Highway Hijackers: Evaluating Patch Attack Susceptibility in Autonomous Driving Lane Detection Systems

Romana Blazevic
romana.blazevic@tugraz.at
Graz University of Technology
Graz, Austria

Alexander Toch
alexander.toch@student.tugraz.at
Graz University of Technology
Graz, Austria

Fikret Basic
basic@tugraz.at
Graz University of Technology
Graz, Austria

Abstract

Recent advancements in autonomous driving technologies have rapidly increased the integration of deep neural networks (DNNs), particularly for lane detection. Despite their improved safety and efficiency, these systems remain highly vulnerable to attacks, such as patch attacks targeting pre-processed input data. This paper examines the susceptibility of lane detection systems using DNNs like ResNet-50, ERFNet, RESA, and SCNN within the MetaDrive simulation environment by evaluating the autonomous pipeline's robustness against patch attacks. For evaluation, we designed a comprehensive framework that encompasses image processing, lane inference, data post-processing, and steering control. Furthermore, we implemented two methods to disrupt lane detection: rendering patches in a 3D environment and injecting perturbations into the camera stream. To assess the impact of the attacks, we employed the End-To-End Lateral Deviation (E2E-LD) metric, which revealed significant impairments in the models' ability to detect the desired lane under the injected attacks. Our study highlights the critical need for enhanced security countermeasures and robust defenses in autonomous lane detection systems to mitigate the risks posed by patch attacks. The findings from this study serve as a contribution to the ongoing development of more robust technologies in autonomous vehicles, with our simulation models being applicable to real-world scenarios.

CCS Concepts

• **Security and privacy**; • **Computing methodologies** → *Simulation evaluation*; • **Hardware** → Safety critical systems;

Keywords

Autonomous driving, patch attacks, lane detection, deep neural network, security, RESA, SCNN, ERFNet, ResNet-50, E2E-LD, MetaDrive

1 Introduction

Autonomous vehicles have seen rapid advancements over the last decades, positioning themselves as a promising solution for revolutionizing transportation and enhancing safety [16]. They are designed to autonomously perform various operations typically requiring human intervention in a time-critical manner and can effectively respond to potential hazards during transportation [34]. These safety and time-critical operations are run as tasks of subsystems integrated with the vehicle, such as automatic brake system, lane keeping assistant, adaptive cruise control, etc. The accuracy of the execution of these autonomous subsystems is of utmost importance for the safety of the drivers and passengers. Among them, the lane detection assistant is crucial for maintaining the vehicle's desired trajectory. It relies on real-time data from cameras and sensors to accurately navigate the vehicle within lane boundaries [12].

Lane detection systems. Various systems are designed to execute the complex tasks of lane detection for answering real-time predictions taken from vehicle sensory inputs [2], with modern systems often relying on advanced machine learning models, such as neural networks. Deep neural networks (DNNs) [19] and convolutional neural networks (CNNs) have been widely employed for the complex task of lane detection due to their ability to process high-dimensional data effectively, starting in 2012 with AlexNet [17]. Subsequent architectures, such as VGG [1], have expanded the applications of CNNs to include semantic segmentation tasks. Notable approaches in semantic segmentation include Fully Connected Neural Networks (FCNNs) and encoder-decoder models [23]. Further advanced segmentation is achieved through architectures like ENet [29] and ERFNet [30]. Recurrent neural networks (RNNs), when combined with long short-term memory (LSTM) cells [39], have shown further improvement in lane detection accuracy. Dedicated models are also developed for more specialized tasks. Namely, spatio-temporal methods are applied to accurately depict lane markings, leveraging spatial and temporal cues [13]. Furthermore, functional link artificial neural networks (FLANNs) [8] are particularly effective for detecting curved lane lines, utilizing color channels.

Vulnerabilities of modern lane detection systems. Despite the capability of lane detection systems to process high-dimensional sensory inputs and make real-time predictions, they are still susceptible to various attacks that can compromise their performance. A compromised or hijacked lane detection system can lead to severe hazards. For instance, if the lane detection system is fooled with the road marking added by an adversary, the vehicle will be misled and even veered off the lane [15]. Furthermore, many attacks can manipulate training and test data to mislead algorithms and compromise their performance. For example, sensor attacks involve tampering with sensor data, while black-box and white-box attacks [35] exploit different levels of access to a model to generate perturbations that cause the model to malfunction. Recent attacks specifically targeting lane detection, such as DeepBillBoard [43] and Dirty Road Patch (DRP) [31], highlight the vulnerabilities in real-world environments, including systems like Tesla Autopilot.

