Effects of Applying Identified Road Lane Lines on Vehicle Autopilot Model Driving Performance

Vjekoslav Diklić*, Franjo Matković†, Krešimir Pardon‡
*, ‡ dSPACE Engineering d.o.o., Zagreb, Croatia
† Fakultet elektrotehnike i računarstva, Zagreb, Croatia
e-mail address: vdiklic@dspace.hr franjo.matkovic@fer.hr kpardon@dspace.hr

***Abstract*—Autonomous Vehicle driving is one of the most actively developed systems in the automotive industry. Producing a reliable autonomous driving system is a complex multistage problem. The road lane lines are traffic signs with high level of semantics that determine how a vehicle should move, what actions are allowed to be performed, etc. Comparing different modifications of the PilotNet neural network model, the usage of detected lane lines, and their implications on vehicle autopilot driving performance is the topic of this paper. The lane lines are identified by neural network that has been trained by processing images with previously marked lane lines from CULane dataset. Vehicle autopilot driving performance is analyzed using autopilot model running on Euro Truck Simulator 2. Proposed models are analyzed and compared in multiple driving and weather environments including highway, off-road and urban driving during day, night and rain conditions. The lane line recognition (classified by a U-Net neural network model from images) is compared with additional control and path planning to determine whether it leads to better autonomous driving performance and smaller driving system model.**

Keywords*— *Autonomous Driving, Autopilot, Convolutional Neural Networks, Euro Truck Simulator 2, Image Processing, Virtual Controller

I. INTRODUCTION

Autonomous vehicle driving is one of the most actively developed systems in the automotive industry. The initiative comes from both industry and science as high-performance computing and artificial intelligence technology become cheaper, widely available and more advanced. Such systems offer increased comfort, driving experience and safety, while reducing road accidents. Based on the degree of automation, there are six levels of autonomous driving, which are: level zero (no automation at all), level one (very light automation; cruise control, etc.), level two (some automation but requires continuous human attention), level three (self driving with required intervention in severe conditions), level four (highly autonomous), level five (completely autonomous). Road lane lines are traffic signs with high level of semantics, since they determine vehicles movement and actions that are allowed to be performed. This makes lane lines critical component in autonomous driving as they describe the path for self driving cars and ensure the vehicle will not unpredictably change lanes. Comparing different modifications of PilotNet neural network model [1], usage of detected lane lines based on LaneUNet model [2], [3], and its implications on vehicle autopilot driving performance is the topic of this paper. The remainder of this paper is organized as follows: section II gives a brief overview of previous and current work on lane detection and autonomous driving pilots. Section III provides description of the problem being addressed, along with the algorithms used and the formation of the dataset. The training process, evaluation and results are displayed in Section IV. Section V concludes this paper.

II. RELATED WORK

Producing a reliable autonomous vehicle driving system is complex multistage problem, and critical part of it is lane detection. The following papers are dealing with different approaches to the problem of visual road lane lines detection.

Zheng et al. [4] proposed a Cross Laxer Refinement Network (CLRNet) for detecting lines by utilizing both high- and low-level semantic features in the images which correlate to the drive lanes.

Dong et al. [5] proposed a hybrid spatio-temporal deep learning network for continuous lane detection based on multiple sequential frames. The described working theory gives that lane lines can be better detected if past knowledge is used by sending multiple frames in the sequence, and the lane is detected only on the last frame. The authors have shown through experiments that their model outperforms other state-of-the-art models by utilizing single frame feature extraction module combined with spatio-temporal recurrent network model feature integrator and coder-decoder structure, which enabled it to be end-to-end learned.

Liu et al. [6] proposed a new approach to tackle lane detection in scenarios with complex topologies. Proposed approach is a top-to-bottom lane detection method that first detects line instances, then predicts lane shape for every detected instance. A conditional lane detection strategy based on conditional convolution and row-wise formulation is introduced to resolve lane instance-level discrimination problem.

Qiu et al. [7] introduced a transformer network to the road lane detection problem. The authors proposed a novel framework called PriorLane, which enhances segmentation of the fully vision transformer by introducing a low-cost prior knowledge. Results on their Zjlab dataset showed that the proposed method outperforms other SOTA methods.

