WIP: On Robustness of Lane Detection Models to Physical-World Adversarial Attacks

Takami Sato
University of California, Irvine
takamis@uci.edu

Qi Alfred Chen
University of California, Irvine
alfchen@uci.edu

***Abstract*—Deep Neural Network (DNN)-based lane detection is widely utilized in autonomous driving technologies. At the same time, recent studies demonstrate that adversarial attacks on lane detection can cause serious consequences on particular production-grade autonomous driving systems. However, the generality of the attacks, especially their effectiveness against other state-of-the-art lane detection approaches, has not been well studied. In this work, we report our progress on conducting the first large-scale empirical study to evaluate the robustness of 4 major types of lane detection methods under 3 types of physical-world adversarial attacks in end-to-end driving scenarios. We find that each lane detection method has different security characteristics, and in particular, some models are highly vulnerable to certain types of attack. Surprisingly, but probably not coincidentally, popular production lane centering systems properly select the lane detection approach which shows higher resistance to such attacks. In the near future, more and more automakers will include autonomous driving features in their products. We hope that our research will help as many automakers as possible to recognize the risks in choosing lane detection algorithms.**

I. Introduction

Lane detection is an essential technology for realizing autonomous driving. Like most other computer vision areas, lane detection has been significantly benefited from the recent advances of deep neural networks (DNNs) as camera is the most frequently used sensor [1]. In the 2017 TuSimple Lane Detection Challenge [2], DNN-based lane detection shows substantial performance as all top 3 teams opt for DNN-based lane detection. However, DNN-based approach has a well-known vulnerability against adversarial attacks [3], [4]. Recent works show that Automated Lane Centering (ALC) systems, Level-2 driving automation, are vulnerable to physical-world adversarial attacks such as malicious patches [5] and stickers [6]. In this work, we report our recent progress on conducting the first large-scale empirical study to evaluate the robustness of major lane detection methods against physical-world adversarial attacks in autonomous driving. These prior works are limited to showing the effectiveness of their attacks on particular lane detection methods. For example, DRP attack [5] is evaluated only on a curve fitting-based lane detection in OpenPilot [7]. The attack shown in [6] is demonstrated only on a segmentation-based lane detection in Tesla Model S. Thus, the effectiveness of attacks on other lane detection methods and the security properties of these lane detection models against adversarial attacks have not been well studied.

In this paper, We first taxonomize state-of-the-art DNN-based lane detection models into 4 major categories (§II-A) We then introduce state-of-the-art physical-world adversarial attacks against ALC systems (§II-B). In §III, we construct a methodology to fairly evaluate the robustness of the 4 major types of lane detection models in the end-to-end evaluation. To simulate end-to-end scenarios, we develop a bridge between lane detection methods and the vehicle lateral control implemented in OpenPilot [7], an open-source production ALC system. In §IV, we evaluate the robustness of 4 major types of lane detection approaches against 3 types of physical-world adversarial attacks by answering 3 research questions. Throughout this study, we find that each type of lane detection model has different security properties against adversarial attacks: several models are even vulnerable to a naive attack which just draws a white line on the road. Surprisingly, but probably not coincidentally, popular production ALC systems, Tesla Model S and OpenPilot [7], properly select the lane detection approach which shows higher resilience to the drawing-lane-line attack. We then discuss the conclusion and further directions of our study in §VI.

II. Background

A. DNN-based Lane Detection

We taxonomize state-of-the-art DNN-based lane detection methods into 4 approaches. Similar taxonomy is also adopted in prior works [8], [9].

Segmentation approach. Segmentation approach handles lane detection as a segmentation task, which classifies whether each pixel is on a lane line or not. Since this approach achieved the state-of-the-art performance in the 2017 TuSimple Lane Detection Challenge [2] (all top-3 winners adopt the segmentation approach [10], [11], [12]), it has been applied in many recent lane detection methods [13], [14], [15]. This segmentation approach is also used in the industry. A reverse-engineering study reveals that Tesla Model S adopts this segmentation-based approach [6]. The major drawback of this approach is its higher computational and memory cost than the other approaches. Due to the nature of the segmentation approach, it needs to predict the classification results for every pixel, the majority of which is just background. Additionally, this approach requires a postprocessing step to extract the lane line curves from the pixel-wise classification result.

