Model-Agnostic Defense for Lane Detection Against Adversarial Attack

Henry Xu
An Ju
David A. Wagner

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-105
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-105.html

May 14, 2021

Acknowledgement

I would like to thank my advisor, Professor David Wagner, for his mentorship and guidance; this report was made possible by your support.

I would also like to thank An Ju for his invaluable advice and insight and Professor Dawn Song for her help as a reader. I am grateful to Mark Wu, Quentin Delepine, Zachary Golan-Strieb, and Norman Mu for their feedback along the way.

Finally, I would like to thank my family for their unconditional love and support throughout my education.

Model-Agnostic Defense for Lane Detection Against Adversarial Attack

by Henry Xu

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor David Wagner
Research Advisor

5/13/2021

(Date)

* * * * * * *

Professor Dawn Song
Second Reader

5/13/2021

(Date)

Model-Agnostic Defense for Lane Detection Against Adversarial Attack

Henry Xu
henryxu@berkeley.edu
UC Berkeley

An Ju
an_ju@berkeley.edu
UC Berkeley

David Wagner
daw@cs.berkeley.edu
UC Berkeley

Abstract

Susceptibility of neural networks to adversarial attack prompts serious safety concerns for lane detection efforts, a domain where such models have been widely applied. Recent work on adversarial road patches have successfully induced perception of lane lines with arbitrary form, presenting an avenue for rogue control of vehicle behavior. In this paper, we propose a modular lane verification system that can catch such threats before the autonomous driving system is misled while remaining agnostic to the particular lane detection model. Our experiments show that implementing the system with simple convolutional neural networks (CNN) can defend against a wide gamut of attacks on lane detection models. We can detect 96% of nonadaptive bounded attacks, 90% of adaptive bounded attacks, and 90% of adaptive patch attacks while preserving accurate identification at least 95% of true lanes using a 3-layer architecture imposing at most a 10% impact to inference time. Using ResNet-18 as a backbone, we can detect 99% of bounded non-adaptive attacks and 98% of bounded adaptive attacks, indicating that our proposed verification system is effective at mitigating lane detection security risks.

1. Introduction

End-to-end lane detection methods have shown great promise; however, their shared foundation with deep neural networks imply a shared weakness to adversarial examples [17]. Given the importance of accurate lane detection in downstream control decisions for autonomous vehicles, a successful attack on lane perception could result in undesirable or outright dangerous vehicle behavior. In particular, we are interested in attacks that could interfere with vehicle guidance through the generation of malicious lane lines, where attack success is marked not by alarm, as is the case when lane lines cannot be found, but by a false sense of normalcy. With no defense, as is the case with current state-of-the-art efforts, a lane detection pipeline is unable to make any judgement of lane validity, and thus the perceived adversarial lanes are indistinguishable from real lanes. To defend against such attacks, we propose a system for lane verification as illustrated in Figure 1, with the goal not to recover the original lanes, but to minimize instances of lane detection model false confidence.

Our lane verification model is fast, lightweight, and applicable to any existing lane detection effort. The simplicity of our verification model imparts very little inference overhead, and its modular nature allows for independent training that avoids the costs associated with redesigning and retraining the large and complex neural networks commonly seen in industrial lane detection systems.

The modularity carries the additional benefit of being lane detection model-agnostic, paving a path for integration into any lane detection pipeline. Given the constant improvement and refinement of lane detection techniques, detaching our defense from a particular architecture allows it to remain viable as the underlying methods become more sophisticated.

Our system is motivated by the framing of secure lane detection as two complementary tasks: lane proposal and lane verification. The former requires discerning the locations of a variable number of lanes in a constantly changing environment; the latter boils down to binary classification: given a set of lane coordinates, determine if they correspond to a lane that is either real or adversarial. Instead of further complicating the optimization problem faced by existing lane proposal models by piling on a secondary goal of security on top of their initial purpose, we propose moving the task of verification into a separate bespoke model, allowing each part of the pipeline to focus on maximizing individual performance without compromise. Since the task of lane verification can take lane coordinates as

