Towards Driving-Oriented Metric for Lane Detection Models

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

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

Abstract

After the 2017 TuSimple Lane Detection Challenge, its dataset and evaluation based on accuracy and F1 score have become the de facto standard to measure the performance of lane detection methods. While they have played a major role in improving the performance of lane detection methods, the validity of this evaluation method in downstream tasks has not been adequately researched. In this study, we design 2 new driving-oriented metrics for lane detection: End-to-End Lateral Deviation metric (E2E-LD) is directly formulated based on the requirements of autonomous driving, a core downstream task of lane detection; Per-frame Simulated Lateral Deviation metric (PSLD) is a lightweight surrogate metric of E2E-LD. To evaluate the validity of the metrics, we conduct a large-scale empirical study with 4 major types of lane detection approaches on the TuSimple dataset and our newly constructed dataset Comma2k19-LD. Our results show that the conventional metrics have strongly negative correlations ($\leq$-0.55) with E2E-LD, meaning that some recent improvements purely targeting the conventional metrics may not have led to meaningful improvements in autonomous driving, but rather may actually have made it worse by overfitting to the conventional metrics. As autonomous driving is a security/safety-critical system, the underestimation of robustness hinders the sound development of practical lane detection models. We hope that our study will help the community achieve more downstream task-aware evaluations for lane detection.

1. Introduction

Lane detection is one of the key technologies today for realizing autonomous driving. For lane detection, camera is the most frequently used sensor because it is a natural choice as lane lines are visual patterns [26]. Like most other computer vision areas, lane detection has been significantly benefited from the recent advances of deep neural networks (DNNs). In the 2017 TuSimple Lane Detection Challenge [8], DNN-based lane detection shows substantial performance as all top 3 teams opt for DNN-based lane detection. After this competition, its dataset and evaluation method based on accuracy and F1 score became the *de facto* standard in lane detection evaluation. These metrics are inherited by the subsequent datasets [13, 37].

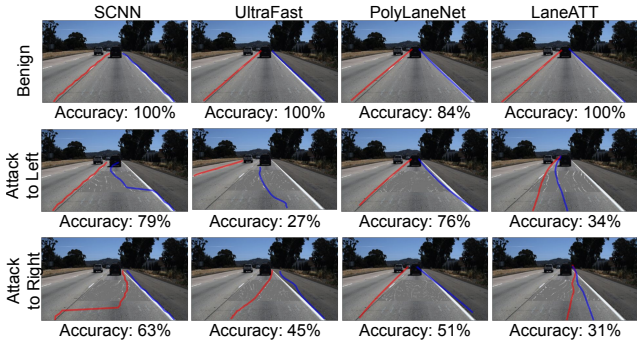


Figure 1. Examples of lane detection results and the accuracy metric in benign and adversarial attack scenarios on TuSimple Challenge dataset [8]. As shown, the conventional accuracy metric does not necessarily indicate drivability if used in autonomous driving, the core downsteam task. For example, SCNN always has higher accuracy than PolyLaneNet, but its detection results are making it much harder to achieve lane centering (detailed in §4.2).

However, the validity of this evaluation method in practical contexts, i.e., whether this is representative of practicality in real-world downstream applications, has not been adequately researched. Specifically, the main real-world applications of lane detection are for autonomous driving (AD), e.g., online detection for automated lane centering (for lower-level AD such as in Telsa AutoPilot [6]), and offline detection for high-definition map creation (for both low-level [5] and high-level AD [50]). With such an application domain as its main target, the robustness of lane detection is highly critical as errors from it could be fatal. Unfortunately, we find that the conventional evaluation metrics (i.e., accuracy and F1 score) have limitations in correctly reflecting the performance of lane detection models in such main downstream application domain, especially in more challenging scenarios (e.g., when under adversarial attacks). Fig. 1 shows a few such examples that motivate this study. In the adversarial attack settings, the lane lines detected by SCNN [37] are largely disrupted, but their performance measured by the conventional accuracy metric is