Jin et al. [8] used a novel method to detect road lanes in eigenlane space. First, authors have introduced Eigenlanes, which are data-driven lane descriptors for structurally diverse lines (like curves and straight lines). Then, by obtaining best candidate lanes through approximation, authors have used a detection neural network to find optimal lanes.

In [9], He et al. proposed a generative adversarial network to find an enhanced feature space where lane features are distinctive while maintaining a similar distribution of road lanes in the wild. Novel Repainting and Imitating Learning (RIL) framework experiments proved effectiveness both on CULane and TuSimple for road lane detection methods.

Borjanski et al. [10] proposed a convolutional neural network (CNN) for learning autonomous driving from raw pixels of a single front-facing camera by mapping them into steering commands.

Chitta et al. [11] proposed a mechanism for integrating image and LIDAR representations using self-attention. This approach uses transformers at multiple resolutions to fuse perspective view with the bird's eye view feature maps. The motivation for the approach was under-performance of the end-to-end learning in complex driving scenarios with heavy traffic and dynamic agents high-density.

Chekroun et al. [12] presented a novel method called General Reinforced Imitation (GRI), which combines benefits from exploration and expert data. The authors proposed a simplifying hypothesis, that expert demonstrations can be seen as perfect data whose underlying policy receives a constant high reward. GRI combines offline demonstration agents and online RL exploration agents.

Hao et al. [13] introduced a novel safety-enhanced autonomous driving framework based on transformers called the Interpretable Sensor Fusion Transformer (InterFuser). The framework fuses information from multi-modal multi-view sensors for comprehensive scene recognition and adversarial event detection. The framework also generates intermediate interpretable features that provide more semantics that are exploited to better constrain actions.

Wu et al. [14] introduced a novel two-branch method for autonomous driving. Branches are based on trajectory planning and direct control of vehicle. While trajectory planning branch predicts the future trajectory, the control branch receives the corresponding guidance from the trajectory branch. The control branch makes a multi-step prediction scheme in the way the relationship between current action and future states can be reasoned. Outputs from the branches are fused together to an achieve advantage from each branch.

Toromanoff et al. [15] presented a novel technique that used a reinforcement learning (RL) approach to autonomous driving. It contains coined implicit affordances to leverage RL for urban driving, which include lane keeping, pedestrian and vehicles avoidance and traffic light detection. The authors have successfully managed to present an RL agent which is capable of handling traffic light detection.

III. The proposed approach

This paper compares how pre-marked road lane lines impact autonomous driving system when the system is trained with and without pre-marked lines. The original authors of PilotNet [16], [17] developed a deep learning neural network for autonomous driving and stated that the deep learning network will perform better due to internal components self-optimizing to maximize overall system performance if it were given no human-selected intermediate optimizing criteria, such as lane line detection. Such criteria are understandably selected for ease of human interpretation, but this doesn't automatically guarantee maximum system performance. This paper examines the influence of human interpretable data as additional information provided to deep neural networks. The Europilot [18] project is an interface between Euro Truck Simulator 2 (ETS2) [19] and Python code that was used to investigate the impact of the pre-marked lane lines (lane detection) on a driving performance of an PilotNet model. The autopilot model uses only raw pixels from simulator as input, extended with pre-marked lane lines generated by LaneUNet.

A. Self-driving algorithms

Authors [1] argue that providing the neural network with raw data is the best way to create a robust self-driving system which can navigate its way even on unmarked lanes (unpaved roads, parking lots):

> "The system automatically learns internal representations of the necessary processing steps such as detecting useful road features with only the human steering angle as the training signal. We never explicitly trained it to detect, for example, the outline of roads. Compared to explicit decomposition of the problem, such as lane marking detection, path planning, and control, our end-to-end system optimizes all processing steps simultaneously. We argue that this will eventually lead to better performance and smaller systems. Better performance will result because the internal components self-optimize to maximize overall system performance, instead of optimizing human-selected intermediate criteria, e.g., lane detection. Such criteria understandably are selected for ease of human interpretation which doesn't automatically guarantee maximum system performance."