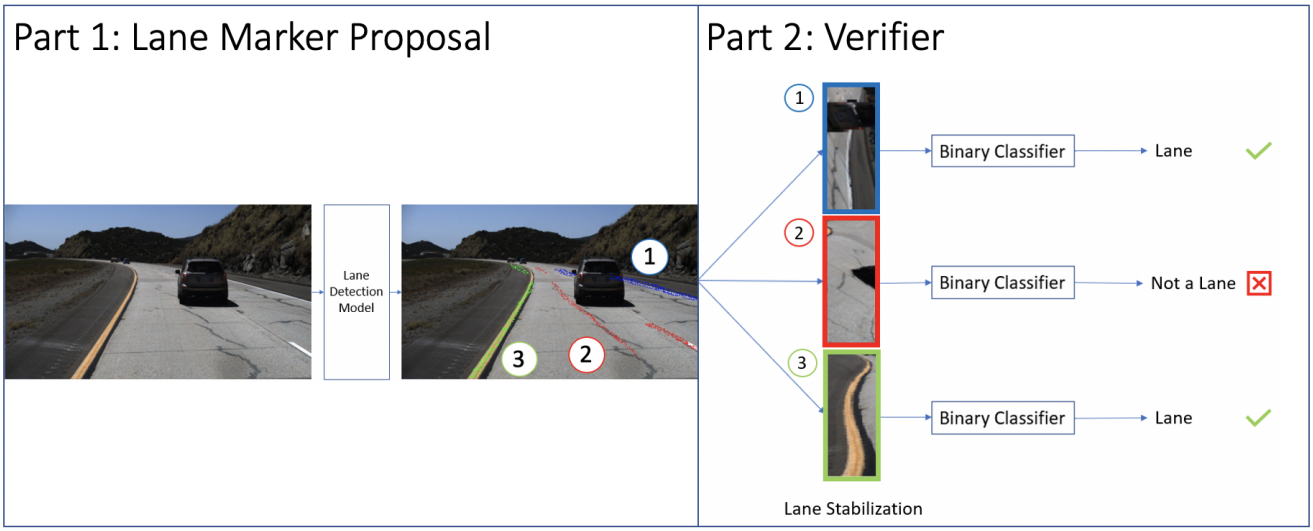

Figure 1. Our proposed defense augmentation to a general lane detection model.

given and only needs to return a binary result, it can be accomplished by models much simpler and faster than those required for lane proposal.

Our experiments show that simple convolutional models are sufficient to significantly improve lane detection pipeline robustness to both digital and physical attack as pictured in Figure 2. When evaluated against a L_p bounded attack and two patch-based attacks, variants of our model can detect over 95% of attacks with minimal impact to model accuracy and inference time. These results suggest that our system is capable of defending against a variety of attack types, including unknown threats.

In summary, strong performance of our defense against both nonadaptive and adaptive versions of such threats indicates that such a system could offer security to lane detection models at very little expense.

Our primary contributions are as follows:

- We propose a simple lane verification defense that can be integrated into the pipeline of any lane detection effort with no retraining of the underlying model required. Its independent and lightweight nature provides marginal inference overhead and allows for quick security updates when new attacks arise.

- We show empirically that verification provides significant lane detection security with minimal cost.

2. Related Work

While work on lane detection model defenses is sparse, there is extensive related work on end-to-end detection models and some work on lane detection attacks which has been summarized below.

2.1. End-to-End Lane Detection Models

Convolutional neural network-based lane detection models typically frame their core task of lane proposal as one of image segmentation, with the goal to label each pixel as one of N classes, each class corresponding to a distinct lane. In the end-to-end formulation, the segmentation is accompanied by a parallel binary labelling of lane existence, allowing the model to narrow down where exactly the lane is within pixels of the same segmentation class. By doing so, an end-to-end lane detection model is able to take a scene and return predictions of lane line locations. Proposed models largely differ on neural network architecture choices and postprocessing cleanup procedures, the details of which our defense treats as a black box.

Given the methodological similarities between the top lane detection models on the TuSimple dataset, we chose to test our proposed defense with LaneNet as proposed by Neven et al. [11] due to its near state-of-the-art performance at time of writing and result reproduction accessibility. We achieve accuracy within 2% of Neven et al.'s results before adding our defense. Note that due to the lane detection model-agnostic nature of our defense, the results from our experiments should be applicable to any other model we could have chosen, such as [6], [7], and [8].

2.2. Image Comprehension Model Attacks and Defenses

The framing of lane detection as a task of image segmentation suggests a sharing of similar security weaknesses, and recent work has shown image comprehension models to be very susceptible to adversarial attack. Adversarial examples, shown to be incredibly effective for image classification, have been shown to be extendable to image segmentation, with [3] specifying a framework generalizing their generation across a variety of tasks, including segmentation, and [17] finding attack success across a variety of segmentation networks. Successful attack need not change every pixel as discovered in [2], and classification can easily be corrupted with a patch a fraction of the total size of the image. Defenses, such as adversarial training, as suggested and explored by [4], [10], [12], and [16] against adversarial attack, often involve retraining the entire model which is costly given the ever increasing complexity of state-of-the-art techniques.

2.3. Lane Detection Model Attacks

Regarding adversarial attacks on lane detection models in particular, recent work has used image segmentation attack methods to great effect. [13] details how a bounded patch, disguisable in practice as road dirt, could be used to fool lane detection models before the passenger catches on. The adversarial patches we test our defense on differ in their much smaller size and unboundedness compared to the full lane covering required by [13]. Although our metric of attack success show the patches are unable to achieve our goal of reshaping all the lanes in the scene, our results do reaffirm that patch attack-based lane deviation is a valid threat necessitating defense.

3. Method

3.1. Defense

Our proposed defense takes place at the end of an existing lane detection pipeline, at which point a set of candidate lanes have been proposed by the lane detection model. Upon attack success, these candidate lanes are corrupted and may include a mix of real and adversarial lanes. The goal of our defense is to verify the real lanes and filter out suspicious lanes before further autonomous vehicle systems make potentially hazardous decisions based on faulty information.

