that learns the important features from each network.

Hou *et al.* [14] exploited heterogeneous auxiliary network feature mimicking to create an accurate and robust steering control system. The authors found that this reduces the run time by discarding the auxiliary networks after the training phase. However, this limits the generalization ability. A method called Feudal Steering was presented in [15]. Feudal networks were utilized in hierarchical reinforcement learning (HRL) to determine vehicles’ steering angles. In this approach, the task was divided into manager and worker sub-networks to efficiently produce accurate results.

Kim and Canny [16] proposed a new steering prediction algorithm that consists of two stages. In the first stage, a visual attention model is used to train a CNN. The attention model is used to indicate the important regions in the image that affect the output of the network. In the second stage, a causal filter is applied to find the region in the image that has an actual influence on the prediction of the steering angle. The proposed approach was tested on the Udacity and Comma.ai [17] benchmarks and promising results were recorded.

Chen [18] introduced a simple end-to-end steering control algorithm for self-driving cars, consisting of a CNN that was trained and tested on the Comma.ai dataset. The system obtains an image from a front view camera and feeds it to the CNN for steering angle prediction. Such schemes produce acceptable speed and robustness.

On the other hand, computer vision-based paradigms generally have two phases: lane line feature extraction and steering angle calculation. The first step may use traditional lane detection techniques, such as edge detection or the Hough transform [19], or it may utilize an NN-based lane detection approach [20].

Rochan *et al.* [21] proposed an algorithm for steering angle calculation that consists of three phases. Phase one uses a Gaussian mixture model and an expectation maximizing approach to extract the drivable road parts. Next, the steering angle is calculated as the deviation of the intersection point of the roads’ edges from the midline of the image. In the last phase, spurious steering angle transition noise is removed using a Kalman filter.

Another approach was proposed by Rasib *et al.* in [22], which has two main stages. First, pixel-level segmentation is performed to extract and indicate the road region in the image using a CNN end-to-end scheme. Then, trigonometric and line interception are used to compute the steering angle. The authors also presented a new benchmark named carl-dataset that can be used to test and evaluate steering angle prediction algorithms.

A comparison study was performed by Nguyen *et al.* in [23] to evaluate the performance of a vision-based method compared with an advanced deep learning method. They presented two methods, the first one based on standard image processing methods such as edge detection and the Hough transform, and the second one based on a CNN. The first method has four steps: pre-processing, edge detection, the Hough transform, and finally steering angle computation. The second method is an end-to-end CNN-based approach. Testing

both methods on a common dataset showed that the image processing-based method is faster and attains more accurate results in static and simple backgrounds. Nevertheless, the CNN-based method was better generalized to the changes in the background.

2. METHOD

Since an autonomous land vehicle in a neural network (ALVINN) [24] successfully used neural networks to drive a vehicle autonomously, a variety of studies have applied NNs and introduced new self-driving algorithms [25]. Unlike the models presented in the literature, the proposed method treats steering angle prediction as a regression problem in a novel manner. The following pseudo-code summarizes the main ideas of the Algorithm 1.

Algorithm 1. Pseudo-code summarizes the main ideas

```

Start
Video frame received with the lane line points detected previously.
Calculate the lane line equation(s) from the detected points.
If: both lane lines are detected: then
  Calculate the path line between them
Feed the path line equation to the NN
Steering angle value = output of NN

```

2.1. Getting points on the lane line

The proposed algorithm needs to determine the lane feature coordinates in the incoming frame for use later in the steering angle prediction process. This information is obtained using lane detection algorithms like CurveLane-NAS [26], E2E-LMD [27], or LaneATT [28]. This information is used later to calculate the quadratic equations representing the lane lines. This step adds great flexibility by allowing the proposed system to adapt to different lane line detection algorithms.

2.2. Calculating lane line equations

The feature coordinates are then used to find one or more equations representing the lane lines. If both left and right lane lines were detected in the last phase, the received coordinates of the feature are categorized into two groups indicating each lane line. All the points in each group are fitted to build a quadratic equation curve that represents that lane line. Then a center curve between both lane curves is calculated, giving the path line:

$$x_{center} = \frac{x_{right} - x_{left}}{2} + x_{left} \quad (1)$$

since $y_{left} = y_{right}$, y_{center} takes the same value. Figure 1 explains this step and shows the incoming and resulting lane lines.