Patch attacks. From the analyzed vulnerabilities, we observe that one of the critical protection assets is the input sensory data. Hence, it is necessary to investigate the susceptibility of lane detection systems to adversarial *patch attacks* [25] which involve inserting adversaries in different forms into input images to alter the algorithm's perception of lane markings and the surrounding environment, as seen in Figure 1. Defenses against these adversaries

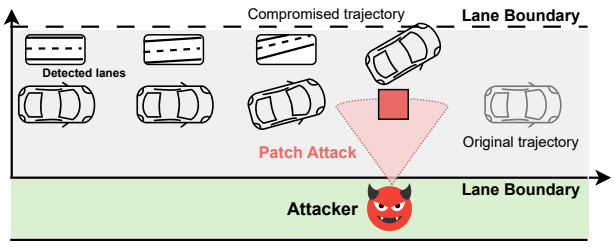

Figure 1: Patch attack on a lane detection system.

depend on a thorough understanding of lane detection threats, proposed adversaries and the development of accurate models that can predict diverse patch inputs, i.e., attack vectors, based on the observed vehicle environments and conditions. Currently, there are no definitive answers in this area, with no clear simulation models that accurately describe the real impact of such attacks [37].

We aim to fill this research gap and assess the effectiveness of patch attacks on open-sourced lane detection models such as ResNet-50 [11], ERFNet [30], RESA [42] and SCNN [28]. The study examines lane detection architectures and design specifics, focusing on how they interpret input data and accurately identify lane markings. To achieve this, the experimental analysis is conducted within the *MetaDrive simulator* [20], an advanced tool that allows the creation of diverse autonomous driving scenarios. MetaDrive can be configured with various road layouts and environmental conditions to challenge selected lane detection algorithms with different adversarial situations and assess their effectiveness.

Contributions. Addressing security challenges within lane detection algorithms is crucial for developing more resilient systems for autonomous driving applications. By investigating lane patch threats and their potential to manipulate lane detection systems, this study contributes to the enhancement of the safety and reliability of autonomous vehicles in the real world. Additionally, the results from our attack simulation models can be directly used as input training metrics in self-adapting systems, resulting in higher robustness for lane detection algorithms. To the best of our knowledge, we are the first to investigate and consider the use of such models to enhance the robustness of in-field lane detection systems.

2 Background and Related Work

In this section, we briefly describe lane detection systems that utilize DNNs as their decision mechanism, review the current state-of-the-art, and examine their vulnerabilities and potential attacks.

2.1 Deep neural networks for lane detection

DNNs have demonstrated exceptional capabilities in lane detection systems, effectively generalizing lane feature extraction and processing of complex data. They are prominent for the performance of advanced lane keeping and lane detection technologies such as Tesla Autopilot and OpenPilot. In this field, Huval et al. [14] were among the first to apply deep learning to lane detection, utilizing Convolutional Neural Networks (CNNs). Building on this foundational work, Neven et al. [26] approached the lane detection problem using instance segmentation. A more recent approach by Dong et al. [5] introduced Spatial CNNs (SCNN), which leverage a feedback mechanism to improve lane detection performance by addressing the absence of visual evidence for lane lines.

Zheng et al. [42] propose the Recurrent Feature-Shift Aggregator (RESA) to enhance the real-time performance of Spatial CNN (SCNN) by improving the model's ability to aggregate global features. Furthermore, Xu et al. [36] introduced the CurveLane Neural Architecture Search (CurveLane-NAS), providing a stable solution and a model for better detection of complex curved lanes.