PilotNet CNN model was used in [1], [18] for autonomous driving system only on captured frames of Euro Truck

TABLE I: PilotNet architecture

Layer type	Layer description
Input planes	200 × 200 RGB
2D CNN layer	24 filters, kernel_size=(5, 5), strides=(2, 2) Normalization layer
2D CNN layer	36 filters, kernel_size=(5, 5), strides=(2, 2) Normalization layer
2D CNN layer	48 filters, kernel_size=(5, 5), strides=(2, 2) Normalization layer
2D CNN layer	64 filters, kernel_size=(3, 3), strides=(1, 1) Normalization layer
2D CNN layer	64 filters, kernel_size=(3, 3), strides=(1, 1) Normalization layer
FC layer	100 neurons Normalization layer
FC layer	50 neurons Normalization layer
FC layer	10 neurons Normalization layer
FC layer	1 neuron Normalization layer

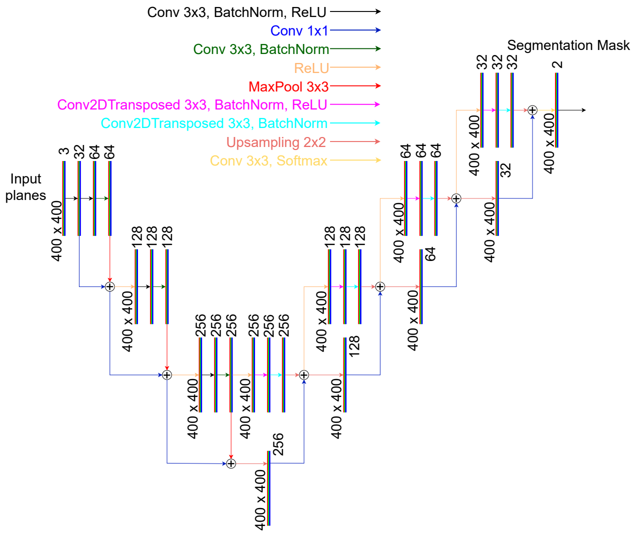


Fig. 1: LaneUNet structure

Simulator 2. The model architecture is composed of following layers as shown in Table I.

B. Lane Lines Detection

A road lane is a portion of roadway allocated to a single line of vehicles and their movement to control and guide drivers in order to reduce traffic confusion. It is indicated by painted longitudinal lines or embedded markings on roadway pavement, along with road surface markings such as lane direction indicators. Lane lines detection is one of the critical detection problems in the vision navigation systems of the intelligent vehicles [4]. It is a traffic sign with high-level semantics which determines how a vehicle should move and what actions it should perform in traffic (such as switching lanes if the line is dashed or not switching/overtaking if the lane is solid or double solid, etc.).

1) Lane Lines Detection Model: Lane line classification from an image was done by a U-Net-like model [2], [3], called LaneUNet. LaneUNet consists of 4 different stages: downscale, encoder, decoder, and output generator, as shown in Figure 1. The model is configured to work with images with a resolution of (400×400) pixels, with two classes: Lane and Not Lane. The chosen training optimizer for this model was RMSprop, with the loss function defined as categorical cross-entropy.

2) PilotNet model and modifications: The PilotNet [1] model that was inherited from Europilot project [18], has five convolutional layers followed by three fully connected layers. The optimizer used for learning PilotNet is stochastic gradient descent (SGD), while the loss function is mean square error (MSE) in reference to the angle of the recorded steering wheel data The LaneUNet model, with frozen internal layer weights, was added to input layer of PilotNet in three different ways:

- Direct input (1 Channel PilotNet), was made for control purposes as proof that lane lines are not reference data for driving and has smallest size of input data $400 \times 400 \times 1$.
- Overlayed in green channel (3 Channel PilotNet), was introduced as a way to include lane lines without increasing size of input data.
- Additional channel of input (4 Channel PilotNet), adds lane lines as additional channel and has biggest size of input data $400 \times 400 \times 4$.

The original PilotNet model was included for comparison as reference with an input data size as $400 \times 400 \times 3$. A complete overview of all aforementioned modifications done to PilotNet model can be seen in Figure 2.

C. Dataset generation

Both the LaneUNet and PilotNet models requires data for training, since pretrained models were not used. For LaneUNet, a modified and prepared dataset was used, while PilotNet training dataset were recorded.

1) LaneUNet Training Dataset: LaneUNet training data is based on CULane [20] dataset and extended with data augmentation. CULane lane line was plotted with a line width of 13 pixels. Data augmentation was done to increase the size of dataset inspired by the work of Mamun et al. [21] and to increase robustness of the model. Every image in the dataset was kept as original, mirrored randomly by up/down or left/right, randomly cropped, and randomly perspective cropped, also brightness was changed by a random value. The complete number of frames after combining original CULane dataset and augmented frames was 668021 frames. Learning and validation subsets were divided from this combined number of frames in an $(80 : 20)$ ratio.