Figure 1 illustrates the case in which both lane lines are detected: Figure 1(a) shows that both left and right lanes (blue and orange respectively) are detected, while Figure 1(b) shows the center curve path denoted by the green dotted line. The center path is the path that the vehicle will follow. In the case in which only one lane line (left or right) is detected, the system calculates the equation for that line and considers it as the path line. After the path line equation is determined, it is fed into the neural network for steering angle prediction.

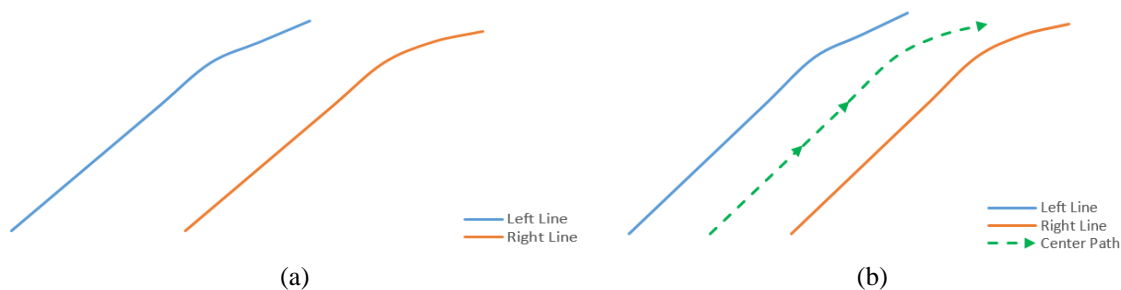


Figure 1. The case in which both lines are detected (a) left and right lane lines and (b) both lines and the new calculated path line

2.3. Network architecture

The Keras library [29] was used to build the artificial neural network (ANN) deployed in the proposed algorithm. The proposed ANN consists of 6 dense layers. The first 5 dense layers each have a rectified linear unit (ReLU) activation function and are each followed by a dropout layer. The first layer has a depth of 1024 with a dropout of 40%, and the second has a depth of 512 and a dropout of 30%. The depth of the following two layers is 256 with dropouts of 20% and 10%, respectively. The fifth dense layer has a depth of 128 with a 5% dropout. The last dense layer is a single neuron that has a linear activation function with one output, which represents the predicted steering angle. Figure 2 illustrates the architecture of the ANN used in this work.

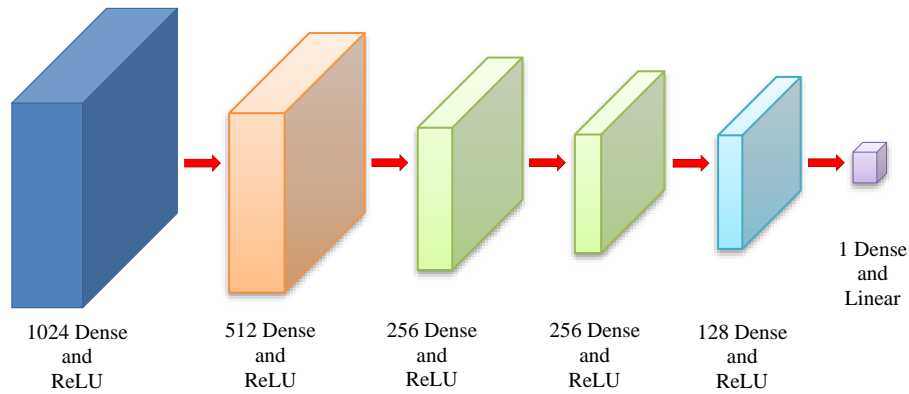


Figure 2. The architecture of the proposed ANN

2.4. Loss function and optimization

The mean square error (MSE) loss function was used to train the model. Due to the square term, this function harshly penalizes large errors, which must be avoided. MSE is a common loss function used in regression problems [10]. It is defined as the average of all square differences between real and predicted values:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \tag{2}$$

where y_i represents the predicted steering angle, \hat{y}_i indicates the real steering angle, and N is the total number of predicted values. The loss was optimized using the Adam optimizer [30], which is a common choice in ANN deep learning applications as it usually outperforms other stochastic gradient descent methods because of its computational efficiency and high adaptive learning ability [13].

