



Transport and Telecommunication, 2023, volume 24, no. 2, 110–119
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2023-0010

EGO VEHICLE LANE DETECTION AND KEY POINT DETERMINATION USING DEEP CONVOLUTIONAL NEURAL NETWORKS AND INVERSE PROJECTION MAPPING

Anudeepsekhar Bolimera¹, Raja Muthalagu², V. Kalaichelvi³ and Abhilasha Singh⁴

¹Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA,
anudeep.sekhar@gmail.com

² Department of Computer Science, Birla Institute of Technology and Science Pilani,
Dubai Campus, Dubai, UAE
raja.m@dubai.bits-pilani.ac.in

³ Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science Pilani,
Dubai Campus, Dubai, UAE
kalaichelvi@dubai.bits-pilani.ac.in

⁴Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science Pilani,
Dubai Campus, Dubai, UAE
p20180906@dubai.bits-pilani.ac.in

Ego lane detection is one of the key techniques in Ego Lane Analysis System (ELAS) implemented in smart autonomous driving cars for lane detection in roads. This technique has been extensively studied in recent years because of its accurate and robust detection of shape and location of lanes. The conventional methods are less robust and computationally expensive since they have several challenges in localization of lanes due to presence of occlusions on roads. So to avoid these issues, this paper uses a novel 2-stage lane detection method using deep convolutional neural network to detect the lanes and its key-points by optimally fit a curve to the lane to compensate on above mentioned issues. The proposed methodology for lane detection predicts the key-points accurately and it robust under various weather conditions and highway driving scenarios. In terms of performance, this technique is fast and robust with low computational cost and has high performance when deployed on autonomous vehicle-based systems.

Keywords: Ego Lane Detection, Deep Convolution Neural Network, Ego Lane Analysis System, Driving Scenarios, Occlusions, Autonomous Vehicle

1. Introduction

In recent years due to the rapid development of the technology, the demand for smart automobiles is also raising all over the world (Singh *et al.*, 2015; Bellis *et al.*, 2008) which thereby increases the research and developments in the area of advanced driver assistance systems (ADAS) and driverless cars using computer vision technology. The lane detection is the foundation of many advanced driver assistance systems such as Lane Departure Warning Systems (LDWS), Ego Lane Analysis System (ELAS) and the Lane Keeping Assistance System (LKAS) (Gayko, 2012; Visvikis *et al.*, 2008) along with other sensors and computer vision techniques to localize themselves for object identification and navigation. Among that lane detection and perception is most widely researched topics for the past few years by individuals and institutions for improving the safety of the drivers and reduce accidents. In either of the applications mentioned above, the main goal is to understand the environment around the car using various sensors like camera and using different intelligent algorithms. The main goal of lane detection algorithms is nothing, but it has to detect the false positives accurately and adapt itself to the environmental changes accordingly (Neven *et al.*, 2018). In the existing literature, Hough Transform is commonly used to detect the edge pixels of the lane lines in the image, but it is computationally extensive and often straight lines are detected as false detections under various road conditions. Hence in this paper, lane lines are identified by scanning only a limited number of horizontal lines within a small region of interest. To improve the reliability of real time road lane detection, Automatic thresholding and spatiotemporal filtering procedures are implemented in this work under different conditions.

In general, there are three main steps in vision-based lane detection namely image pre-processing, lane detection, and lane tracking (Yan *et al.*, 2017). There are various image pre-processing techniques like region of interest (ROI) selection, colour space conversions, distortion correction, and inverse perspective mapping (IPM) (Oliveira *et al.*, 2015). Next, the lane detection is the most crucial step that involves feature extraction using Canny Edge Detector (Muthalagu *et al.*, 2021) and model fitting using colour space exploration but Hough Transform algorithm, is commonly used for lane detection (Amaradi *et al.*, 2016). There are many works reported in the area of lane detection using both conventional and deep learning techniques. In Conventional vision-based lane detection, there are two main methods namely feature-based techniques and model-based techniques, and authors implemented the above techniques using the driver's indoor video and developed a driver's movement tracking (DMT) algorithm in (Muthalagu *et al.*, 2021). In Feature-based techniques, Features like lane colour, texture, and edges are detected and extracted using Sobel filters or canny edge detectors and finally converts the RGB images into grayscale or HSV. In colour space transformation the colours like yellow and white are mostly used for lane markings as they are more distinct which can be used to filter the outliers. Hence feature-based methods have better accuracy and efficiency when lane markings are vibrant, but they lose robustness under dynamic environmental conditions. Furthermore, in model-based algorithms, lanes are described as a linear model, parabolic model or spline model using suitable polynomial equations (Muthalagu *et al.*, 2021). They are hard to implement and more vigorous than feature-based methods but due to numerous iterations they are computationally expensive.