2) PilotNet Training Dataset: Data for training PilotNet models was recorded using ETS2 and a steering wheel Controller (Thrustmaster F430) by utilizing Europilot [18] data generation script. While recording data, the ETS2 internal resolution was set to 1152×864 and the graphical

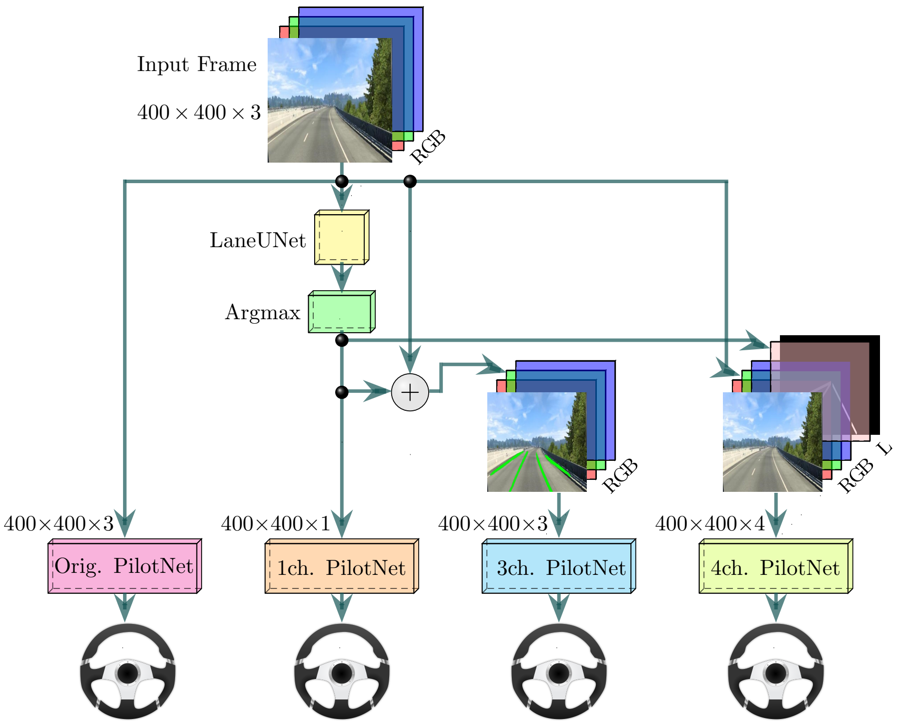


Fig. 2: PilotNet Modifications Overview

TABLE II: Rough estimate: recorded data locations and conditions

Condition / Location	Day& Clear	Day& Rain	Night& Clear
Urban	7.86%	0%	5.96%
Offroad	3.46%	0%	1.41%
Contry road	1.12%	0%	0.91%
Highway	34.48&	6.21%	38.64%

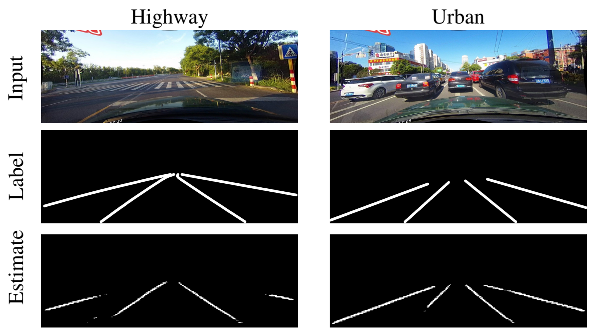
level of details was set to "High" in order to make recorded frames close to real-life frames of CULane dataset. The recorded data consists of 249190 frames captured at rate of 25 frames per second (fps) which equals of 2 hours and 46 minutes. Dataset was divided into training and validation subsets by $80 : 20$ ration. Recorded data consist of multiple driving conditions as shown in Table II. After recording, frames were cropped to include only road view out of the vehicle cabin as suggested in the Europilot [18] project post processing script. After recording, frames were cropped to include only road view out of vehicle cabin as suggested in Europilot [18] project post processing script . Captured steering wheel data contains three values: steering angle, brake and acceleration pedal position. For the purpose of this paper only steering angle values were used in order to simplify the autonomous driving problem.