3.1.1 Stabilizing Lanes

The verifier model takes detected lanes as input; however, due to the nature of perspective and exacerbated

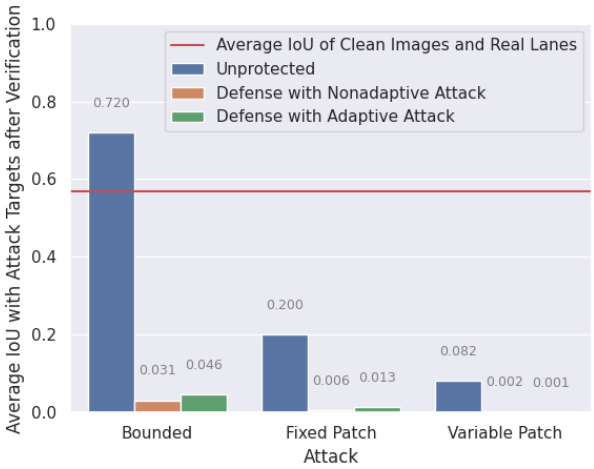

Figure 2. Defense Performance. Our defense is able to filter out all but a handful of adversarial attacks that would otherwise fool unprotected lane detection models. Intersection over Union (IoU) between detected lanes and attack targets is our chosen attack metric due to its measurement of both how well the induced adversarial lane matches the target adversarial lane and how much of the original scene was preserved, providing a sense of amount of control an attacker has over the scene. The bounded attack before verification surpasses the IoU achieved by the lane detection model on real lanes, suggesting the adversarial lanes are indistinguishable from real ones to the model. While patch attack IoU is relatively muted compared to that of bounded attack due to its largely local impact, it can still cause significant lane deviation as shown in Figure 6.

by the fact that lane lines can curve either left or right, extracting lanes using masks formed from the pixel-level segmentation as provided by the lane detection model results in lane lines that can take an arbitrary number of forms, lending itself to a classification problem with an unbounded domain. To address this issue, we construct a stabilized image of each lane as follows:

1. Given a set of points corresponding to a lane, we first perform a least squares polynomial regression to get its underlying shape.

2. For each pixel that lies on the curve, we compute the derivative of the polynomial at that point, and extract the pixels corresponding to the line centered around the point on the curve and rotated by the angle formed by the derivative and the horizontal axis.

3. Rotating each extracted line such that the pixels in each line are lined up horizontally, we can then stack the horizontal lines vertically to obtain a stabilized image of the lane that is not influenced by

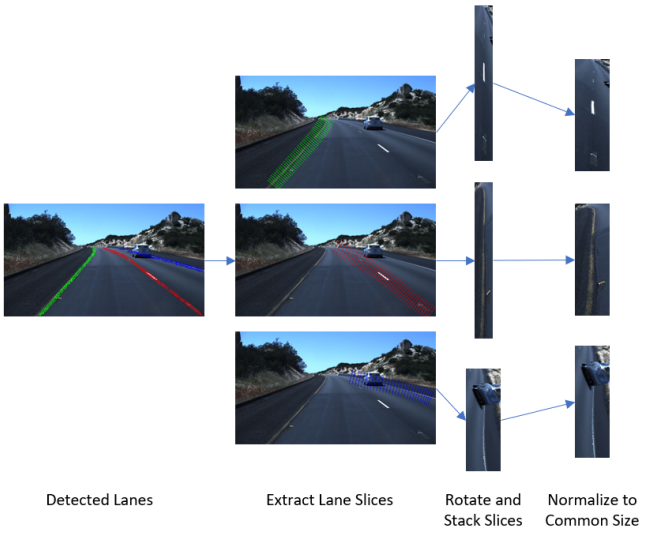

Figure 3. Lane Stabilization. Stabilized lanes are classified as either real or suspicious by the verification system.

perspective or lane curvature.

While inverse perspective transforms [1] can mitigate perspective distortions as well, we found little benefit from applying a fixed homography before extracting the stabilized lanes. Figure 4 shows examples of positive and negative samples generated by our stabilization process.

3.1.2 Training

Given the absence of applicable existing datasets, to generate a training dataset for the verifier model, we extract stabilized real and suspicious lanes from a dataset labelled for lane detection, using the ground truth labels as the basis for real lanes and generating curves that start from real lanes but deviate at arbitrary locations as the basis for suspicious lanes. While each scene has a fixed number of real lanes, an arbitrary number of suspicious lanes can be generated from them, creating a class imbalance. A potential enhancement of the defense could involve tailoring suspicious lane design in anticipation of specific attacks.

To address the aforementioned class imbalance, the verifier model is trained using focal loss as described in [9] with hyperparameters determined experimentally. To improve robustness, we employ adversarial training [10] with an emphasis on missed adversarial lane detection via asymmetric weighting of the loss function, prioritizing the class of the suspicious lanes.

Figure 4. Examples of positive and negative examples generated by the lane stabilization process for the defense training dataset.