Workshop on Automotive and Autonomous Vehicle Security (AutoSec) 2022
24 April 2022, San Diego, CA, USA
ISBN 1-891562-75-4
https://dx.doi.org/10.14722/autosec.2022.23037
www.ndss-symposium.org

Row-wise classification approach. This approach [16], [17], [18], [9] leverages the domain-specific knowledge that the lane lines should locate the longitudinal direction of driving vehicles and should not be so curved to have more than 2 intersections in each row of the input image. Based on the assumption, this approach formulates the lane detection task as multiple row-wise classification tasks, i.e., only one pixel per row should have a lane line. Although it still needs to output classification results for every pixel similar to the segmentation approach, this divide-and-conquer strategy enables to reduce the model size and computation while keeping high accuracy. For example, UltraFast[16] reports that their method can work at more than 300 FPS with a comparable accuracy 95.87% on the TuSimple Challenge dataset [2]. On the other hand, SAD [14], a segmentation approach, works at 75 frames per second with 96.64% accuracy. This approach also requires a postprocessing step to extract the lane lines similar to the segmentation approach.

Curve-fitting approach. The curve-fitting approach [19], [20] fits the lane lines into parametric curves (e.g., polynomials and splines). This approach is applied in an open-source production driver assistance system, OpenPilot [7]. The main advantage of this approach is lightweight computation, allowing OpenPilot to run on a smartphone-like device without GPU. To achieve high efficiency, the accuracy is generally not high as other approaches. Note that prior work mentions that this approach is biased toward straight lines because the majority of lane lines in the training data are straight [19].

Anchor-based approach. Anchor-based approach [8], [21], [22] is inspired by region-based object detectors such as Faster R-CNN [23]. In this approach, each lane line is represented as a straight proposal line (anchor) and lateral offsets from the proposal line. Similar to the row-wise classification approach, this approach takes advantage of the domain-specific knowledge that the lane lines are generally straight. This design enables to achieve state-of-the-art latency and performance. LaneATT [8] reports that it achieves a higher F1 score (96.77%) than the segmentation approaches (95.97%) [14], [10] on the TuSimple dataset.

B. Physical-world Attacks on Automated Lane Centering

After researchers found DNN models generally vulnerable to adversarial examples or adversarial attacks [3], [4], the following work further explored such attacks in the physical world [24], [25], [26]. Recent studies demonstrate that ALC systems, Level-2 driving automation, are also vulnerable to physical-world adversarial attacks.

Dirty Road Patch Attack [5]. Dirty Road Patch (DRP) attack is proposed as a domain-specific adversarial attack to DNN-based ALC systems [5]. DRP attack pretends to be a benign but dirty road patch. The dirty surface pattern is generated by a white-box optimization-based method to work as an adversarial example to lane detection models. To mimic a road patch, the DRP attack has stealthiness constraints such as the gray-scale color restriction and perturbable area ratio. While it has high attack success rates, DRP attack requires white-box access to the target system and relatively heavy deployment effort.

Drawing-Lane-Line Attack. As the nature of lane detection, drawing a line on the road can be an effective attack vector. A recent work [6] demonstrates that they can mislead Tesla Model S to the adjacent lane by putting several small stickers on the road without the original lane line. Phantom attack [27] also demonstrates that they can mislead Tesla Model S by projecting fake lane lines from a drone in the nighttime. The drawing-lane-line attack is not as effective as the DRP attack based on our experience, but its vulnerability to this attack is more severe because of its ease of deployability.

III. Methodology

To fairly evaluate the security properties of the 4 major types of lane detection approaches, we design evaluation methodology in end-to-end driving scenarios under 3 types of physical-world adversarial attacks.

A. Attack Implementation

We implemented 3 types of state-of-the-art physical-world adversarial attacks based on prior attacks against ALC systems discussed in II-B.