The performance of lane detection process can be improved by using deep learning techniques particularly for complex scene understanding under various environmental conditions. In the field of computer vision, Convolutional Neural Network (CNN) methods are primarily used for feature extraction (Oliveira *et al.*, 2015; Muthalagu *et al.*, 2021) and semantic segmentation to segment the shape and the location of lanes (Xing *et al.*, 2018; Wang *et al.*, 2018). In (Amaradi *et al.*, 2016), weakly-supervised adversarial consisting of three deep neural networks domain adaptation is implemented to improve the segmentation performance from synthetic data to real scenes and the object detection and prediction of segmentation map uses Detection and segmentation (DS) model. Though some of the previous works can detect the lanes accurately, it has some limitations as this framework needs pre-processing step making the algorithm computationally expensive. The algorithm may fail to detect the lanes from the pre-defined ROI so this motivated us to improve the efficiency and robustness of the algorithm by reducing its dependency on heavily pre-processed images and extend the above framework using deep learning techniques. Based on the previous studies, the contributions in this paper are summarized as follows.

1. A novel 2- stage lane detection method is proposed using deep convolutional neural networks to detect the lanes and key-points which optimally fit a curve to the lane.
2. To detect the lane markings robustly in various weather conditions and highway driving scenario, deep convolutional neural networks (CNNs) and Inverse Projection Mapping which are used
3. To predict the key-points which is left and right lane-markings of the ego vehicle lane accurately the proposed model combines lane mask proposal network with a lane key-point determination network

The organization of the paper is as follows. The proposed pipeline for semantic and instance lane segmentation with our approach in conversion from segmented lane instances into parametric lines is discussed in depth in Section II. The experimental results of the proposed methodology is illustrated in Section III followed by conclusion in Section IV.

2. Proposed Methodology

2.1. Introduction

In this section, we proposed a new pipeline for lane detection which overcome the shortcomings of conventional lane detection approaches as discussed in the previous section. The main drawback of the conventional approaches is a delay in decision making because a lane detection algorithm is running either at the server or computational centre and the prediction accuracy and the robustness of diverse scenarios are crucial. In the proposed method, the lane detection algorithms are running at the vehicle which reduces the time delay in decision making and reduces the computational cost. In this paper, the framework for the lane detection is proposed based on two different neural network design which can work in diverse scenarios. The first design detects the lane marking and produces a pixel-wise lane marking proposal map. The second localization network determines the localization and shape of each lane in the form of key-points.

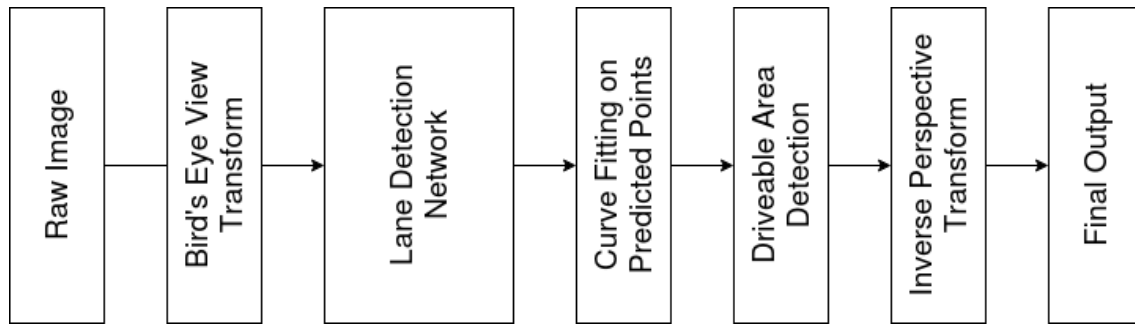


Figure 1. Framework of the Proposed Approach for Lane Detection

Figure 1. Shows the framework of the proposed lane detection approach. The raw undistorted input image is converted into a “birds-eye view” of the road by using perspective transformation to improve the accuracy of the lane detection. The perspective transformed applied image is then fed to Lane Key Point Detection Deep Neural Network to predict the key points describing the lane markings. These key points are later used to determine the lanes and safe drivable area for the ego vehicle. Figure 2(a) and Figure 2(b) show the input raw image captured from the camera and bird's-eye view of the original image after fed to the perspective transform respectively.