The overall security of the listed systems heavily depends on the robustness of their underlying DNN models. Recent studies have shown that the models are susceptible to meticulously designed perturbations, posing significant risks [31]. Several specific lane detection models have been the focus of adversarial attacks. For example, Spatial CNNs (SCNNs) have been subjected to physical attacks, such as perturbing the road surface or placing stickers on the road to mislead the model [10]. LaneNet, which detects lane markings using a combination of deep learning and clustering, is also susceptible to digital attacks via adversarial perturbations [37]. A more prominent example is ENet, a real-time semantic segmentation model often used for lane detection tasks because of its speed and performance [29]. However, we believe ENet is left open to patch threats and can be attacked via small adversarial perturbations to input compromised images. In summary, we can determine that while the recent publications on lane detection systems are novel, they potentially suffer from security risks that can directly affect the entire vehicle. In this work, we expand on this dilemma and offer an analysis that shows the potential impact of the patch attacks against the robustness of the DNN models in use.

2.2 Targeted attacks, scenarios, and defenses

Under patch attacks, we consider different attack vectors depending on the observed scenario. In general, advanced driver-assistance systems (ADAS), such as lane detection, suffer from attacks that target autonomous driving. These include phantom attacks, dirty road patch attacks, and algorithm attacks, among others [4].

Phantom attacks involve creating depthless objects that deceive the autonomous vehicle into perceiving them as real physical obstacles or lane lines [24]. These perturbations can be projected onto the road using projectors or displayed on billboards.

Backdoor attacks manipulate a DNN during the training phase, causing it to make incorrect predictions when specific conditions are met [10]. This is achieved by poisoning a portion of the training data with backdoor triggers. In the context of autonomous vehicles, a lane detection system could be reverse-engineered to insert a backdoor trigger, leading to the vehicle steering off the lane.

Dirty Road Patch (DRP) attacks involve placing adversarial patches designed to fool the lane detection system without being placed on the actual lines, but rather in-between them [31]. Another way to fool the camera system is using the *blinding attacks*, which exploit the camera sensors' sensitivity to light and external conditions [38].

Attacks can also directly target the underlying algorithms used in data post-processing. For example, universal perturbations can be crafted using the Fast Gradient Sign Method (FGSM) [9], which is an algorithm used to generate adversarial examples by making small, intentional changes to input images. When these perturbations are

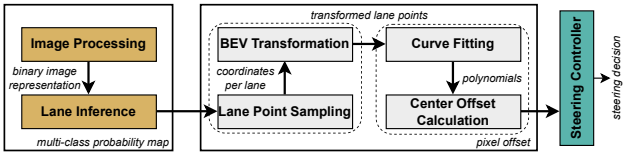

Figure 2: Autonomous driving pipeline. It consists of three main process tasks divided into seven sequential tasks.

added to the images, they potentially lead to misclassification by the model. Similarly, the Expectation Over Transformation (EOT) algorithm helps create adversarial patches that remain effective under transformations such as rotation and scaling [3].

In order to validate these attacks, researchers are using digital simulation environments and real-world testing. Simulators such as CARLA [6] and MetaDrive [20] provide environments for creating various scenarios in which a vehicle can be placed. Different kinds of attacks can then be integrated into each scenario to assess their impact on lane detection models. ART (Adversarial Robustness Toolbox) [27] is a comprehensive toolkit for evaluating the robustness of models against adversarial attacks, which supports various attacks, evaluation metrics, and defenses. Physical attacks involve printing perturbations, placing billboards, and modifying road signs, which are deployed in real-world scenarios to demonstrate the vulnerability of the algorithms to these attacks. These attacks reveal critical weaknesses in the systems, significantly impacting autonomous driving technologies. Defenses such as enhanced training, improved sensor fusion, and robust lane detection must be developed to protect from malicious attacks. Especially the effectiveness of adversarial training has to be evaluated, as to [40] the classical training with augmented data seems to be ineffective. Understanding and addressing the sensitivity of lane detection systems to adversarial attacks is essential for their safe integration into real-world autonomous vehicles, which we address in this work.

3 Methodology of Evaluating Patch Attacks Susceptibility for Lane Detection Systems

To evaluate the robustness of the DNN models used for lane detection across various scenarios, we selected MetaDrive [20] as the simulator. According to recent research by Li et al. [21], MetaDrive is the only actively maintained driving policy simulator that provides realistic sensor output and supports the dynamic generation of infinite driving scenarios. In this section, we will outline our proposed methodology for evaluating patch attacks in lane detection systems. Our solution is designed to be flexible and easily reusable across different evaluation models, accommodating the rapid advancements in autonomous vehicle research.