2.5. Training

Before starting the training process, the dataset was tested to check the distribution of the steering angle values. It is important to have a balanced dataset in the training process to prevent any bias [31]. As shown in Figure 3 shows the normalized steering angle distribution of the dataset from Comma.ai [17].

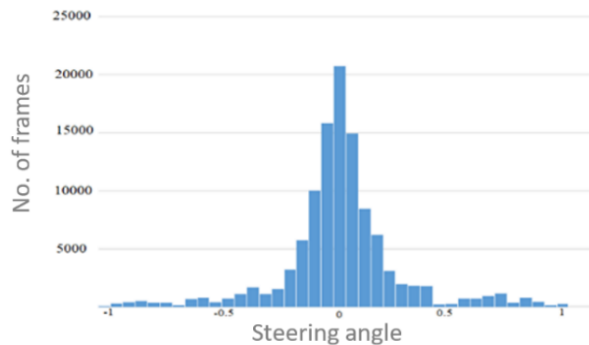


Figure 3. The normalized steering angle distribution of the training dataset

Figure 3 indicates that the dataset is fairly distributed over the range of steering angle values, with a near-Gaussian distribution except for a sharp peak around zero, as the driver keeps the steering wheel in the neutral position most of the time. The dataset is therefore suitable to train the system. The dataset was split with a 7:3 ratio, with 70% (771) for training and the remaining 30% (329) for validation. After the training process started, the model began to learn how to predict the steering angle based on the available line curve equation coefficients. Figure 4 shows the model loss during training and validation. It can be seen from Figure 4 that the system started to converge after around 25 epochs of iterative training.

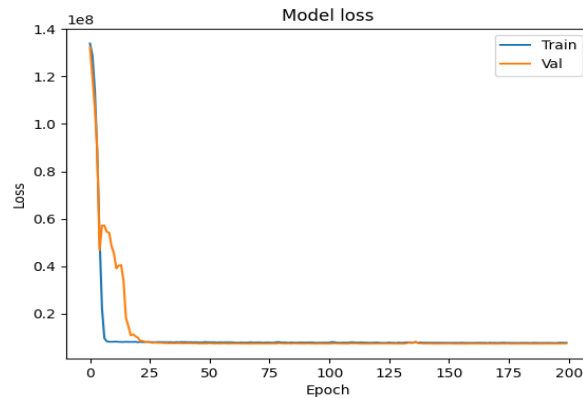


Figure 4. The training and validation loss curves

2.6. Steering angle prediction

The proposed method estimates the steering angle of the autonomous vehicle using the path line equation coefficients. So, to operate the proposed algorithm, the coefficients of the path line equation obtained during the last phase are introduced to the first dense layer, with the connection weights changing accordingly. The outputs from the first layer are fed to the next layer and so on until the final neuron produces the predicted steering angle.

2.7. Evaluation metrics

To evaluate the proposed method's performance, the mean absolute error (MAE) and the root mean square error (RMSE) were used. The MAE, defined by (3), gives all predictions the same weight to calculate the average error. The RMSE, defined by (4), is the standard deviation of the tested data and also provides a unit-based evaluation of the system [13]. The prevalence of the predicted data refers to the accuracy of the tested system. A low RMSE value indicates high accuracy.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (3)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (4)$$

The MAE is always less than the RMSE; the difference indicates the variance in the predictions. Lower difference values indicate low variance, while higher values indicate high variance [13].

3. RESULTS AND DISCUSSION

Extensive experiments have been conducted on various datasets to test and evaluate the proposed method. To evaluate the proposed system fairly, the results of those experiments were compared to some recent studies' results. The results for each benchmark are presented separately, followed by a detailed discussion.

3.1. Datasets

The Udacity [11] and Comma.ai [17] benchmarks were used to test and evaluate the proposed algorithm. The Udacity dataset has 8 hours of driving data in sunny and overcast weather, containing several challenging scenes. Comma.ai has 11 video clips with a total of 7.25 hours of driving data. Further details about the datasets are given in Table 1.