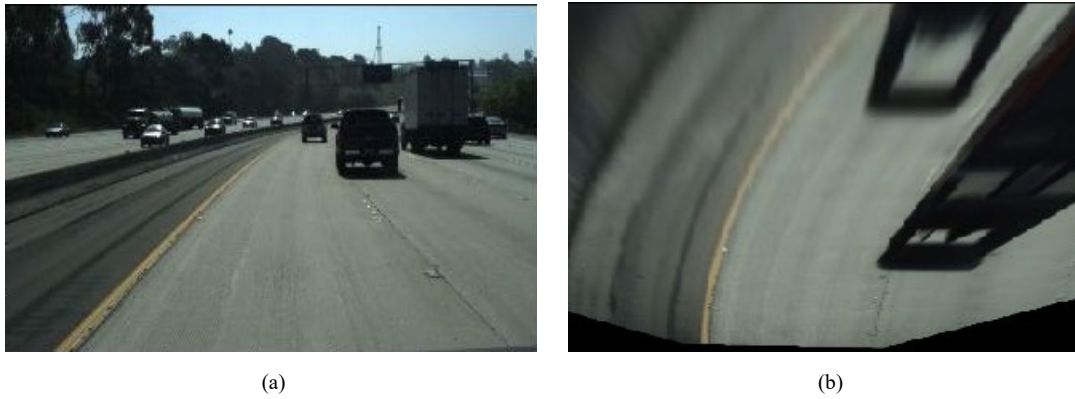


Figure 2. (a) Input Raw Image captured from the camera (b) Image View transformed to 'Bird's Eye View' using the perspective transformation

2.2. Proposed Lane Detection Model Architecture

The proposed network architecture has two major parts; the first part is an Encoder-Decoder Lane mask proposal network which is used to filter out the irrelevant information contained in optical images. Next it is used to generate a binary lane proposal map to extract the position of the pixels that lie on the lane markings. The second part of the network is a Lane Key Point Determination network, which generates key-points to describe the whole lane and distinguishes the points into each lane instance. Figure 3. provides an overview of our proposed model architecture.

2.2.1. Lane Mask Proposal Network

We use Unet (Ronneberger *et al.*, 2015) as our base feature encoder and decoder framework. In our proposed method, the hand-craft feature is replaced with a deep neural network for proposal maps since it is difficult to find robust handcraft features for filtering lane edges in an input image. The Deep convolutional neural network is a suitable tool for deriving global appearance, contextual information, and accurate lane mask proposals on each pixel and it helps to suppress false positive detection. The dense prediction tasks like semantic segmentation (Ronneberger *et al.*, 2015; Shelhamer *et al.*, 2017; Badrinarayanan *et al.*, 2017) are using encoder-decoder architecture which typically utilize convolutional layers and transpose convolution layers for feature encoding and decoding.