3.2. Threat Model

We propose two different threat models dependent on attacker access to lane detection equipment. The threats share the goal of perturbing the original scene such that the binary segmentation component of an end-to-end lane detection model is disrupted, allowing the attacker to inducing an arbitrary lane existence pattern of their choosing. Due to the downstream dependence of lane detection models on lane existence, the attack can fatally disrupt lane detection model function. All attacks are carried out using Projected Gradient Descent [10] until convergence.

L-Infinity Threat Model In our first threat model, the attacker is able to manipulate any pixel in the input image, but is constrained by a bound on the size of perturbation for each pixel, akin to digital corruption of the input or a lens filter over the camera.

Patch Threat Model In our second threat model, the attacker is able to manipulate a subset of pixels in the input image, but is not constrained by a bound on the size of perturbation for each pixel. We can further break the subset of pixels into two sizes: fixed and variable. In a fixed size patch attack, there is a fixed number of pixels the attacker can modify, akin to having a patch on the lens of the camera. In a variable size patch attack, the number of pixels able to be modified is a function of distance away from the camera, akin to being able to lay a physical patch on the road in the scene. We can simulate such an threat model by scaling patch size based on pixel height, using the lane width and lane marker size at a given pixel height as reference.

3.3. Attacks

3.3.1 Nonadaptive Attacks

Nonadaptive attacks under each threat model aim to convince the proposal model of the existence of an ad-

versarial lane, but do not explicitly make an attempt to also convince our verification model that the lane is real.

3.3.2 Adaptive Attacks

Adaptive attacks [14] seek to fool the proposal model as in the nonadaptive case, but also take bypassing our verification model into account.

The lane stabilization procedure is non-differentiable, which poses an issue when computing gradients for an end-to-end attack. We instead propose an adaptive attack that takes place in two stages. We discuss the attack in the context of an L-infinity threat model, but the method can easily be extended to the patch scenario.

1. The first stage is an L-infinity attack on the lane detection model, with the goal to induce an arbitrary binary segmentation map of our choosing. The output is a perturbed scene in which the lane proposal model identifies the arbitrary lane we choose.

2. Once the first stage has converged, the pixels corresponding to the arbitrary lane are extracted from the perturbed scene. The lane is then stabilized as described in the process above, and subject to the second stage of the attack. The goal of the second stage is to find a perturbation to the stabilized lane such that the defense is fooled into thinking the arbitrary lane is real. Upon convergence of the L-infinity attack on the defense, the resulting perturbation on the stabilized lane is mapped back to the original location of the pixels that form the stabilized lane in the scene perturbed in the first stage.

The final result is a perturbed scene designed to both convince the lane detection model of the existence of an adversarial lane and the defense that the perceived lane is real.

4. Experiments

4.1. Datasets

The lane detection model, LaneNet, is trained on the TuSimple dataset [15], which consists of 3,626 training images and 2,782 testing images taken from a camera mounted on the front of a vehicle. The images are scaled to be of size 512x288.

The defense is trained on stabilized lanes extracted from the TuSimple dataset. Each stabilized lane is fit with a polynomial of degree 3 and resized to be of size 128x40.

4.2. Implementation Details

We test two different architectures for the verifier model. In the first architecture, the verifier is comprised of two convolutional layers and one linear layer. Both convolutional layers use a 3x3 filter size with stride 3 and no padding, with BatchNorm and ReLU applied after each. Inputs to the this architecture take the form of stabilized lanes of size 128x40 extracted from an original image size of 256x512. The second is a variant of ResNet-18 [5] modified to accept stabilized lanes of size 256x100 extracted from an original image size of 720x1280 and to perform binary classification.

Both LaneNet and the defense are trained using one GPU (GTX 1080). LaneNet is implemented as specified in [11], with no modifications for robustness, keeping in line with the modular nature of our defense. Adversarial training for the defense is tuned for bounded threat models and evaluated both bounded and patch threat models.

All attacks are performed until convergence is achieved. Some parameter details are as follows:

1. The L-infinity attack is bounded by a per-pixel perturbation of at most $8/255$, where the input image's pixels have a range of $[0, 1]$.

2. The fixed size patch attack is a 100x100 square with no bound on pixel deviations inside the square. The square is centered around a point on the targeted arbitrary lane.

3. The variable size patch attack is specified by a 100x100 square at the foot of the camera, corresponding to roughly a 3-foot by 3-foot physical patch. For each scene, an arbitrary distance from the camera is selected, and the square is scaled down accordingly.

Additional information on training details and hyperparameters can be found in the appendix.

4.3. Evaluation

We evaluate performance of the defense on the test set of TuSimple. For each scene in the original dataset, LaneNet is used to identify lanes in clean and attacked variants. The identified lanes are stabilized and supplied to the defense for classification. We report adversarial lane missed detection rates (false negative) and real lane misclassification rates (false positive) at a classification threshold such that we see a 5% real lane misclassification rate in a validation set. Additionally, we report the average Intersection over Union (IoU) values between all lanes in attacked scenes and attack targets once flagged lanes have been removed.