always higher than the one of PolyLaneNet [48], which are generally aligned with actual lane lines (and indeed lead to less lane center deviation than SCNN when used with driving models as quantified later in §4.2). In the benign settings, PolyLaneNet has the lowest accuracy and is underestimated, despite its seemingly perfect detection for humans. As lane detection has been evaluated using mainly relatively clean and homogeneous driver's view images, it is not easy to identify such a great discrepancy at the metric level. Considering the criticality of robust lane detection to correct and safe AD, it is important to address such a metric-level limitation since (1) the cornerstone of real-world deployment and commercialization of AD today is exactly on the handling of those more challenging driving scenarios [21, 30, 53]; and (2) with increasingly more discoveries of physical-world adversarial attack on lane detection in AD context [31, 44], it is desired to have a more downstream task-aware performance metric when judging the model robustness (and its enhancement).

Motivated by such critical needs, we design 2 new driving-oriented metrics, *End-to-End Lateral Deviation metric* (E2E-LD) and *Per-frame Simulated Lateral Deviation metric* (PSLD), to measure the performance of lane detection models in AD, especially in Automated Lane Centring (ALC), a Level-2 driving automation that automatically steers a vehicle to keep it centered in the traffic lane [7]. E2E-LD is designed directly based on the requirements of driving automation by ALC. PSLD is a lightweight surrogate metric of E2E-LD that estimates the impact of lane detection results on driving from a single frame. This per-frame lightweight design allows the metric to be usable during upstream lane detection model training. To evaluate the validity of the metrics, we conduct a large-scale empirical study of the 4 major types of lane detection approaches on the TuSimple dataset and our newly constructed dataset, Comma2k19-LD, which contains both lane line annotation and driving information. To simulate corner-case but physically-realizable scenarios as in Fig. 1 for lane detection, we utilize and extend physical-world adversarial attacks on ALC [44]. We formulate attack objective functions to fairly generate adversarial attacks against the 4 major types of lane detection approaches. Throughout this study, we find that the conventional metrics have strongly negative correlations ($r \leq$-0.55) with E2E-LD in the benign scenarios, meaning that some recent improvements purely targeting the conventional metrics may not have led to meaningful improvements in AD, but rather may actually have made it worse by overfitting to the conventional metrics. In the attack scenarios, while we observe a slight positive correlation ($r \leq$0.08), it is not statistically significant. Consequently, we find that the conventional metrics tend to overestimate less robust models. On the contrary, our newly-designed PSLD metric is always strongly positively correlated with E2E-LD ($r \geq$0.38), and all correlations are statistically significant ($p \leq 0.001$).

While the TuSimple Challenge dataset and its evaluation metrics have played a substantial role in developing performant lane detection methods, the recent improvement on the conventional metrics does not lead to the improvement on the core downstream task AD. We thus want to inform the community of such limitations of the conventional evaluation and facilitate research to conduct more downstream task-aware evaluation for lane detection, as the gap between upstream evaluation metrics and downstream application performance may hinder the sound development of lane detection methods for real-world application scenarios.

In summary, our contributions are as follows:

- We design 2 new driving-oriented metrics, E2E-LD and PSLD, that can more effectively measure the performance of lane detection models when used for AD, their core downstream task.
- We design a methodology to fairly generate physical-world adversarial attacks against the 4 major types of lane detection models.
- We build a new dataset Comma2k19-LD that contains lane annotations and driving information.
- We are the first to conduct a large-scale empirical study to measure the capability of 4 major types of lane detection models in supporting AD.
- We highlight and discuss the critical limitations of the conventional evaluation and demonstrate the validity of our new downstream task-aware metrics.

Code and data release. All our codes and datasets are available on our project websites [1].

2. Related Work

2.1. 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 [35, 47].