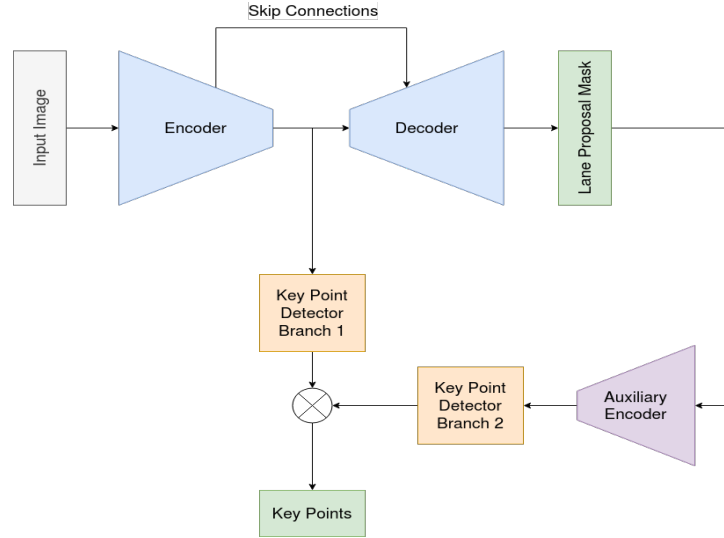


Figure 3. Proposed Lane Detection Model Architecture

The encoder architecture consists of two 3x3 convolutions followed by the Batch Normalization layer and a rectified linear unit (ReLU). The batch normalization layer standardizes the inputs and makes the training faster whereas ReLU gives only the positive output. Secondly it has 2x2 max pooling operation with stride 2 for down sampling the images and featured channels are doubled at each down sampling steps. The decoder architecture consists of up sampling of the feature maps followed by a 2x2 convolution that halves the number of feature channels. Next the concatenation of cropped feature map from the contracting path is performed and finally it has two 3x3 convolutions, each followed by a ReLU. The high-resolution features for accurate lane edge localization are provided by skip connections and the final high-resolution feature map is then used to generate the pixel-wise lane marking probability map. The value of each pixel indicates the confidence value of pixels that lies on the edge of a lane segment in the input image.

2.2.2. Auxiliary Encoder

The auxiliary encoder creates feature embeddings from the lane proposal mask. The module consists of residual bottleneck blocks similar to the ones proposed in (He *et al.*, 2016). Figure 4. shows the composition of the blocks used to the auxiliary encoder. The module consists of four auxiliary blocks, each followed by a 2x2 max pooling operation to reduce the size of the feature map. Finally, a 1x1 convolution is applied to create the feature embeddings for the key points determination network.

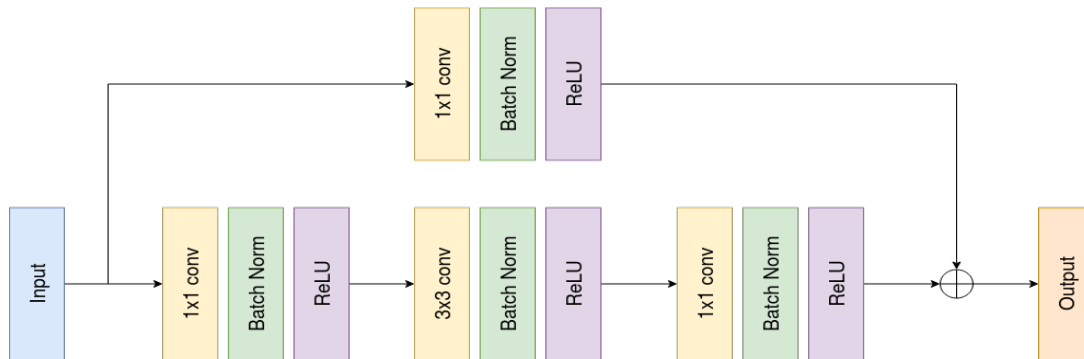


Figure 4. Architecture of a Single Unit of the Auxiliary Encoder

2.2.3. Lane Key Points Determination Network

The lane key-point determination network predicts the x and y coordinates of 30 key points (15 key-points per lane), which describe the locations of the two-lane markings between which the ego vehicle is driving. The network comprises two distinct branches as two fully connected networks, each of which tries

to predict the key points using different sets of input feature embeddings. The network takes two kinds of feature embeddings as input, one from the lane mask proposal network's encoder end, and a set of features are derived from the output of the lane mask proposal network's decoder end by the auxiliary encoder. The input to each of the branches is 1024 long tensors of the feature embeddings and each of the branch's outputs a 60-unit long tensor. The key points are determined as a linear combination of the outputs from both branches of the key-point determination network.

2.2.4. Loss Functions

The key-point generated is used to improve the accuracy of lane detection and the auxiliary training task is used to predict the lane mask proposals. It is important to calculate the hybrid loss function as the nature of these two predictions and they are completely diverse from each other. The lane mask proposal prediction is a pixel-wise classification task, and a binary cross-entropy loss is considered as segmentation loss. The key-points prediction is a regression problem, and the Mean Squared Error (MSE) is considered as a loss function. For segmentation loss, a pixel-wise soft-max over the feature map is combined with a cross-entropy loss function to calculate an energy function. The softmax is combined as

$$p_k(x) = \frac{e^{a_k(x)}}{\sum_{k'=1}^K e^{a_{k'}(x)}}, \quad (1)$$

where $a_k(x)$ - the activation in feature channel k at the pixel position $x \in \Omega$ with $\Omega \subset \mathbb{Z}^2$, K is the number of classes and $p_k(x)$ is the approximated maximum-function. i.e. $p_k(x) \approx 1$ for the k that has the maximum activation $a_k(x)$ and $p_k(x) \approx 0$ for all other k . The cross entropy then penalizes at each position the deviation of $p_{l(x)}(x)$ from (2) using