Figure 5. Example of L-infinity bounded attack. From left to right, the vertical pairs of images represent a clean image, a nonadaptive attack, and an adaptive attack against a 3-layer verification model. The second row shows the proposal output by the lane detection model superimposed on the scene, and IoU values below. IoU values are with respect to the target lane binary segmentation map, shown in the upper right, and the resulting stabilized lane is shown below. Under bounded attack, the attacker is able to assert full control over the scene and induce arbitrary lane configurations.

Table 1. Effectiveness of our defense, with a three-layer verifier. False positive rate (FPR) refers to real lanes the defense mistakenly flagged as adversarial, whereas false negative rate (FNR) refers to adversarial lanes the defense believed to be real. As a measure of post-defense attack success, the false negative (FN) average IoU is the average IoU between all attacked scenes and targets once flagged lanes have removed.

	Defense Metrics		*Attack Metric*
Bounded Attack	**FPR**	**FNR**	**FN IoU**
Unprotected	0	1	0.720
Nonadaptive	0.040	0.039	0.031
Adaptive	0.043	0.098	0.046

	Defense Metrics		*Attack Metric*
Patch Attack	**FPR**	**FNR**	**FN IoU**
Fixed Unprotected	0	1	0.200
Fixed Nonadaptive	0.050	0.025	0.006
Fixed Adaptive	0.049	0.100	0.013
Variable Unprotected	0	1	0.082
Variable Nonadaptive	0.044	0.034	0.002
Variable Adaptive	0.050	0.072	0.001

IoU values were of interest due to their dual purpose of measuring how much of the target was achieved and how much of the original scene was preserved, providing a sense of attacker control over the scene.

5. Results

5.1. L-Infinity Threat Model

5.1.1 3-Layer Verifier

Attacking the lane proposal model nearly always succeds under bounded attack, with all traces of the real lanes wiped out and the induced adversarial lanes matching target adversarial lanes with an average IOU of 0.720. Note that due to the polynomial fitting during lane stabilization, the induced adversarial lane and the target adversarial lane have almost identical stabilized forms. Examples of scenes and their corresponding binary segmentation maps before and after targeted attack are in Figure 5.

Table 1 shows the defense is effective at detecting bounded attacks when used with a 3-layer verifier architecture. With an unprotected model, adversarial lanes slip by undetected 100% of the time. Under nonadaptive attack, the defense is very capable of detecting adversarial lanes while very rarely mistaking real lanes for adversarial. We do see some gains in attack strength under adaptive attack; however, we are still able to detect 90.2% of adversarial lanes while maintaining accurate classification of at least 95% of real lanes. Table 3 shows selected ROC curve data.

5.1.2 ResNet-18 Verifier

Table 2 shows that using a ResNet-18 architecture for the verifier improves the effectiveness of the defense under bounded attack. Examples of output binary seg-