3.1 Autonomous driving pipeline

To create a realistic autonomous driving pipeline, our driving policy consists of seven sequential steps grouped into three main processes, shown in Figure 2 with corresponding numbered steps.

Image processing and lane inference (1-2). For the first step, the images are captured and processed internally (1), after which, the lane interference (2) is applied via Pytorch semantic segmentation models. MetaDrive provides an RGB Camera sensor, which is mounted on the vehicle center and provides road images for every simulation step. To avoid resizing and the resulting loss of image information, we always configure the simulator to match the input size of the used DNN model. Since MetaDrive outputs the image as a CuPy array, we can directly input the image to the Pytorch model on the GPU without copying it there. The model then outputs multiple probability maps depending on its configuration (i.e., RESA supports up to 4 lanes), which represent the probability of each road image pixel belonging to a lane.

Lane data post-processing (3-6). As a next step, we sample lane coordinates (3) for each lane class by thresholding the probability map values in regular intervals, similar to the implementation of Pan et al. [28]. In order to fit the lanes to 2nd-degree polynomials, the lane coordinates need to be transformed into a birds-eye-view (BEV) perspective (4), where parallel lanes appear as parallel lines. For this transformation, we now calculate the intrinsic matrix K:

$$K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

Where f_x and f_y are the focal lengths in horizontal and vertical direction, and x_0 and y_0 are the principal point offsets. Fortunately, we can retrieve all these values from the Panda3D engine MetaDrive is based on. Afterward, we calculate the camera projection matrix $P = K[R|t]$ (R is the camera rotation in the world, t the translation), which maps 3D points into their 2D equivalent, and transforms this matrix to match the birds-eye view using a shift matrix M. The inverse projection mapping is then calculated with:

$$IPM = (P \cdot M)^{-1} \tag{2}$$

Finally, we transform the lane coordinates (5) to BEV and calculate the polynomials by using a least squares fit. In order to infer a steering decision, we derive the center offset (6) by calculating the midpoint between the ego lanes at the image bottom, assuming that the desired midpoint is in the image center.

Steering control (7). For steering the vehicle (7), we are using a proportional–integral–derivative (PID) controller provided by MetaDrive, which gets updated by a fixed steering value on the left if the detected midpoint is right to the desired midpoint and to the right if it's left of it. If no lanes are detected, the vehicle will brake, causing it to stop if there are no qualitative inferences possible.

3.2 Attack design

Our attack model uses the target probability map illustrated in Figure 3 that results in a big offset to the left and, therefore, a steering decision to the right. The custom *Robust DPatch* pipeline aims to optimize the "patch" to match the target of the attack. This is made possible with cross-entropy as a loss function, using the loss gradients to update the patch image.

To evaluate the susceptibility of DNN lane detection models to adversarial examples, we implemented two approaches to inject the generated patches into our driving pipeline:

(1) Direct injection of the attack patch into the camera stream
(2) Rendering the patch into the 3D environment

(a) Annotated RGB image **(b) Probability map**

Figure 3: Attack target with patches. The vehicle moves in the simulation model (a) and uses the (b) probability map as input.

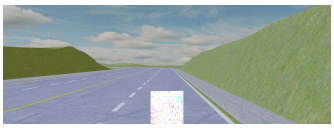

Figure 4: Example model input for attack approach 1.

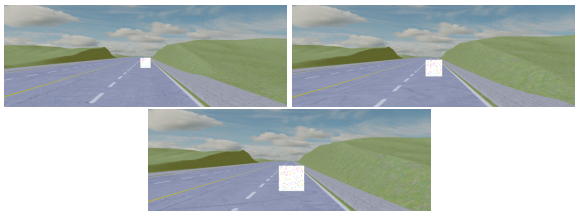

Figure 5: Example model inputs for attack approach 2.

The first approach hooks into the MetaDrive and generates the adversarial patch during the simulation at a given step or driving distance. After the attack is engaged, the patch is generated via our modified ART pipeline, and the resulting patch image is directly injected into the lane detection model. Figure 4 shows a "patched" model input with the adversarial patch at the image's bottom. This image is in a static position until the simulation finishes.