White-Box DRP Attack We implement the DRP attack [5]. While the original DRP attack uses the lane bending objective function, we apply a newly-designed attack objective discussed in §III-B to conduct a fair comparison with other attacks and to deal with the output space different from the original DRP attack, which outputs the detected lane lines in the bird's-eye view. All lane detection methods evaluated in this study detected lane lines in the driver's view.

Black-Box DRP Attack To make the DRP attack work in a black-box setup, we apply a query-based black-box attack approach [28] to extend the DRP attack to a black-box attack. We replace the gradient calculation in the original white-box DRP attack with the gradient estimation technique NES [28].

Black-Box Drawing-Lane-Line Attack We explore the most effective line with a metaheuristic strategy according to prior work [6]. We parameterize the drawing lane line as the start point, endpoint, and line width and optimize the parameters with the tree-structured Parzen estimator [29] implemented in Hyperopt [30]. As the objective of the original attack [6] is only applicable to the segmentation approach, we optimize our original attack objective introduced in §III-B to conduct a fair comparison with other attacks.

B. Attack Objective

To fairly evaluate the attack capability of each attack, we formulate an attack objective function that can be commonly used for all 4 types of lane detection models. We named it the *expected road center function*, which averages all detected lane lines weighted with their probabilities. Intuitively, the average of all lane lines is expected to represent the road center. If the expected center locates at the center of the input image, its value will be 0.5 in the normalized image width. We maximize the expected road center to attack to the right and minimize it to attack to the left. Detailed calculation of the expected road center for each method is in our preprint paper [31]. When attacking multiple frames, we average the objective of each frame over all attacking frames.

C. End-to-End Simulation

To evaluate the system-level consequence in autonomous driving, we simulate vehicle trajectories under attacks with the same methodology used in [5]. We combine a vehicle motion model [32] and perspective transformation [33], [34] to dynamically synthesize camera frame updates according to a driving trajectory. This approach enables us to evaluate the attacks on the real-world driving traces in a lightweight way. To control a vehicle based on the lane detection results, we develop a bridge between the lane detection model and the vehicle lateral control implemented in OpenPilot [7], an open-source production ALC system. It calculates the desired driving path based on detected lane lines and makes a steering plan to follow the desired driving path with Model Predictive Control (MPC) [35]. In our implementation, the desired driving path is the center of the left and right lane lines.

Attack Goal. To judge the attack success in the end-to-end simulation, we follow the criteria proposed in the DRP attack [5]. We use the attack goal achieving over 0.735 m lateral deviation on the highway within the average driver reaction, 2.5 sec. 0.735 m is the required distance to touch the lane line when a vehicle driving at the center of a 3.6m-wide highway lane. The lateral deviation is calculated between the generated trajectories with attack and without attack. Since the original human driving in the dataset sometimes does not drive at the center of the road, we compare the case with attack and without attack to more precisely measure the attack effect. For each scenario, we consider two attack success criteria: *Targeted goal* is the case that the vehicle deviates over 0.735 m to the attacking direction. *Untargeted goal* is the case that the vehicle deviates over 0.735 m to either the left or right.

We also quantify the ability to drive in a benign scenario. We define a metric called *benign failure rate*, which is whether the human driving and the simulated trajectory deviate by more than 0.735 m. Although the benign failure rate is expected to be always zero because ALC systems should be able to handle normal scenarios, some failure cases occur due to several reasons such as motion model inaccuracy and unstable human driving, e.g., not driving at the center of the road.

IV. Experiments

We conduct a large-scale evaluation of 4 major types of lane detection approaches against 3 adversarial attacks: white-box DRP, black-box DRP, black-box drawing-lane-line attacks. This evaluation is designed to answer the following questions:

RQ1: Is black-box attack as effective as white-box attack?
RQ2: Which attack vector is more effective to attack?
RQ3: Are attacks transferable to other models?

A. Evaluation Setup

We evaluate the robustness of 4 major types of lane detection approaches against 3 adversarial attacks: white-box DRP, black-box DRP, black-box drawing-lane-line attacks. For each approach, we select a representative model for each approach as shown in Table I with the selection reasons. The pretrained weights of all models are obtained from the authors' or publicly available websites[1]. All pretrained weights are trained on the TuSimple Challenge training dataset [2].