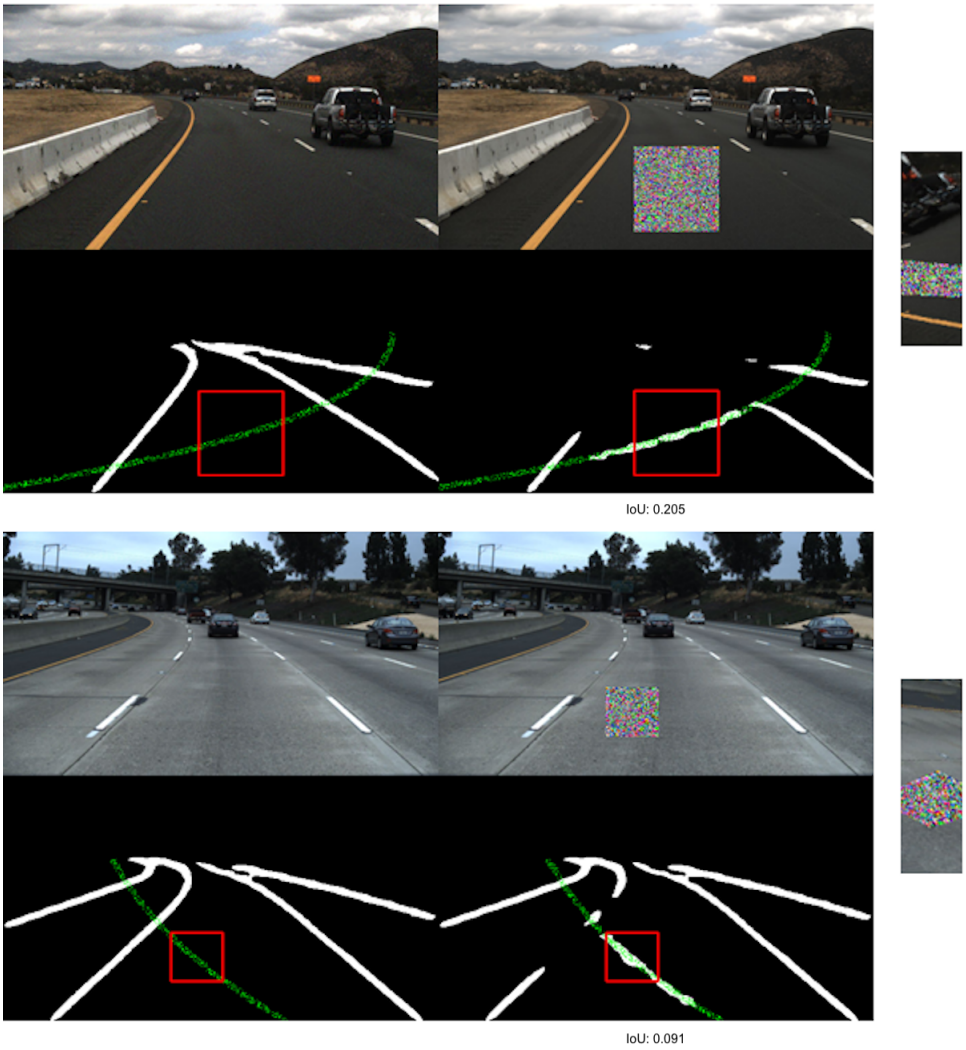

Figure 6. Examples of fixed and variable size adaptive patch attacks. Each group of four images has the original and applied patch on the first row, and the corresponding binary segmentation maps on the second. The target lane is dotted in green, with the patch's bounding box outlined in red. IoU values are below each applied patch segmentation map, with the resulting stabilized lane shown to the right of each group. While full scene control is limited, the results are substantial enough to cause rogue vehicle behavior.

mentation maps before and after adaptive attack are shown in Figure 7.

The ResNet-18 verifier yields a marked improvement over the 3-layer verifier, with detection of 99.3% of non-adaptive attacks and 98.1% of adaptive attacks. Classification accuracy on real lanes remains at least 95%. This demonstrates substantial benefits if constraints on model size are relaxed. Table 3 shows selected ROC curve data.

5.2. Patch Threat Model

Results for defense against the patch threat model are provided only for the 3-layer verifier architecture. Future work could involve assessing the attacks described on the ResNet-18 backbone.

5.2.1 Fixed Size

The fixed patch attack results reveal a strong reliance of LaneNet on spatially local features. Unlike the previous L-infinity attack, which could manipulate the entire scene to achieve its goal, the patch attack is unable to induce change outside of a small region around the patch location. An example of the attack can be found in the first half of Figure 6. Note that while the patch attack does not do well against our attack success metric of full scene control, the achieved result is still capable of causing undesired lane deviation.

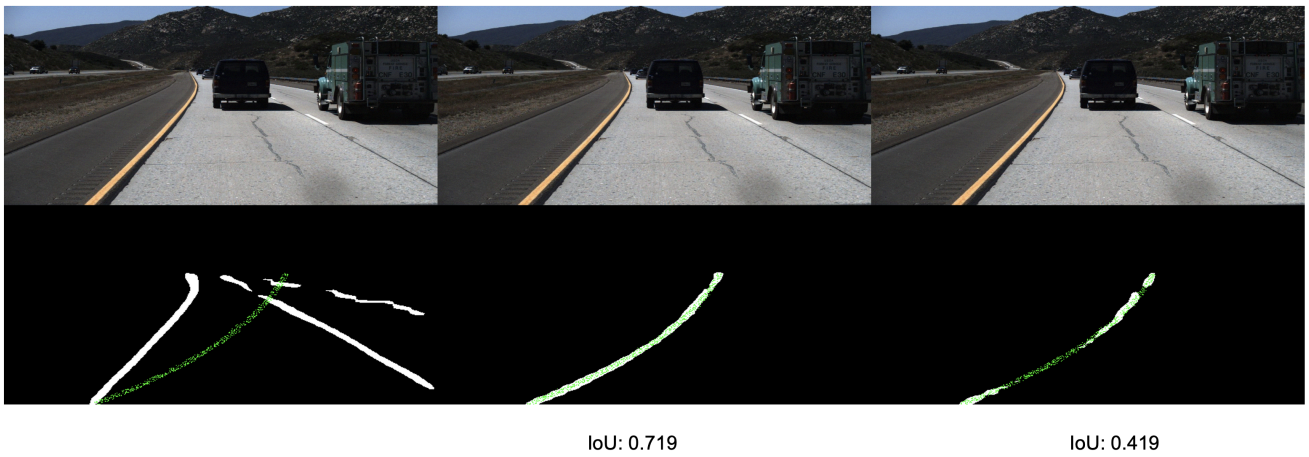

Figure 7. Example of a bounded attack with a ResNet-18 verifier. From left to right, the vertical pairs of images represent clean, nonadaptive, and ResNet-18 backbone adaptive attack, accompanied by the corresponding binary segmentation maps directly below. For each binary segmentation map, the target adversarial lane is superimposed in green, with an IoU value provided for attacked scenes.

Table 1 shows defense results under fixed patch attack. Similar to the bounded attack, we see strong detection rates for nonadaptive attack with a slight drop when subject to adaptive attack. Table 3 shows selected ROC curve data for the fixed size adaptive attack.