The second approach consists of two phases: First, the simulation is run without the attack. We generate an adversarial patch and store it in memory at a specified simulation step. After the simulation, we regenerate the terrain and place a white rectangle at the supposed location of the adversarial patch. Now, the simulation is started again, and the white rectangles in the RGB camera image are dynamically replaced with the previously generated patch image. This approach provides a direct comparison between the baseline simulation and the attack. Figure 5 illustrates how the generated adversarial patch is rendered into the simulation.

3.3 Attack evaluation

We use the implementation and CULane pre-trained weights of lane detection networks as provided in the PyTorchAutoDrive [7] framework, where we chose the following backbone networks and two specialized lane detection networks: ResNet-50 [11], ERFNet [30], RESA [42] and SCNN [28] (both with ResNet-50 backbone). The selection was limited to networks based on a CNN architecture. Despite SCNN, none of the networks was previously shown to be affected by adversarial examples in a lane detection context [32].

To distinguish between a successful and failed attack, we consider a scenario to be attacked successfully if one of the following events occurs:

(1) The vehicle steers out of the roadway
(2) The lane detection pipeline fails to detect less than 2 lanes for more than 50% of the last 20 simulation steps

We rely on the *End-To-End Lateral Deviation (E2E-LD)* metric to perform an end-to-end performance evaluation of the lane detection system after the attack has been initiated. Unlike the conventional metrics such as the F1-score, the E2E-LD metric proposed by Sato et al. [32] provides a more robust metric tailored to the specialized task of lane detection. The metric is defined as:

$$\max_{t \leq T_E}(|L_t - C_t|) \tag{3}$$

where L_t is the lateral vehicle coordinate provided by the MetaDrive simulator at the simulation step t. C_t is the lane width at step t. T_e is defined by the maximum step count in the simulator configuration. They also propose a Per-frame Simulated Lateral Deviation metric (PSLD), which is a computationally more lightweight surrogate metric that can calculate the center offset based on a single frame and can be applied during model training. The simulator that we use gives us direct access to the driving metrics with the E2E-LD.

4 Evaluation Results and Discussion

Evaluation environment. For our simulation setup, we utilized pre-trained models from the PytorchAutoDrive [7] framework. The attack model is based on the implementation of the *Robust DPatch* attack for object detection models from Xin et al. [22] and the work of Lee et al. [18] in the Adversarial Robustness Toolbox [27]. We adapted the *Robust DPatch* attack implementation for the semantic segmentation models provided by PytorchAutoDrive. In order to confirm if an attack was the reason for the vehicle going out of the road or not recognizing lanes, we first run the scenario without an attack and only consider the scenario for evaluation if it completes successfully. All evaluation tests were performed on a Nvidia GeForce RTX 3060 Ti GPU and an Intel Core i5-10400F CPU. We model two attack scenarios, "Attack approach 1" (At_1) performed using patch placements directly positioned in the camera stream, and "Attack approach 2" (At_2) with a dynamic patch placement in the 3D environment. Table 1 shows the configuration setup.

Our initial experiments showed that (At_1) is highly effective and results in a successful attack scenario directly after its enablement. Consequently, we employed a lower maximum simulation step count and earlier patch placement than in (At_2). Furthermore, with

Configuration	At_1	At_2
Number of distinct scenarios	100	100
Initial scenario generation seed	1000	1000
Simulation step duration	0.02s	0.02s
Maximum simulation steps	400	500
Patch generation iterations	250	500 (250 SCNN)
Patch size	1x1	1x1
Simulator resolution	800x200	800x200
Patch placement at longitude	30m	60m
Target speed	100 km/h	100 km/h

Table 1: Simulation configuration of the attack approaches.

	Model	Attack success	E2E-LD $[m]$	
			Benign	Attack
Variant 1	ResNet-50	94.74%	0.16	0.07†
	ERFNet	98.96%	0.07	0.26
	RESA	100.00%	0.10	0.40
	SCNN	100.00%	0.00	0.00†
Variant 2	ResNet-50	98.33%	0.20	0.00†
	ERFNet	20.69%	0.05	2.71
	RESA	66.66%	0.22	1.64
	SCNN	47.50%	0.01	0.06†

†All successful attacks resulted in a fully stopped vehicle

Table 2: Experiment results of run attack scenarios.