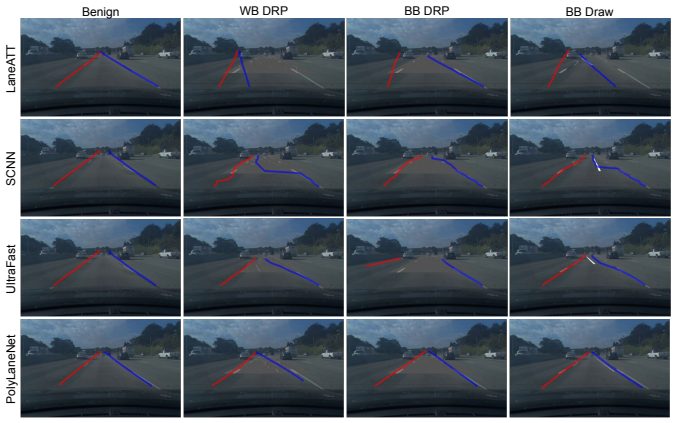

Fig. 1: Examples of the end-to-end benign and 3 different attack scenarios on Comma2k19. Each image is taken at the 4th frame (0.2 sec after the start of the attack). The red and blue lines are the detected left and right lines respectively.

TABLE I: Target lane detection methods and its selection reason. *Acc.* is the accuracy of the TuSimple Challenge dataset [2] in the reference papers.

Approach	Selected Method	Acc.	Selection Reason
Segmentaion	SCNN [10]	96.53%	TuSimple Challenge winner's model
Row-wise classif.	UltraFast (ResNet18) [16]	95.87%	Highest accuracy among those whose official code is available.
Curve-fitting	PolyLaneNet (b0) [19]	88.62%	Highest accuracy among those whose official code is available.
Anchor-based	LaneATT (ResNet34) [8]	95.63%	Highest accuracy among those whose official code is available.

We collect 20 free-flow[2] highway driving traces from the comma2k19 dataset [36]. For each driving trace, we consider two attack scenarios: attack to the left, and to the right. Thus, in total, we evaluate 40 different attack scenarios. For the lateral control, we use OpenPilot v0.7.0. For the longitudinal control, we used the velocity in the original trace. For the motion model, we use the parameters of Toyota RAV4 2017 (e.g., wheelbase), which is used to collect the traces of the comma2k19 dataset. We manually adjust the input image size and field-of-view to be similar to the TuSimple dataset. We use a 5.4 m x 36 m patch size, which is the same as the one used in the DRP attack [5]. The patch is placed at 7 m away from the vehicle at the first frame. When the patch covers lane lines, we draw lane lines on the patch to keep the original lane line information. When generating the attack, we use the first 20 frames (1 second). When evaluating the attack, we use all 50 frames (2.5 seconds), the average driver's reaction time. More details of each attack implementation and parameters are in our preprint paper [31].

[1]We obtained the pretrained models from:
LaneATT https://github.com/lucastabelini/LaneATT
SCNN https://github.com/harryhan618/SCNN_Pytorch
UltraFast https://github.com/cfzd/Ultra-Fast-Lane-Detection
PolyLaneNet https://github.com/lucastabelini/PolyLaneNet

[2]Vehicle has at least 5-9 seconds headway.

TABLE II: Attack success rate under the end-to-end benign and 3 different attack for targeted and untargeted goals. *Benign* is the benign failure rate defined in §IV-B. The **bold** and underlined letters mean the highest and lowest attack success rates, respectively.