IV. Experiments

A. *LaneUNet Evaluation*

Examples for the evaluation of final LaneUNet model after 8935 epochs with an evaluation loss of 0.0598 and a validation accuracy of 97.94229% on a validation part of dataset is shown in Table III. Additional evaluation results of the model on ETS2 frames are shown in Table IV. It is obvious that model was not able to detect lane lines in off-road scenarios. This comes from the fact that CULane does not include frames that are similar to off-road scenarios from ETS2.

TABLE III: Evaluation Example: LaneUNet on CULane



B. *PilotNet evaluation*

PilotNet and its modifications were trained for 320 epochs. Neural network models for evaluation in the simulator were chosen based on the best results of validation loss and accuracy value shown in Table V, and additionally for each model epochs 100, 200 and 320 were also included. These models were chosen as a uniform distribution through learning epochs. This totals 20 models for live evaluation tests in ETS2. Due to limitations of ETS2, evaluation tests of each neural model were done

TABLE IV: Evaluation Example: LaneUNet on ETS2

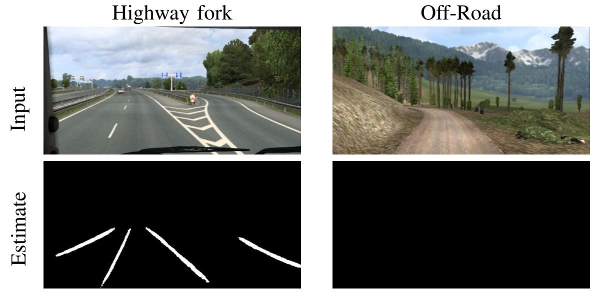


TABLE V: PilotNet Models Training results

Model	Value	Loss	Accuracy	Validation Loss	Validation Accuracy
Original PilotNet	epoch	305	305	315*	307**
	value	1.13104	0.21225	0.75563	0.26586
1 Channel PilotNet	epoch	170	1	183*	1**
	value	59.70996	0.01470	58.38287	0.0
3 Channel PilotNet	epoch	320	300	243*	308*
	value	0.21720	0.27705	0.54382	0.27416
4 Channel PilotNet	epoch	320	315	276*	284**
	value	0.29463	0.27160	0.57606	0.27568

by accelerating the vehicle to specific velocity, engaging the built in ETS2 speed cruise control and then activating the neural model for steering input through a virtual steering wheel controller. The evaluation test is manually stopped either when the vehicle hits roadside or it drives off road. There are 4 evaluation locations: Urban, Off-Road, Country Road, Highway. Due to the nature of the simulation, each test was run only once in identical initial conditions and yielded consistent results when repeated. Results of evaluation are shown in Tables VI, VII, VIII and IX. The displayed results show PilotNet models that have LaneUNet pre-marked lane lines in preprocessing have better driving performances. Time value in the table represents the time that model under test is able to drive before hitting a road barrier or making a dangerous action. The result in tables shows the best performing model on Urban location for Day&Rain is 3 Channel PilotNet. For Night&Clear shows that 4 Channel PilotNet is the best. For Night&Rain the best performing network is 4 Channel PilotNet, even though this condition is not present in learning dataset. Original PilotNet is the best performing in Day&Clear conditions.

The bast performing model for Off-Road location is 4 Channel PilotNet, except for Night&Rain condition where 3 Channel PilotNet is slightly better. Original PilotNet has performance values close to the best model in Day&Rain and Night&Rain conditions. 4 Channel PilotNet model is the best in Country Road location at Day&Rain and Night&Clear conditions, while 3 Channel PilotNet is better at Day&Clear and Night&Rain conditions.

The highway location is dominated by 4 Channel PilotNet model except for the Day&Rain condition where the Original PilotNet model is slightly better.