(At_2), we observed that the system-level effects of the attack could occur either before or after the patch placement. Therefore, we extended the simulation duration for the second approach.

Table 2 shows the simulation results of (At_1) & (At_2). The lower number of successful benign simulations with (At_2) compared to (At_1) lies in the higher maximum step count, resulting in a higher error rate for the CULane-trained models misclassifying lanes in the MetaDrive environment. The reason for the relatively low *E2E-LD* values in Table 2 is due to the patch attack significantly influencing the inference output, preventing any lane detection from the probability map and causing the vehicle to come to a complete stop. This occurs early when the vehicle is positioned centrally within the lane, resulting in minimal or no deviation from the lane center.

Attack susceptibility. Our attack evaluation shows that in scenarios with succeeded attacks, ERFNet and RESA are susceptible to the used target probability map, causing the vehicle to steer out of the road, whereby ResNet-50 and SCNN completely fail to infer enough correct lane pixels to fit polynomials after enabling the attack. This is especially evident with the (At_1), where the attack's success is almost a certainty. We assume that the slightly higher attack success rate for ResNet-50 in (At_2) is caused by the different initial scenario generation seed for both approaches, resulting in different simulation terrains. Surprisingly, ERFNet, which - to our knowledge - has not been evaluated so far regarding its robustness, has the lowest attack success rate in (At_2). Our assumption is that ERFNet has a higher dependency on the patch location in the image than the other models, mitigating the attack in the more realistic scenarios of (At_2), where the patch has no static position in the camera input stream. (At_1) shows that DNN models trained on open-source datasets show almost no robustness against adversarial examples in an attack setting with direct access to the camera stream. However, in the more realistic scenarios of (At_2), the attack success rate is significantly lower. As the patch image is only optimized for one single camera frame from the benign simulation, its transferability to a dynamic multi-frame simulation is limited. Still, all evaluated models showed attack success rates that we consider not suitable for real-world driving scenarios, emphasizing the need for standardized evaluation metrics for lane detection tasks.

Evaluation metrics. For models susceptible to the target probability map, the *E2E-LD* metric proved to be a reliable indicator of attack success, as it correlated with lane deviation during benign simulations. However, if an attack causes a total "blackout" of the lane inference pipeline — meaning no information is received from step (2) described in Section 3.1 — the *E2E-LD* metric becomes less impactful in evaluating the robustness. This is because it does not account for situations where no lanes are detected at all. Therefore, additional factors and metrics should be considered to validate the *E2E-LD* metric in such evaluation contexts, such as a clear definition of a successful attack in relation to the simulation configuration (i.e., simulations steps or vehicle speed). The vehicle's behavior, particularly when no lanes are detected, is a critical factor in interpreting metrics like *E2E-LD*. E.g., the deactivation or intermittent failure of the safety-critical automated lane change (ALC) system must be considered when designing these systems for real vehicles.

5 Application for Autonomous Vehicles

The simulation methodology we presented in Section 3, and subsequent attack simulation results from Section 4, can be applied to enhance the real-world development and deployment of autonomous vehicles. By replicating potential adversarial scenarios in a controlled environment, we can identify and address vulnerabilities in lane detection algorithms before they manifest on actual roads. The results of our simulation models can be used as input for improving the general robustness of lane detection systems and, therefore, the protection of the same.

Impact on real-world driving scenarios. From the observed results, it is evident that (At_1) should be considered only in ideal scenarios where the attacker, through an "oracle", possesses all the resources required to successfully compromise the camera image stream. Conversely, (At_2) more accurately represents real-world scenarios and can be adjusted to align with successful physical-world attacks, such as those demonstrated also in real-world scenarios by Sato et al. [31]. The *Robust DPatch* used in this work has been shown to be adversarial under physical lighting conditions and placements by Lee et al. [18]. **Improving the robustness of lane detection systems**. The simulation models developed for evaluating patch attacks in this study can be easily expanded and integrated with other lane detection models to conduct large-scale, end-to-end evaluations, assessing system robustness using reliable task-specific metrics. Furthermore, the simulation pipeline can be utilized to automatically generate new training data, enabling the application of the *adversarial training* technique to enhance model robustness [33]. However, as the positive effect of adversarial training is questionable, future research may evaluate the effects of Gaussian Data

Augmentation (GDA) [40], or Out of Distribution (OOD) Detection in the context of lane detection models [41].