$$L_{\text{Segementation}} = \sum_{x \in \Omega} w(x) \log(p_{l(x)}(x)), \quad (2)$$

where $l: \Omega \rightarrow \{1, \dots, K\}$ is the true label of each pixel and $w: \Omega \rightarrow \mathbb{R}$ is a weight map that we introduced to give some pixels more importance in the training. The lane markings are defined with 30 key points and MSE between the ground truth x and y coordinates and the prediction is calculated to know the success rate of the detection of lane key points. MSE is the average of the square of that pixel-to-pixel difference between two point's positions.

$$L_{\text{Lane}} = \frac{1}{N} \sum (Y_{\text{true}} - Y_{\text{predicted}})^2. \quad (3)$$

The total loss is calculated as a weighted sum of the segmentation loss and the MSE loss.

$$L_{\text{Total}} = L_{\text{Segementation}} + L_{\text{Lane}}. \quad (4)$$

2.3. Post Processing and lane Detection

In post processing, the output of the network is a collection of pixels per lane and fitting a linear polynomial through these pixels is not ideal because for curved lanes higher-order polynomials is needed to face the problem of the vanishing point. To deal with this issue, curve fitting in the birds-eye-view is performed and since the lanes appear nearly parallel to each other second to third-order polynomial is used in this study. The detected lane boundaries are transformed back into the original image which is used for numerical estimation of lane curvature as shown in Figure 5.

3. Results and Discussion

3.1. Dataset Preparation

The tuSimple lane dataset is commonly used for land detection task which (GitHub, 2018) has 6408 images. The deep learning network is trained with 3626 training images and tested with 2782 testing images. The datasets consist of 2-lane/3-lane/4-lane or more highway roads video recordings at different daytimes and weather conditions. The annotations of each image come in a JSON format which has x-position of the lanes with discretized y-positions and the same is annotated on the test set. To increase the

datasets and create a robust model, simple data augmentation methods like flip, translation, rotation, adding Gaussian noise, changing intensity, and adding shadow are applied to datasets.



Figure 5. Final Result Visualization after Curve Fitting and Inverse Projection Mapping

3.2. Kaiming He Weight Initialization

The random weights derived from Gaussian distributions are used to initialize the deep CNNs models (Krizhevsky *et al.*, 2014). It is complicated to coverage the very deep models (Simonyan *et al.*, 2014) (e.g., >8 conv layers) with fixed standard deviations (e.g., 0.01 in (GitHub, 2018)), and it is observed in our experiments. Hence the Kaiming He weight initialization was implemented in this work which is a theoretically sound initialization method. It uses ReLU activation which helps in faster convergence when it is trained from scratch giving more flexibility to explore powerful deep learning network architectures (He *et al.*, 2015).

3.3. Training

The weight initialization for the encoder and decoder are done using the same approach reported by He *et al.*, 2015, Stochastic gradient descent (SGD) was used (Bottou, 2010) with a fixed learning rate of 0.001 and a momentum of 0.09 to train all the variants. The entire work is trained using the Pytorch until the training loss converges. The training set shuffled, one mini-batch (1 image) was picked to ensure that each image is used only once in every epoch as shown in Figure 6.

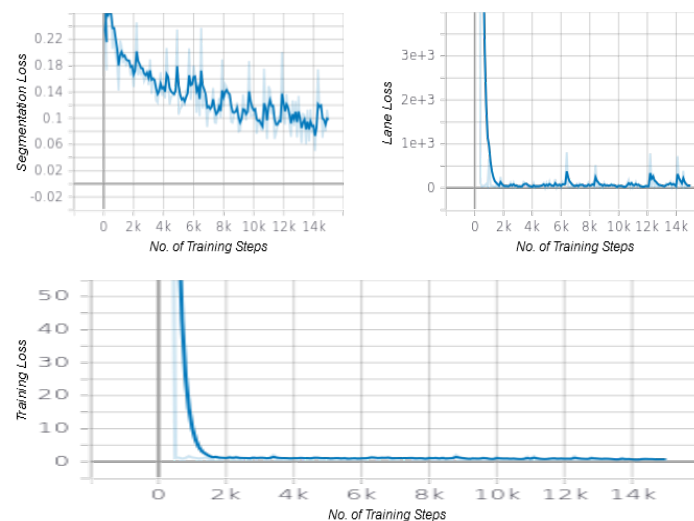


Figure 6. Results of Training Process

3.4 Evaluation Metrics

It required to fix the y-axis value and pick the corresponding x-axis value by applying linear interpolation to evaluate the TUsimple dataset. The complex curve fitting methods are not necessary as the distance between predicted points are very close. The root mean square error (RMSE) is used as an evaluation metric to evaluate the accuracy of the proposed approach as it is more suitable for calculating the errors between the predicted lane and ground truth. The RMSE for the detected lane using the proposed algorithm and the ground truth values is defined by,