TABLE VI: PilotNet Models Evaluation Results, Location: Urban

Location: **Urban**	Conditions:	Day & Clear	Day & Rain	Night & Clear	Night & Rain
Model	Epoch	Time [s]			
Original PilotNet	100	3.83834	2.71345	10.96311	14.28656
	200	11.23782	2.96604	11.00926	14.27493
	320	4.09575	2.71534	10.96144	26.98413
	307**	**16.23458**	16.55962	14.14848	14.38183
	315*	4.35174	16.51747	14.09681	14.34698
3 Channel PilotNet	100	7.13	3.43547	10.19311	11.28226
	200	9.24653	3.65635	11.05906	11.22218
	320	7.46921	3.26165	14.05149	10.92544
	243*	9.2222	**17.83614**	10.09567	10.61436
	308**	9.008	12.51868	14.15266	11.36733
4 Channel PilotNet	100	10.88346	4.5254	11.0694	**45.60521**
	200	10.67793	12.44566	**18.45405**	30.41055
	320	3.12718	7.26811	11.49798	17.51122
	276*	3.89543	10.67699	11.26646	17.54752
	284**	2.98151	9.71796	11.06521	17.72489

TABLE VII: PilotNet Models Evaluation Results, Location: Off-Road

Location: **Off-Road**	Conditions:	Day & Clear	Day & Rain	Night & Clear	Night & Rain
Model	Epoch	Time [s]			
Original PilotNet	100	54.20072	53.19633	6.55412	7.84882
	200	57.95325	92.22533	9.05661	21.15158
	320	53.32286	12.84984	6.95825	20.03792
	307**	53.11098	53.8781	3.49425	15.174
	315*	52.54838	54.28565	3.39048	12.439
3 Channel PilotNet	100	52.34927	53.25124	12.75766	19.81261
	200	16.60616	52.3291	21.84162	**21.41379**
	320	19.10798	47.94633	19.51536	20.50395
	243*	50.51435	45.65567	18.86555	21.32946
	308**	19.47457	53.00088	21.68742	21.01616
4 Channel PilotNet	100	7.75637	53.66708	**21.86947**	17.89866
	200	**133.53286**	38.02302	25.09455	3.60717
	320	52.75734	91.98988	21.74667	3.98164
	276*	57.83613	**92.43867**	6.53626	4.50465
	284**	14.25077	3.34132	21.65484	3.69087

V. Conclusion

Comparing different modifications of PilotNet neural network model [1], usage of detected lane lines based on LaneUNet model [2], [3], and its implications on vehicle autopilot driving performance is the topic of this paper.

The first modification is based on sending the results of LaneUNet directly to the PilotNet model, replacing its original input. The second modification overlays lane lines detection results with the green channel of raw input frames from ETS2. The third modification adds extra channel to the input data of PilotNet which consists of LaneUNet detection results on ETS2 raw frames. The Original PilotNet model is used as reference during training and evaluation. Overall, the best result is obtained by the third modification of PilotNet, followed by the second modification, the Original PilotNet and lastly the first modification. ETS2 is used to train and validate PilotNet

TABLE VIII: Evaluation Results, Location: Country Road

Location: **Country Road**	Conditions:	Day & Clear	Day & Rain	Night & Clear	Night & Rain
Model	Epoch	Time [s]			
Original PilotNet	100	93.59565	93.66441	35.67297	9.7689
	200	94.48444	91.17316	47.53247	9.88838
	320	95.95082	86.93309	47.17973	38.77904
	307**	96.15499	54.81467	47.33901	48.01871
	315*	49.66787	95.80688	47.79666	48.64261
3 Channel PilotNet	100	**111.42274**	93.6237	38.54933	12.17085
	200	93.86836	98.7547	38.62285	87.21712
	320	16.3591	98.97377	39.31017	**88.75275**
	243*	92.71974	98.42113	38.49544	18.61541
	308**	99.97491	98.56087	37.84902	85.84725
4 Channel PilotNet	100	53.6726	**103.0861**	**68.14147**	75.26335
	200	60.62461	79.25365	67.89574	67.955
	320	55.4819	92.57005	14.27094	67.53863
	276*	60.73591	85.66176	69.02957	31.79299
	284**	60.68326	95.33997	67.81089	68.2345