6 Conclusion

In this paper, we explored the vulnerability of lane detection models that rely on DNNs such as ResNet-50, ERFNet, RESA, and SCNN to patch attacks using MetaDrive. To achieve this, we constructed a novel simulation model and derived two attack scenarios: one based on an idealistic approach and one that references real-world conditions. Our experiments demonstrated that patches can significantly impact the robustness of lane detection systems. In the idealistic attack scenario, rendering in the 3D environment and injecting the attack into the camera stream showed that minor alterations to input images can lead to model malfunctions, potentially causing the vehicle to veer off the lane. Similarly, the realistic models displayed modest to high susceptibility to patch attacks, with ResNet-50 being particularly affected. Furthermore, utilizing the E2E-LD metric, we assessed the severity of compromising patch attacks, revealing several instances where the vehicle's lane detection capability was critically impaired. To enhance the robustness of lane detection systems against patch attacks, future work could potentially focus on integrating event cameras for high temporal resolution data to improve performance in dynamic environments, while also experimenting with recent advancements in lane detection systems such as BEV-LaneDet and LaneCPP to mitigate attacks for 3D-based lane detection systems. Our findings highlight the critical need to enhance the security and robustness of autonomous driving systems against malicious threats. By identifying and examining weaknesses of lane detection algorithms, this research contributes to the development of more robust and resilient autonomous vehicle technologies. The insights of our study navigate future improvements in defense strategies for the design and development of safer autonomous vehicles in real-world driving environments.

References

[1] Mahmoud Abouelyazid. 2022. Comparative Evaluation of VGG-16 and U-Net Architectures for Road Segmentation. *Eigenpub Review of Science and Technology* 6, 1 (2022), 75–91.

[2] Romana Blazevic, Fynn Luca Maaß, Omar Veledar, and Georg Macher. 2024. Intelligent Decision-Making in Lane Detection Systems Featuring Dynamic Framework for Autonomous Vehicles. In *International Conference on Computer Safety, Reliability, and Security*. Springer, 21–33.

[3] Tom B Brown et al. 2017. Adversarial Patch. *ArXiv* abs/1712.09665 (2017).

[4] Yao Deng et al. 2021. Deep Learning-based Autonomous Driving Systems: A Survey of Attacks and Defenses. *IEEE Transactions on Industrial Informatics* 17, 12 (2021), 7897–7912.

[5] Yongqi Dong et al. 2023. A Hybrid Spatial-temporal Deep Learning Architecture for Lane Detection. *Computer-Aided Civil and Infrastructure Engineering* (2023).

[6] Alexey Dosovitskiy et al. 2017. CARLA: An Open Urban Driving Simulator. In *Conference on Robot Learning*. PMLR, 1–16.

[7] Zhengyang Feng et al. 2022. Rethinking Efficient Lane Detection via Curve Modeling. In *IEEE/CVF Conference on CVPR*.

[8] Safwan Ghanem et al. 2023. An improved and low-complexity neural network model for curved lane detection of autonomous driving system. *Soft Computing* 27, 1 (2023), 493–504.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. *ArXiV* abs/1412.6572 (2014).

[10] Xingshuo Han et al. 2022. Physical Backdoor Attacks to Lane Detection Systems in Autonomous Driving. In *Proceedings of the 30th ACM IMC*. 2957–2968.

[11] Kaiming He et al. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[12] Bar Hillel et al. 2014. Recent progress in road and lane detection: a survey. *Machine vision and applications* 25, 3 (2014), 727–745.

[13] Yuhao Huang et al. 2018. Spatial-Temproal Based Lane Detection Using Deep Learning. In *Artificial Intelligence Applications and Innovations*. Springer.

[14] Brody Huval et al. 2015. An Empirical Evaluation of Deep Learning on Highway Driving. *ArXiv* abs/1504.01716 (2015).

[15] Pengfei Jing et al. 2021. Too Good to Be Safe: Tricking Lane Detection in Autonomous Driving with Crafted Perturbations. In *30th USENIX Symposium*.