$$e = \frac{\sum_{j=0}^M (d(t_i - p_i))}{\sqrt{M}}, \quad (5)$$

where t_i is the target value, p_i is the predicted value, d is distance and M is the number of samples. The mean RMSE value observed from the simulation results is *15.922 pixels* over the test set. On each image, the current (ego) left/right lanes are annotated, and this is also expected on the test set. The accuracy (γ) of lane detection is defined as ratio between average number of correct points per image (c_{image}) and number of ground-truth points per image (s_{image}):

$$\gamma = \frac{\sum_{image} c_{image}}{s_{image}}. \quad (6)$$

A point is correct when the difference between a ground-truth and predicted point is less than a certain threshold. The false positive (F_p) is as ration between total number of wrongly predicted lane (P_f) and total number of lanes predicted (P_r)

$$F_p = \frac{P_f}{P_r}. \quad (7)$$

The false negative (F_n) is as ration between total number of missed ground-truth lanes (P_{mgt}) and the total number of ground-truth lanes (P_{gt})

$$F_n = \frac{P_{mgt}}{P_{gt}}. \quad (8)$$

The results and comparisons of our experiments are given in Table 1.

Table 1. Lane detection performance on the TUsimple test set.

Method	γ	F_p	F_n
PolyLaneNet (Lucas <i>et al.</i> , 2020)	93.36%	0.0942	0.0933
SCNN (Pan <i>et al.</i> , 2018)	96.53%	0.0617	0.0180
Muthalagu <i>et al.</i> , 2020	95.75%	-	-
Muthalagu <i>et al.</i> , 2021	96.89%	0.0265	0.0257
Proposed approach	94.63%	0.0245	0.0169

3.5. Results and Time Analysis of the Proposed Algorithm

The input images are 1280×720 pixels (HD quality) resolution are trained on a computer with Intel Core i5 (8th generation) processor with 32 GB of RAM and NVIDIA RTX 2060 GPU. The training time and inference in seconds were noted for each testing case. The proposed algorithm was effective as it took only 29.11 ms on average for post processing and 4.72 ms for inference averaging to 34.35 fps in real-world situations. The processing time of our proposed system is given in Table 2. The original input image to proposed algorithm and lane predicted output from the proposed algorithm are shown in Figure 7 and Figure 8.