TABLE IX: PilotNet Models Evaluation Results, Location: Highway

Location: **Highway**	Conditions:	Day & Clear	Day & Rain	Night & Clear	Night & Rain
Model	Epoch	Time [s]			
Original PilotNet	100	22.89169	2.33809	41.45732	22.26164
	200	23.93349	2.86061	41.88455	30.73024
	320	24.81868	**5.97896**	41.3217	31.16824
	307**	7.82739	2.56142	41.61631	9.56341
	315*	24.05545	2.43687	41.03164	30.54046
3 Channel PilotNet	100	2.69504	2.56999	23.25907	25.09734
	200	2.71988	2.59072	29.00045	24.95291
	320	2.00677	2.36893	59.41756	25.29419
	243*	4.23775	2.22972	59.00256	23.57244
	308**	2.25181	2.46994	59.54084	24.61422
4 Channel PilotNet	100	2.48298	2.87298	62.71282	**31.1772**
	200	14.85202	5.44053	41.97563	4.46748
	320	11.49844	3.03276	26.73725	3.47817
	276*	14.99146	3.66295	**89.29299**	5.14221
	284**	**60.05419**	3.14683	10.62356	6.1917

models on four different locations. The LaneUNet model, used for lane line detection, is trained on augmented CULane dataset.

In future work, both models for lane line detection and PilotNet can be improved by including a temporal component such as LSTM or GRU. The learning dataset for the PilotNet model can be further extended by diversifying locations and conditions. Also, instructions for lane line changes and pedal controls (acceleration, braking) can be added.

REFERENCES

[1] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016. [Online]. Available: https://arxiv.org/abs/1604.07316

[2] F. Chollet, Deep Learning with Python, Second Edition. Manning, 2021. [Online]. Available: https://books.google.hr/books?id=mjVKEAAAQBAJ

[3] ——, Apr 2020. [Online]. Available: https://keras.io/examples/vision/oxford_pets_image_segmentation/

[4] T. Zheng, Y. Huang, Y. Liu, W. Tang, Z. Yang, D. Cai, and X. He, "Clrnet: Cross layer refinement network for lane detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 898–907.

[5] Y. Dong, S. Patil, B. van Arem, and H. Farah, "A hybrid spatial–temporal deep learning architecture for lane detection," Computer-Aided Civil and Infrastructure Engineering, vol. 38, no. 1, pp. 67–86, feb 2022. [Online]. Available: https://doi.org/10.1111%2Fmice.12829

[6] L. Liu, X. Chen, S. Zhu, and P. Tan, "Condlanenet: a top-to-down lane detection framework based on conditional convolution," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3753–3762, 2021.

[7] Q. Qiu, H. Gao, W. Hua, G. Huang, and X. He, "Priorlane: A prior knowledge enhanced lane detection approach based on transformer," 2022. [Online]. Available: https://arxiv.org/abs/2209.06994

[8] D. Jin, W. Park, S.-G. Jeong, H. Kwon, and C.-S. Kim, "Eigenlanes: Data-driven lane descriptors for structurally diverse lanes," 2022. [Online]. Available: https://arxiv.org/abs/2203.15302

[9] Y. He, M. Jiang, X. Ye, L. Du, Z. Zou, W. Zhang, X. Tan, and E. Ding, "Repainting and imitating learning for lane detection," 2022. [Online]. Available: https://arxiv.org/abs/2210.05097

[10] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016. [Online]. Available: https://arxiv.org/abs/1604.07316

[11] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," 2022. [Online]. Available: https://arxiv.org/abs/2205.15997

[12] R. Chekroun, M. Toromanoff, S. Hornauer, and F. Moutarde, "Gri: General reinforced imitation and its application to vision-based autonomous driving," 2021. [Online]. Available: https://arxiv.org/abs/2111.08575

[13] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," 2022. [Online]. Available: https://arxiv.org/abs/2207.14024

[14] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," 2022. [Online]. Available: https://arxiv.org/abs/2206.08129

[15] M. Toromanoff, E. Wirbel, and F. Moutarde, "End-to-end model-free reinforcement learning for urban driving using implicit affordances," 2019. [Online]. Available: https://arxiv.org/abs/1911.10868

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

[17] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," arXiv preprint arXiv:1704.07911, 2017.

[18] MARSAUTO, "Europilot." [Online]. Available: https://github.com/marsauto/europilot

[19] SCS Software, "Euro truck simulator 2," DVD,Steam, 2012.

[20] P. Xingang, S. Jianping, L. Ping, W. Xiaogang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in AAAI Conference on Artificial Intelligence (AAAI), February 2018.

[21] A. A. Mamun, E. P. Ping, J. Hossen, A. Tahabilder, and B. Jahan, "A comprehensive review on lane marking detection using deep neural networks," Sensors, vol. 22, no. 19, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/19/7682