[16] Philip Koopman and Michael Wagner. 2017. Autonomous Vehicle Safety: An Interdisciplinary Challenge. *IEEE Intelligent Transportation Systems* 9 (2017).

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60 (2012).

[18] Mark Lee and Zico Kolter. 2019. On Physical Adversarial Patches for Object Detection. *ArXiv* abs/1906.11897 (2019).

[19] Jun Li, Xue Mei, Danil Prokhorov, and Dacheng Tao. 2017. Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene. *IEEE Transactions on Neural Networks and Learning Systems* 28, 3 (2017), 690–703.

[20] Quanyi Li et al. 2023. MetaDrive: Composing Diverse Driving Scenarios for Generalizable Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2023), 3461–3475.

[21] Yueyuan Li et al. 2024. Choose Your Simulator Wisely: A Review on Open-source Simulators for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles* (2024), 1–19. https://doi.org/10.1109/TIV.2024.3374044

[22] Xin Liu et al. 2018. DPATCH: An Adversarial Patch Attack on Object Detectors. *ArXiv: Computer Vision and Pattern Recognition* abs/1806.02299 (2018).

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on CVPR*.

[24] Ben Nassi et al. 2020. Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks. In *Proceedings of the 2020 ACM SIGSAC (CCS '20)*. Association for Computing Machinery, 293–308.

[25] Federico Nesti et al. 2022. Evaluating the Robustness of Semantic Segmentation for Autonomous Driving against Real-World Adversarial Patch Attacks. In *Proceedings of the 2022 IEEE/CVF WACV*.

[26] Davy Neven et al. 2018. Towards End-to-End Lane Detection: an Instance Segmentation Approach. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. 286–291.

[27] Maria-Irina Nicolae et al. 2018. Adversarial Robustness Toolbox v1. 0.0. *ArXiv* abs/1807.01069 (2018).

[28] Xingang Pan et al. 2018. Spatial As Deep: Spatial CNN for Traffic Scene Understanding. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*.

[29] Adam Paszke et al. 2016. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *ArXiv* abs/1606.02147 (2016).

[30] Eduardo Romera et al. 2018. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE T-ITS* 19, 1 (2018), 263–272.

[31] Takami Sato et al. 2021. Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack. In *30th USENIX*.

[32] Takami Sato and Qi Alfred Chen. 2022. Towards Driving-Oriented Metric for Lane Detection Models. arXiv:2203.16851 [cs.CV] https://arxiv.org/abs/2203.16851

[33] Christian Szegedy et al. 2013. Intriguing properties of neural networks. *ArXiv* abs/1312.6199 (2013).

[34] Jun Wang, Li Zhang, Yanjun Huang, and Jian Zhao. 2020. Safety of Autonomous Vehicles. *Journal of Advanced Transportation* (2020).

[35] Qixue Xiao et al. 2019. Seeing is not believing: Camouflage attacks on image scaling algorithms. In *28th USENIX Security Symposium*. 443–460.

[36] Hang Xu et al. 2020. CurveLane-NAS: Unifying Lane-Sensitive Architecture Search and Adaptive Point Blending. In *Proceedings of the 16th ECCV*. Springer.

[37] Henry Xu, An Ju, and David A. Wagner. 2021. Model-Agnostic Defense for Lane Detection against Adversarial Attack. *ArXiv* abs/2103.00663 (2021).

[38] Chen Yan et al. 2016. Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle. *Def Con* 24, 8 (2016).

[39] Wei Yang et al. 2020. Lane position detection based on long short-term memory (LSTM). *Sensors* 20, 11 (2020).

[40] Valentina Zantedeschi et al. 2017. Efficient Defenses Against Adversarial Attacks. arXiv:1707.06728 [cs.LG] https://arxiv.org/abs/1707.06728

[41] Wenjie Zhao et al. 2024. Segment Every Out-of-Distribution Object. arXiv:2311.16516 [cs.CV] https://arxiv.org/abs/2311.16516

[42] Tu Zheng et al. 2021. RESA: Recurrent Feature-Shift Aggregator for Lane Detection. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

[43] Husheng Zhou et al. 2020. DeepBillboard: systematic physical-world testing of autonomous driving systems. In *Proceedings of the 42nd ACM/IEEE ICSE*.