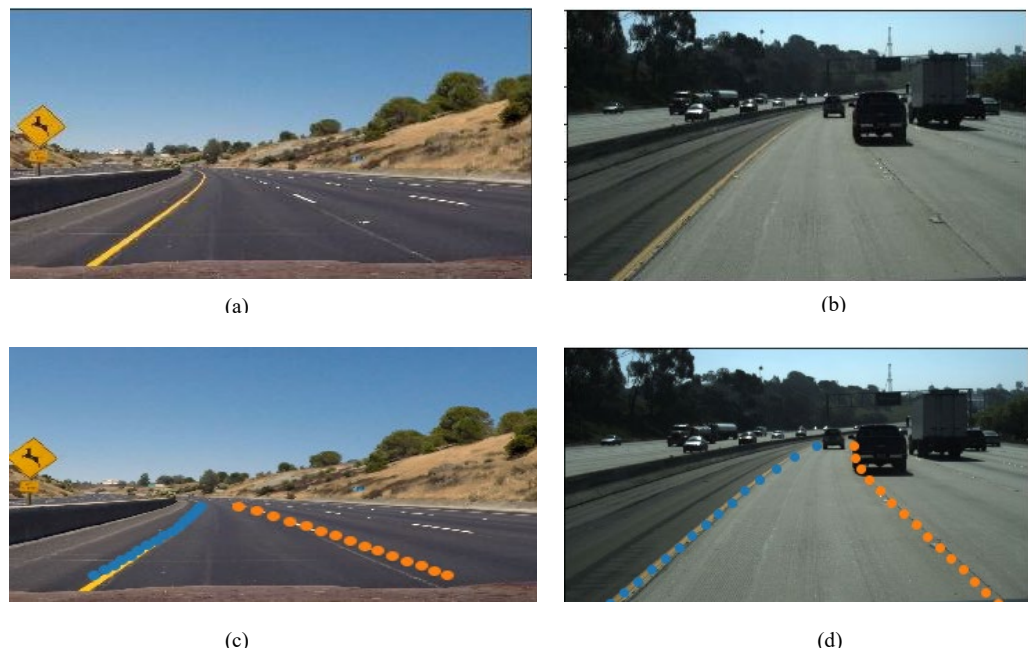


Figure 7. (a)-(b) Raw Input image to the proposed lane detection model (c)-(d) final lane predicted output from proposed lane detection model after lane fitting

Table 2. Processing Time Analysis of the Proposed System (in Milliseconds)

Perspective Transformation	1.94
Finding Lanes	4.72
Curve Fitting and Projection Remapping	22.45
Total	29.11

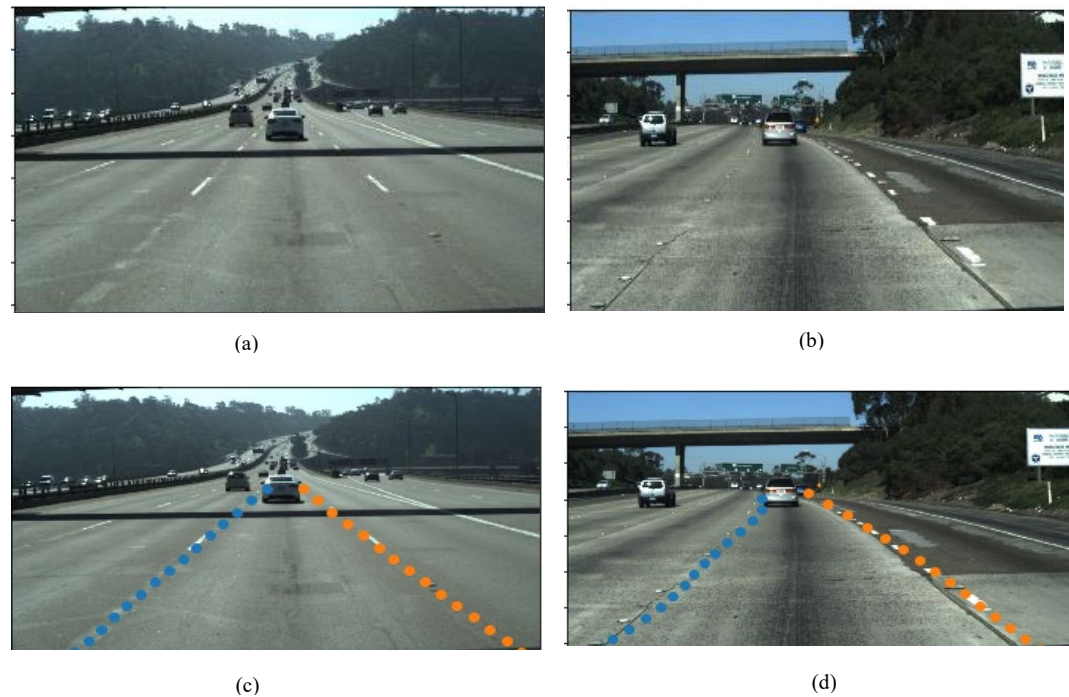


Figure 8. (a)-(b) Raw Input image to the proposed lane detection model (c)-(d) final lane predicted output from proposed lane detection model after lane fitting.

4. Conclusion

In this work, end-to-end lane detection using key point determination is implemented in this paper. A novel two-stage deep neural network is also proposed to detect the ego vehicle lanes in different environment situations to overcome the drawbacks of all the legacy computer vision approaches and single-stage segmentation techniques. The network trained uses low order polynomials and generates the key points of a lane from a perspective transformed lane image where lane fitting was optimal. The overall study found that it can run at 35 fps in real-world situations, taking 4.72 ms for lane detection. In the future, the accuracy of lane prediction can be improved by using a generative adversarial training framework. Also, methods like recurrent neural networks can be used to improve the key-points determination network focusing on improving the model's performance on video data to identify lane markings.