Table 1. Details of the datasets used for testing

Dataset	No. of frames	FPS	Duration (hours)	Lighting	Road
Udacity	577,900	20	8	Day	Urban
Comma.ai	522,434	20	7.25	Day and Night	Highways and Urban

Both datasets were used separately. During the tests, each dataset was processed and, as explained in section 2.2, a table containing the lane line information for the current video frame was fed into the network. Then, the system predicted the steering angle for that frame, and the process was repeated for each frame.

3.2. Experimental results

Extensive tests were performed on the two datasets to evaluate the proposed algorithm and compare the results with those of some recent studies, namely [10], [12]-[16], [18]. First, the scheme was tested on the Udacity dataset and the results were compared with other models that have been applied to the steering angle prediction task. The results are listed in Table 2.

Table 2. Comparison of the proposed algorithm with existing methods on the Udacity dataset

Method [ref.]	MAE (degrees)	RMSE (degrees)
3D LSTM [10]	-	6.44
ResNet50 Transfer [10]	-	4.06
RGB + Seg [13]	1.19	2.43
RGB + Depth [13]	1.02	2.32
RGB + Seg + Depth [13]	1.25	2.56
FM-Net [14]	1.62	2.35
Feudal Steering [15]	1.09	2.67
MSINet [12]	-	2.81
CNN + FCN [16]	4.12	4.83
CNN + LSTM [16]	4.15	4.93
Ours	1.04	2.33

Table 2 illustrates the strong performance of the proposed algorithm (shown in bold): it achieves almost the same scores (an MAE of 1.04 and an RMSE of 2.33) as the best results in the literature. In the second test, the Comma.ai benchmark was used to evaluate the performance of the proposed method. Table 3 shows a comparison of the proposed system with other schemes on the same dataset. It is clear from Table 3 that the proposed method outperforms other methods in both the MAE and RMSE evaluation metrics. New state-of-the-art scores were achieved: an MAE of 0.64 and an RMSE of 0.87.

Table 3. Comparison of the proposed algorithm with existing methods on the Comma.ai dataset

Method [ref.]	MAE (degrees)	RMSE (degrees)
FM-Net [14]	0.7	0.98
CNN + FCN [16]	2.54	3.26
CNN + LSTM [16]	2.58	3.39
Chen [18]	2.42	-
Ours	0.64	0.87

3.3. Discussion

The proposed method was tested on two standard benchmarks, Udacity and Comma.ai. In the first test results listed in Table 2, our model shows almost the same results as [13]. Their scores were slightly above ours due to the additional processing stages they used: their approach connects several networks together, each of which performs its task and feeds the processed data to a “super” network that learns the important features from each network. This method has a high computational cost compared with our algorithm. Still, our method achieves more than 98% accuracy, which is close to the best-registered results for the Udacity dataset. In the second experiment, the Comma.ai dataset was used to test and evaluate the proposed system. The results in Table 3 demonstrate the superiority of the proposed method over other approaches on the same dataset: our algorithm outperformed the best-registered results by almost 9% and 11% for MAE and RMSE, respectively.

However, the accuracy values presented in this and previous studies are imperfect. Both evaluation metrics (MAE and RMSE) depend on the difference between the predicted steering angle and the recorded ground-truth angle. This ground-truth angle represents the driving style of a specific human driver and is not necessarily the optimal steering angle at that time. It is also not possible for the driver to keep the car precisely

in the middle of the lane for the entire time they are driving. This means that the predicted steering angle does not have to be exactly the same as the value attained by a human driver, because the same driver may record different steering angle values when driving again on the same road. So, one can say that as long as the car stays in the lane and follows its path, the current steering angle can be considered a correct angle.

4. CONCLUSION

This work investigates steering angle prediction for an autonomous vehicle as a regression problem and solves it using an ANN. The proposed prediction algorithm obtains data that represent the coordinates of some points on the lane line and then checks if one or both lines were detected in the previous stage. Based on that, the system determines the equation of the path line. Finally, the coefficients of the evaluated equation are fed into the trained NN, which predicts the steering angle for that frame. The experimental results, from testing on two different benchmarks, demonstrated the accuracy and robustness of the proposed algorithm and its superiority over other state-of-the-art NN-based methods. Nevertheless, the accuracy of the scheme is affected by the accuracy of the lane detection algorithm and its ability to extract the required features efficiently.