		Targeted Goal			Untargeted Goal		
	Benign	WB DRP	BB DRP	BB Draw	WB DRP	BB DRP	BB Draw
LaneATT	20%	**78%**	**53%**	**90%**	**98%**	**88%**	**95%**
SCNN	**30%**	**78%**	43%	58%	**98%**	75%	70%
UltraFast	25%	75%	50%	83%	90%	48%	93%
PolyLaneNet	5%	48%	25%	30%	78%	43%	48%

B. Evaluation on End-to-End Driving Scenario

To evaluate the system-level effects in autonomous driving, we conduct an end-to-end evaluation with the methodology introduced in §III-C. Table II shows the results of the end-to-end evaluation. As shown, PolyLaneNet demonstrates the highest robustness as it has the lowest attack success rates in all attack scenarios. We can observe a typical trade-off of accuracy and robustness. As in Table I, PolyLaneNet is reported as a lesser performance model. However, in terms of the robustness, PolyLaneNet has the best robustness among 4 major lane detection models.

RQ1: Is black-box attack as effective as white-box attack?

Generally, the white-box attack has more attack capability as it can leverage more specifications of target models. However, recent studies report that black-box attacks can outperform white-box attacks [37] because gradient descent-based methods tend to suffer from local minima. In our evaluation, the same phenomenons are observed: the black-box drawing-lane-line attack outperforms to the white-box DRP attack on LaneATT and UltraFast. The black-box DRP attack has generally lower attack capability than the white-box DRP attack. We think that the stealthiness constraints of the DRP attack (e.g., gray-scale color and perturbable area ratio) could be too complex to be effectively optimized by the NES-based gradient estimation. Meanwhile, the black-box DRP attack has a high attack success rate (88% for the untargeted goal) against LaneATT. Our results demonstrate that the black-box attacks have close or even effectiveness as the white-box attacks.

RQ2: Which attack vector is more effective to attack?

The black-box drawing-lane-line attack has the highest attack effectiveness on LaneATT and UltraFast. For PolyLaneNet, the black-box drawing-lane-line attack is more effective than The black-box DRP attack, while the white-box DRP attack is the most effective. SCNN has less vulnerable to the black-box DRP attack and the black-box drawing-lane-line attack, as the black-box drawing-lane-line attack has a higher attack success rate on the targeted goal, but the black-box DRP attack has a higher attack success rate on the untargeted goal. In summary, each lane detection approach has different sensitivity to the drawing-lane-line attack vector. For LaneATT, it could be due to the structure of anchor proposals as discussed in §II-A. However, LaneATT is the only anchor-based method that the source code is available so far. Further research is required to confirm if the vulnerability to the drawing-lane-line attack is derived from a particular design of LaneATT or a fundamental problem of the anchor-based approach. For UltraFast, it shows different sensitivity to the drawing lane line attack compared to SCNN, even though both UltraFast and SCNN predicts the lane line for each pixel. Due to the divide-and-conquer strategy of UltraFast, it may rely too much on local features, i.e., SCNN may judge the lane lines based on more global features such as semantics on the road (e.g., lane lines should be roughly parallel with other lanes). Due to the ease of attack deployability, the vulnerability to the drawing-lane-line attack is severe. For autonomous driving, we should choose relatively robust models against naive attacks like drawing-lane-line attacks.

WB DRP	Lane ATT	SCNN	Ultra Fast	Poly Lane Net
LaneATT	98%	90%	93%	88%
SCNN	88%	98%	88%	78%
UltraFast	73%	78%	90%	48%
PolyLaneNet	95%	88%	85%	78%

BB DRP	Lane ATT	SCNN	Ultra Fast	Poly Lane Net
LaneATT	88%	73%	75%	38%
SCNN	95%	75%	78%	38%
UltraFast	93%	80%	48%	38%
PolyLaneNet	95%	75%	65%	43%

BB Draw	Lane ATT	SCNN	Ultra Fast	Poly Lane Net
LaneATT	95%	73%	90%	90%
SCNN	85%	70%	83%	70%
UltraFast	90%	83%	93%	63%
PolyLaneNet	58%	53%	48%	48%

Fig. 2: Transfer success rate of all pairs of models for the untargeted goal in the end-to-end scenarios. Each row indicates the source model that generates the attack and each column indicates the target model.

RQ3: Are attacks transferable to other models?