5.2.2 Variable Size

Variable patch attack success follows a similar trend to that of the fixed size patch attack, with the region of effect largely localized around the patch location. An example of the variable patch attack can be found in the second half of Figure 6.

Table 1 shows defense results under variable patch attack. Mirroring both previous attacks, robust nonadaptive attack detection rates are accompanied by weaker results when under adaptive attack. Variable patch sizes are generally smaller than the fixed patch, which is reflected in the stronger observed defense results. Table 3 shows selected ROC curve data for the adaptive attack.

5.3. Speed

The 3-layer varies has a marginal impact on pipeline inference time, reducing throughput from 29.8 to 27 frames per second. Due to differences in system configuration, we were unable to achieve the 50 frames per second as presented in [11], but as a rough estimate, the 3ms inference time of our defense as measured locally would translate to a drop from 52.6 to 45.5 frames per second using the timings provided by [11].

Table 2. The effectiveness of our defense with a ResNet-18 verifier. False positive rate (FPR) refers to real lanes the defense mistakenly flagged as adversarial, whereas false negative rate (FNR) refers to adversarial lanes the defense believed to be real. As a measure of post-defense attack success, the false negative (FN) average IoU is the average IoU between all attacked scenes and targets once flagged lanes have removed.

	Defense Metrics		*Attack Metric*
Bounded Attack	**FPR**	**FNR**	**FN IoU**
Unprotected	0	1	0.720
Nonadaptive	0.044	0.008	0.009
Adaptive	0.047	0.019	0.032

5.4. Ablation Studies

5.4.1 Adaptive Attack Cycling

Given that the second stage of the adaptive attack is unable to assess its impact on the first stage, it is possible that there exists adverse feedback between the two stages. We upper bound the extent of this adverse feedback by using the target adversarial lane in place of the perceived lane in the ResNet-18 based model. To evaluate the extent of this feedback, we test a bounded attack where the stages are cycled up to four times. Results are in the first half of Table 4. Comparing them with the figures in Table 1, we note that the false positive and false negative rates are slightly better for the defense in the cycled attack, suggesting that while repeated cycling may be helping first stage output, the effect is outweighed by an adverse impact to the efficacy of the second stage. Since polynomial fitting

Table 3. Selected defense ROC curve results. False positive rate (FPR) refers to real lanes the defense mistakenly flagged as adversarial, whereas false negative rate (FNR) refers to adversarial lanes the defense believed to be real. ↓ - lower is better ↑ - higher is better.

		3-Layer Arch., Bounded Attack	
		Nonadaptive	Adaptive
FNR at 0.01 FPR	↓	0.160	0.496
FNR at 0.02 FPR	↓	0.075	0.210
FNR at 0.05 FPR	↓	0.036	0.082
FNR at 0.10 FPR	↓	0.026	0.048
Area Under Curve	↑	0.984	0.971

		ResNet-18 Arch., Bounded Attack	
		Nonadaptive	Adaptive
FNR at 0.01 FPR	↓	0.024	0.054
FNR at 0.02 FPR	↓	0.016	0.042
FNR at 0.05 FPR	↓	0.007	0.017
FNR at 0.10 FPR	↓	0.005	0.008
Area Under Curve	↑	0.998	0.997

		3-Layer, Adaptive Patch Attack	
		Fixed Size	Variable Size
FNR at 0.01 FPR	↓	0.334	0.200
FNR at 0.02 FPR	↓	0.196	0.187
FNR at 0.05 FPR	↓	0.100	0.072
FNR at 0.10 FPR	↓	0.049	0.034
Area Under Curve	↑	0.976	0.973

cleans up much of the noise in the binary segmentation map, first stage gains may provide marginal benefits to overall attack strength.

5.4.2 Simpler Defense Architecture

We found that although the philosophy of our defense encourages simple defense designs, purely linear models struggle with the task of lane classification as shown in the second half of Table 4. This result hints again at the highly local nature of lane detection as previously seen in Figure 6, a property linear layers are less adept at taking advantage of compared to the convolutional layers used in both our 3-layer and ResNet-18 model.

6. Conclusion

Our proposed defense is a modular extension of existing lane detection models and can defend against adversarial attacks with minimal impact to underlying lane detection capabilities.

The orthogonal nature of the defense allows it to be trained independently from the underlying lane detection model, eliminating the cost of retraining. Amid the rising complexity of image processing models, our defense can provide security with very little overhead. Taking only lane locations as inputs, the defense does not depend on a particular lane detection model's features or assumptions, streamlining integration into any lane detection pipeline. The lightweight nature of the defense promotes fast inference and quick updates as new attacks arise.

Table 4. Ablation results. False positive rate (FPR) refers to real lanes the defense mistakenly flagged as adversarial, whereas false negative rate (FNR) refers to adversarial lanes the defense believed to be real. As a measure of post-defense attack success, the false negative (FN) average IoU is the average IoU between all attacked scenes and targets once flagged lanes have removed.