ACKNOWLEDGEMENTS

This work was supported by the UoD and DPU, Ministry of Higher Education (MHE), Kurdistan Region Government (KRG), Iraq, as part of the higher studies program in the Ph.D. plan for the academic years 2020–2023.




REFERENCES

- [1] P. Velaskar, A. Vargas-Clara, O. Jameel, and S. Redkar, "Guided navigation control of an unmanned ground vehicle using global positioning systems and inertial navigation systems," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 4, no. 3, pp. 329–342, Jun. 2014, doi: 10.11591/ijece.v4i3.5183.
- [2] A. I. N. Alshbatat, "Behavior-based approach for the detection of land mines utilizing off the shelf low cost autonomous robot," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 2, no. 3, Sep. 2013, doi: 10.11591/ijra.v2i3.2038.
- [3] H. Saleem, F. Riaz, L. Mostarda, M. A. Niazi, A. Rafiq, and S. Saeed, "Steering angle prediction techniques for autonomous ground vehicles: a review," *IEEE Access*, vol. 9, pp. 78567–78585, 2021, doi: 10.1109/ACCESS.2021.3083890.
- [4] A. Oussama and T. Mohamed, "A literature review of steering angle prediction algorithms for self-driving cars," in *Advances in Intelligent Systems and Computing*, vol. 1105 AISC, 2020, pp. 30–38.
- [5] U. M. Gidado, H. Chiroma, N. Aljojo, S. Abubakar, S. I. Popoola, and M. A. Al-Garadi, "A survey on deep learning for steering angle prediction in autonomous vehicles," *IEEE Access*, vol. 8, pp. 163797–163817, 2020, doi: 10.1109/ACCESS.2020.3017883.
- [6] J. Zhang and H. Huang, "Steering angle prediction for autonomous cars based on deep neural network method," in *2020 Australian and New Zealand Control Conference (ANZCC)*, Nov. 2020, pp. 205–208, doi: 10.1109/ANZCC50923.2020.9318380.
- [7] T. Qiu and Z. Huang, "Learning a steering decision policy for end-to-end control of autonomous vehicle," in *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*, Apr. 2019, pp. 347–351, doi: 10.1109/ICCAR.2019.8813428.
- [8] A. K. Jain, "Working model of self-driving car using convolutional neural network, Raspberry Pi and arduino," in *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, Mar. 2018, pp. 1630–1635, doi: 10.1109/ICECA.2018.8474620.
- [9] D. V. P. Mygapula, S. Adarsh, V. V. S. Variyar, and K. P. Soman, "CNN based End to end learning steering angle prediction for autonomous electric vehicle," in *2021 4th International Conference on Electrical, Computer and Communication Technologies, ICECCT 2021*, Sep. 2021, pp. 1–6, doi: 10.1109/ICECCT52121.2021.9616875.
- [10] S. Du, H. Guo, and A. Simpson, "Self-driving car steering angle prediction based on image recognition," *arXiv preprints*, Dec. 2019, [Online]. Available: <http://arxiv.org/abs/1912.05440>.
- [11] Udacity, "udacity/self-driving car," *github*, 2021. [Online]. <https://github.com/udacity/self-driving-car> (accessed Sep. 01, 2022).
- [12] T. Wu, A. Luo, R. Huang, H. Cheng, and Y. Zhao, "End-to-end driving model for steering control of autonomous vehicles with future spatiotemporal features," in *IEEE International Conference on Intelligent Robots and Systems*, Nov. 2019, pp. 950–955, doi: 10.1109/IROS40897.2019.8968453.
- [13] J. Munger and C. W. Morato, "How many features is an image worth? multi-channel CNN for steering angle prediction in autonomous vehicles," in *AI, Machine Learning and Applications*, Aug. 2021, pp. 01–19, doi: 10.5121/csit.2021.111301.
- [14] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning to steer by mimicking features from heterogeneous auxiliary networks," in *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, Nov. 2019, pp. 8433–8440, doi: 10.1609/aaai.v33i01.33018433.
- [15] F. Johnson and K. Dana, "Feudal steering: Hierarchical learning for steering angle prediction," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2020, vol. 2020-June, pp. 4316–4325, doi: 10.1109/CVPRW50498.2020.00509.
- [16] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, vol. 2017-October, pp. 2961–2969, doi: 10.1109/ICCV.2017.320.
- [17] E. Santana and G. Hotz, "Learning a driving simulator," *arXiv preprints*, Aug. 2016, [Online]. Available: <http://arxiv.org/abs/1608.01230>.
- [18] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1856–1860, doi: 10.1109/IVS.2017.7995975.
- [19] V. Kamat and S. Ganesan, "Robust Hough transform technique for description of multiple line segments in an image," in *IEEE International Conference on Image Processing*, 1998, vol. 1, pp. 216–220, doi: 10.1109/icip.1998.723460.