Acknowledgements

The authors like to thank Birla Institute of Technology and Science, Pilani Dubai campus for their constant support in resources and in research.

References

1. Singh, S. (2015) Critical reasons for crashes investigated in the national motor vehicle crash causation survey (No. DOT HS 812 115). *Washington, DC: National Highway Traffic Safety Administration*.
2. Bellis, E., & Page, J. (2008) *National motor vehicle crash causation survey (NMVCCS) SAS analytical users manual* (No. HS-811 053).
3. Gayko, J. E. (2012) Lane departure and lane keeping. *Handbook of intelligent vehicles*, 1, 689-708.
4. Visvikis, C., Smith, T. L., Pitcher, M., & Smith, R. (2008) Study on lane departure warning and lane change assistant systems. *Transport Research Laboratory Project Rpt PPR*, 374, 1-128.
5. Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., & Van Gool, L. (2018) Towards end-to-end lane detection: an instance segmentation approach. In: *2018 IEEE intelligent vehicles symposium (IV)*, Changshu, China, June 2018, 286-291.
6. Yan, X., & Li, Y. A method of lane edge detection based on Canny algorithm. In: *2017 Chinese Automation Congress (CAC)*, China, October 2017, pp. 2120-2124.
7. Oliveira, M., Santos, V., & Sappa, A. D. (2015) Multimodal inverse perspective mapping. *Information Fusion, Information Fusion*, 24, 108-121.
8. Muthalagu, R., Bolimera, A., and Kalaichelvi, V. (2021) Vehicle lane markings segmentation and keypoint determination using deep convolutional neural networks. *Multimedia Tools and Applications*, 80(7), 11201-11215.
9. Amaradi, P., Sriramoju, N., Dang, L., Tewolde, G.S. & Kwon, J. (2016) Lane following and obstacle detection techniques in autonomous driving vehicles. In: *2016 IEEE International Conference on Electro Information Technology (EIT)*, Grand Forks, North Dakota, May 2016, pp. 0674-0679.
10. Xing, Y. *et al.* (2018) Advances in Vision-Based Lane Detection: Algorithms, Integration, Assessment, and Perspectives on ACP-Based Parallel Vision, in *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 645-661, May 2018, doi: 10.1109/JAS.2018.7511063.
11. Wang, J. Gao & Y. Yuan (2018) Embedding Structured Contour and Location Prior in Siamesed Fully Convolutional Networks for Road Detection, *IEEE Transactions on Intelligent Transportation Systems*, 19(1), 230-241, doi: 10.1109/TITS.2017.2749964.
12. Ronneberger, O., Fischer P., & Brox T. (2015) U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*, Munich, Germany, October 2015, pp. 234-241. Springer, Cham.
13. Shelhamer, E., Long, J. & Darrell, T. (2017) Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 640-651. doi: 10.1109/TPAMI.2016.2572683.
14. Badrinarayanan, V., Kendall, A. & Cipolla, R. (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12), 2481-2495.
15. He, K., Zhang, X., Ren, S. & Sun, J. (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, June 2016, 770-778.

16. The tuSimple lane challenge, github. (2018). *GitHub*. Retrieved from <http://benchmark.tusimple.ai/>
17. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017) Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.
18. Simonyan, K. & Zisserman, A. (2014) Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*. 1-14.
19. He, K., Zhang, X., Ren, S. & Sun, J. (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, December 2015, pp. 1026-1034.
20. Bottou L. (2010) Largescale machine learning with stochastic gradient descent. In: *Proceedings of COMPSTAT 2010*, Paris, France, August 2010, pp. 177–186, Springer.
21. Lucas, T., Rodrigo, B., Thiago, M. P., Claudine, B., Alberto, F. De Souza, & Thiago, O-S. (2020) PolyLaneNet: Lane Estimation via Deep Polynomial Regression. *arXiv preprint arXiv: 2004.10924*. 1-7.
22. Pan, X., Shi, J., Luo, P., Wang, X., & Tang, X. (2018) Spatial as deep: Spatial CNN for traffic scene understanding. In: *Association for the Advancement of Artificial Intelligence*, New Orleans, Louisiana USA, February 2018, pp 1-8.
23. Muthalagu, R., Bolimera, A. & Kalaichelvi, V. (2020) Lane detection technique based on perspective transformation and histogram analysis for self-driving cars. *Computers & Electrical Engineering*, Elsevier, 85, 106653, 1-17.