As shown in Fig. 2, the attack success rate is mostly less than the attack generated with the target model (diagonal cells). However, the transfer success rates still keep high attack success rates. Moreover, the attack generated with LaneATT has high transferability to PolyLaneNet in the drawing-lane-line attack: The transfer success rate is 90% attack success rate in the untargeted Goal. The results indicate that PolyLaneNet also has a vulnerability to the drawing-lane-line attack, but the robustness of PolyLaneNet makes it more difficult to generate attacks. Hence, the attacks to one lane detection model are likely to have high transferability to another model, and it is sometimes helpful to find the vulnerability of lane detection models which are robust to normal adversarial attacks.

V. Discussions

While the vulnerability of DNN models against adversarial attacks is widely reported, we may have optimistic expectations that it is almost impossible to exploit it in autonomous driving due to the low deployability mentioned in [5] and the lack of demo in reasonable settings. For example, the demo in [6] is in the intersection without lane lines, which are generally out of the operational domain of ALC. Thus, to our knowledge, I have not observed that any production autonomous driving systems have defense mechanisms against adversarial attacks. However, it does not mean that we can select a lane detection method just based on benign performance. Several lane detection approaches may have high sensitivity against naive attacks like drawing-lane-line attacks. Surprisingly, the production ALC systems we mention select the lane detection approach which shows higher resilience against the drawing-lane-line attack: Tesla Models S adopts the segmentation approach and OpenPilot [7] adopts the Curve-fitting approach. We think this is not coincident but due to the careful design choice of the company. In the near future, more and more automakers will install autonomous driving features in their products. We would like to facilitate more research to build robust lane detection methods so that as many automakers as possible are aware of the risks involved in algorithm selection.

Possible Defenses: So far, effective DNN model-level defenses against adversarial attacks are not reported. Accord-

ing to [5], none of the input transformation-based defenses can effectively defend against their attack without harming performance in normal scenarios. Another possible defense is cross-checking with other data sources. For example, Level-4 autonomous driving systems typically obtain driving lane information from HD maps. However, this method incurs large additional costs as it needs quite accurate localization and continuous maintenance of the HD map.

VI. Conclusion and Future Direction

In this work, we report the recent progress on conducting the first large-scale empirical study to evaluate the robustness of 4 major types of lane detection methods under state-of-the-art 3 physical-world adversarial attacks in autonomous driving scenarios. We find that each lane detection method has different security properties. Particularly, several models show high vulnerability to the drawing-lane-line attack. Thus, it is essential to be aware of the robustness to such naive attacks as Tesla and OpenPilot choose relatively robust methods against the drawing-lane-line attack. We hope that our research will help as many automakers as possible to recognize the risks in choosing lane detection algorithms. In future work, we plan to evaluate a more wide variety of lane detection models and adversarial attacks, especially effective black-box attacks. We also plan to explore more research questions such as dataset transferability. Although Comma2k19 and TuSimple datasets are similar driver's view images, there can be some domain shifts between them. The evaluation of the attack applicability and transferability on different datasets is also a considerable aspect of the robustness of lane detection models. Based on the insight of this study, we would like to work on the development of effective defense methods and robust model training that can improve the robustness of lane detection models in practical autonomous driving.

Acknowledgment

This research was supported in part by the NSF CNS-1850533, CNS-1932464, CNS-1929771, CNS-2145493, and USDOT UTC Grant 69A3552047138.

References

[1] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent Progress in Road and Lane Detection: A Survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.

[2] "TuSimple Lane Detection Challenge," https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection, 2017.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing Properties of Neural Networks," in *ICLR*, 2014.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, "Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack," *29th USENIX Security*, 2021.

[6] P. Jing, Q. Tang, Y. Du, L. Xue, X. Luo, T. Wang, S. Nie, and S. Wu, "Too Good to Be Safe: Tricking Lane Detection in Autonomous Driving with Crafted Perturbations," in *30th USENIX Security Symposium*, 2021.

[7] "OpenPilot," https://github.com/commaai/openpilot.

[8] L. Tabelini, R. Berriel, T. M. P. ao, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, "Keep Your Eyes on the Lane: Real-time Attention-guided Lane Detection," in *CVPR*, 2021.