	Defense Metrics		*Attack Metric*
Cycled Attack	**FPR**	**FNR**	**FN IoU**
Bounded	0.042	0.044	0.030

	Defense Metrics		*Attack Metric*
Linear Defense	**FPR**	**FNR**	**FN IoU**
Bounded	0.049	0.966	0.720
Fixed Patch	0.045	0.879	0.180
Variable Patch	0.052	0.921	0.075

Under a bounded threat model that is able to fully take over the scene, we show that a simple 3-layer model employing our defense structure on top of LaneNet can detect over 90% of attacks while maintaining a maximum 5% impact to clean accuracy. Under a patch-based threat model where attacker control is limited but still capable of causing undesired lane deviation, our defense is able to identify 98% of abnormal activity while preserving the same 5% threshold. Relaxing constraints on model size, we find that a ResNet-18 backbone can boost bounded attack detection to over 98%. Thus, in such situations where the lane detection model would have otherwise passed adversarial lanes off as real to the autonomous control system, our model is able to call attention to potentially malicious actors.

Future work could involve designing a differentiable adaptive attack and collecting performance of our proposed defense on a larger sample of lane detection models. An application of transfer learning could be explored by training the defense on a separate dataset from the lane detection model and examining performance. While our defense is able to alert the vehicle to the presence of an attack, it does not provide guidance on the safest response, which we leave as an open question. Variants in the architecture of the verification system may also be worth investigating; for example, defense model enhanced with domain-specific knowledge of the highly local nature of lane detection

could see further improvement in verification accuracy. Finally, a similar system could prove useful for bolstering robustness against corruption.

References

[1] M. Bertozz, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: theory and applications," *Image and Vision Computing*, vol. 16, no. 8, pp. 585 – 590, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885697000930

[2] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," 2018.

[3] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured prediction models," 2017.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[6] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection cnns by self attention distillation," 2019.

[7] S. Jung, S. Choi, M. A. Khan, and J. Choo, "Towards lightweight lane detection by optimizing spatial embedding," 2020.

[8] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, "Key points estimation and point instance segmentation approach for lane detection," 2020.

[9] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[11] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: an instance segmentation approach," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 286–291.

[12] A. Nøkland, "Improving back-propagation by adding an adversarial gradient," 2016.

[13] T. Sato, J. Shen, N. Wang, Y. J. Jia, X. Lin, and Q. A. Chen, "Hold tight and never let go: Security of deep learning based automated lane centering under physical-world attack," 2020.

[14] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," 2020.

[15] TuSimple. [Online]. Available: http://benchmark.tusimple.ai/

[16] Y. Wang, X. Ma, J. Bailey, J. Yi, B. Zhou, and Q. Gu, "On the convergence and robustness of adversarial training," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 6586–6595. [Online]. Available: http://proceedings.mlr.press/v97/wang19i.html

[17] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. L. Yuille, "Adversarial examples for semantic segmentation and object detection," *CoRR*, vol. abs/1703.08603, 2017. [Online]. Available: http://arxiv.org/abs/1703.08603

Appendix

Verification Model Details

All verification models are trained normally and adversarially using the Adam optimizer with a learning rate of 0.001 and weight decay of 0.001. Each the model is trained normally for 100 epochs. The epoch with the highest validation accuracy is then adversarially trained for an additional 10 epochs with early stopping.

3-Layer Architecture

We use focal loss as the loss function for both normal and adversarial training, with $\gamma = 1$ and $\alpha = 0.75$ during normal training and $\gamma = 1$ and $\alpha = 0.01$ during adversarial training. During adversarial training, the model was trained on adversarial negative examples and clean positive examples, motivated by the goal of our verification model to detect adversarial lanes pretending to be real, but not real lanes pretending to be adversarial. The deemphasis of positive examples during adversarial training was determined to be helpful experimentally.

Adversarial training uses a bounded attack using random starts and $\epsilon = 8/255$. Number of attack iterations are 3 and 100 iterations for the patch and bounded attack defenses, respectively. Attack step size for both is computed as $2 * \epsilon/\texttt{attack iterations}$.

ResNet-18 Architecture

We use a pretrained implementation of ResNet-18 in PyTorch capable of accepting variable sized inputs, and modify the last linear layer to return a 1-dimensional output for binary classification.

Normal training is done using focal loss with parameters $\gamma = 1$ and $\alpha = 0.5$. Adversarial training is done on adversarial negative examples and clean positive examples, using a weight of 3 for positive examples during the binary cross-entropy step required to compute probabilities for focal loss. The attack uses random starts, $\epsilon = 8/255$, and 20 attack iterations, which we found sufficient for convergence. Attack step size is computed as $2 * \epsilon/\texttt{attack iterations}$.

All ResNet-18 training hyperparameters were determined experimentally.

Adaptive Attack Details

The end-to-end attack uses binary cross-entropy loss as its criterion. For the bounded attack, we use $\epsilon = 8/255$ with a maximum of 100 attack iterations for both the base attack on LaneNet and the adaptive attack on classifier. For both variable and patch attack, we use 100 attack iterations and an attack step size of 100 for both the base attack on LaneNet and the adaptive attack on classifier.