- [20] V. Umamaheswari, S. Amarjyoti, T. Bakshi, and A. Singh, "Steering angle estimation for autonomous vehicle navigation using hough and Euclidean transform," in *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, SPICES 2015*, Feb. 2015, pp. 1–5, doi: 10.1109/SPICES.2015.7091469.
- [21] M. R. Rochan, K. A. Alagammai, and J. Sujatha, "Computer vision based novel steering angle calculation for autonomous vehicles," in *Proceedings - 2nd IEEE International Conference on Robotic Computing, IRC 2018*, Jan. 2018, vol. 2018-January, pp. 143–146, doi: 10.1109/IRC.2018.00029.
- [22] M. Rasib, M. A. Butt, F. Riaz, A. Sulaiman, and M. Akram, "Pixel level segmentation based drivable road region detection and steering angle estimation method for autonomous driving on unstructured roads," *IEEE Access*, vol. 9, pp. 167855–167867, 2021, doi: 10.1109/ACCESS.2021.3134889.
- [23] K. D. N. Tu, H. D. Nguyen, and T. H. Tran, "Vision based steering angle estimation for autonomous vehicles," in *International Conference on Advanced Technologies for Communications*, Oct. 2020, vol. 2020-October, pp. 187–192, doi: 10.1109/ATC50776.2020.9255456.
- [24] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in Neural Information Processing Systems 1*, pp. 305–313, 1989, doi: 10.1184/r1/6603146.
- [25] N. Haque and M. H. Riaz, "Autonomous vehicle control system as a mobile robot by artificial neural network," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 6, no. 3, p. 200, 2017, doi: 10.11591/ijra.v6i3.pp200-206.
- [26] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "CurveLane-NAS: Unifying lane-sensitive architecture search and adaptive point blending," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12360 LNCS, 2020, pp. 689–704.
- [27] S. Yoo *et al.*, "End-to-End lane marker detection via row-wise classification," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, vol. 2020-June, pp. 4335–4343, doi: 10.1109/CVPRW50498.2020.00511.
- [28] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Keep your eyes on the lane: real-time attention-guided lane detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 294–302, doi: 10.1109/CVPR46437.2021.00036.
- [29] F. Chollet, "Keras," *Keras.io*, 2022. [Online]. <https://keras.io/> (accessed Sep. 01, 2022).
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014, pp. 1–15, [Online]. Available: <http://arxiv.org/abs/1412.6980>.
- [31] J. Sompokiala, "Prediction of steering angle for autonomous vehicles using pre-trained neural network," *European Journal of Engineering and Technology Research*, vol. 6, no. 5, pp. 171–176, Aug. 2021, doi: 10.24018/ejeng.2021.6.5.2537.

BIOGRAPHIES OF AUTHORS



Fayez Saeed Faizi    is a lecturer at the Technical College of Engineering, Duhok Polytechnic University, Iraq. He received a B.Sc. degree in Electronic Engineering from Mosul University, Iraq, and an M.Sc. degree in Computer Engineering from the University of Duhok, Iraq. Now he is a Ph.D. candidate in Computer Engineering at the University of Duhok with a specialization in deep learning robotic techniques. The research areas he is interested in include image processing, robotic control, and deep learning. He can be contacted at email: fayez.faizi@dpu.edu.krd.



Ahmed Khorsheed Alsulaifanie    received B.Sc., M.Sc., and Ph.D. degrees in Electrical Engineering in 1977, 1980, and 2003 respectively from the University of Mosul, Iraq. He worked for 40 years as an academic staff member. His-research interests in the fields of computer architecture, digital signal processing, and image processing. He can be contacted at email: ahmed.khorsheed@uod.ac.