[9] L. Liu, X. Chen, S. Zhu, and P. Tan, "CondLaneNet: A Top-To-Down Lane Detection Framework Based on Conditional Convolution," in *ICCV*, 2021.

[10] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as Deep: Spatial CNN for Traffic Scene Understanding," in *AAAI*, 2018.

[11] Y.-C. Hsu, Z. Xu, Z. Kira, and J. Huang, "Learning to Cluster for Proposal-Free Instance Segmentation," in *IJCNN*, 2018.

[12] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards End-to-End Lane Detection: An Instance Segmentation Approach," in *IEEE IV*, 2018.

[13] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai, "RESA: Recurrent Feature-Shift Aggregator for Lane Detection," 2020.

[14] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning Lightweight Lane Detection CNNs by Self Attention Distillation," in *CVPR*, 2019.

[15] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai, "RESA: Recurrent Feature-Shift Aggregator for Lane Detection," *AAAI*, 2021.

[16] Qin, Zequn and Wang, Huanyu and Li, Xi, "Ultra Fast Structure-Aware Deep Lane Detection," in *ECCV*, 2020.

[17] S. Yoo, H. S. Lee, H. Myeong, S. Yun, H. Park, J. Cho, and D. H. Kim, "End-to-End Lane Marker Detection via Row-Wise Classification," in *CVPR Workshop*, 2020.

[18] Y. Hou, Z. Ma, C. Liu, T.-W. Hui, and C. C. Loy, "Inter-Region Affinity Distillation for Road Marking Segmentation," in *CVPR*, 2020.

[19] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Polylanenet: Lane Estimation via Deep Polynomial Regression," in *ICPR*, 2021.

[20] J. Philion, "FastDraw: Addressing the Long Tail of Lane Detection by Adapting a Sequential Prediction Network," in *CVPR*, 2019.

[21] X. Li, J. Li, X. Hu, and J. Yang, "Line-CNN: End-to-End Traffic Line Detection with Line Proposal Unit," *IEEE T-ITS*, 2019.

[22] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, "Focus on Local: Detecting Lane Marker From Bottom Up via Key Point," in *CVPR*, 2021.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *NeurIPS*, 2015.

[24] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Examples in the Physical World," *arXiv preprint arXiv:1607.02533*, 2016.

[25] T. Brown, D. Mane, A. Roy, M. Abadi, and J. Gilmer, "Adversarial Patch," *arXiv preprint arXiv:1712.09665*, 2017.

[26] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, T. Kohno, and D. Song, "Physical Adversarial Examples for Object Detectors," in *WOOT*, 2018.

[27] B. Nassi, D. Nassi, R. Ben-Netanel, Y. Mirsky, O. Drokin, and Y. Elovici, "Phantom of the ADAS: Phantom Attacks on Driver-Assistance Systems." *IACR Cryptol*, 2020.

[28] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-Box Adversarial Attacks with Limited Queries and Information," in *ICML*, 2018.

[29] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," in *NeurIPS*, 2011.

[30] "Hyperopt," https://github.com/hyperopt/hyperopt.

[31] T. Sato and Q. A. Chen, "On Robustness of Lane Detection Models to Physical-World Adversarial Attacks in Autonomous Driving," *arXiv preprint arXiv:2107.02488*, 2021.

[32] R. Rajamani, *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.

[33] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.

[34] S. Tanaka, K. Yamada, T. Ito, and T. Ohkawa, "Vehicle Detection Based on Perspective Transformation Using Rear-View Camera," *International Journal of Vehicular Technology*, 2011.

[35] "Lane Keeping Assist System Using Model Predictive Control," https://www.mathworks.com/help/mpc/ug/lane-keeping-assist-system-using-model-predictive-control.html, 2020.

[36] H. Schafer, E. Santana, A. Haden, and R. Biasini, "A Commute in Data: The comma2k19 Dataset," *arXiv preprint arXiv:1812.05752*, 2018.

[37] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "Nattack: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks," in *ICML*, 2019.