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 [8] (all top-3 winners adopt the segmentation approach [29, 36, 37]), it has been applied in many recent lane detection methods [28, 54, 55]. This segmentation approach is also used in the industry. A reverse-engineering study reveals that Tesla Model S adopts this segmentation-based approach [31]. 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

[1] https://github.com/ASGuard-UCI/ld-metric
https://sites.google.com/view/cav-sec/ld-metric

which is just background. Additionally, this approach requires a postprocessing step to extract the lane line curves from the pixel-wise classification result.

Row-wise classification approach. This approach [27, 35, 39, 52] 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 [39] reports that their method can work at more than 300 FPS with a comparable accuracy 95.87% on the TuSimple Challenge dataset [8]. On the other hand, SAD [28], 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 [38, 48] 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 [4]. 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. Additionally, prior work mentions that this approach is biased toward straight lines because the majority of lane lines in the training data are straight [48].

Anchor-based approach. Anchor-based approach [34, 40, 47] is inspired by region-based object detectors such as Faster R-CNN [42]. 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 [47] reports that it achieves a higher F1 score (96.77%) than the segmentation approaches (95.97%) [28, 37] on the TuSimple dataset.

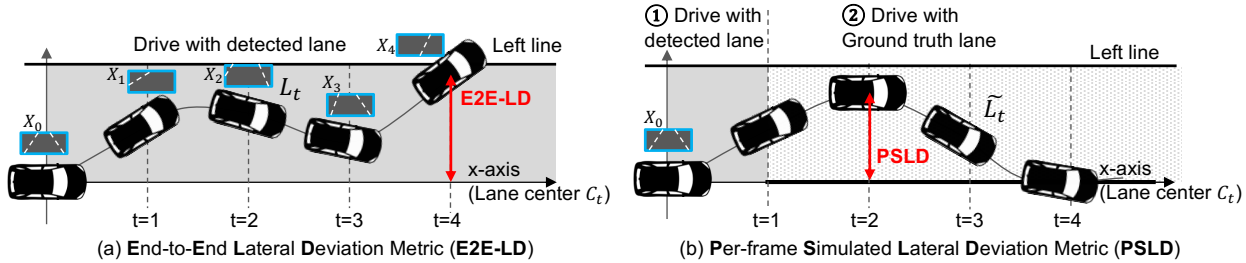
2.2. Evaluation Metrics for Lane Detection

All lane detection methods we discuss in §2.1 evaluate their performance on the *accuracy* and *F1 score* metrics used in the 2017 TuSimple Challenge [8]. The *accuracy* is calculated by $\sum_{i \in H} \frac{\mathrm{tp}_i}{|H|}$, where H is a set of sampled y-axis points in the driver's view image and tp_i is 1 if the difference of a predicted lane line point and the ground truth point at $y = i$ is within α pixels; otherwise is 0. α is set to 20 pixels in the TuSimple Challenge. The detected lane line is associated with a ground truth line with the highest accuracy. In other datasets [13, 37], IoU (Intersection over Union) is also used instead of accuracy. However, the ground-truth area is only defined as a 30-pixel wide line based on lane points, and this metric is almost equivalent to accuracy. The *F1 score* is a common metric to measure the performance of binary classification tasks. This is the harmonic mean of precision and recall: $\frac{2}{\mathrm{recall}^{-1}+\mathrm{precision}^{-1}}$. In the TuSimple Challenge, the precision and recall are calculated at the lane line level: The precision is the true positive ratio of detected lane lines and the recall is the true positive ratio of ground truth lines. The true positive is defined if the accuracy of a pair of the ground truth line and detected line is $\geq \beta$. β is set to 0.85 in the TuSimple Challenge. Although the *accuracy* and *F1 score* can measure the capability of lane detection at a certain level, these metrics do not fully represent the performance in the main real-world downstream application, AD [5, 6, 50], as concretely shown later in §4.2.

Specifically, to reflect its performance if used in AD, or *drivability*, accuracy and F1 score metrics have 2 major limitations: **(1)** There is no justification of $\alpha = 20$ pixels and $\beta = 0.85$ accuracy thresholds. For example, the ALC system can keep at the lane center even if the detection error is more than 20 pixels, as long as the detected lane lines are *parallel* with actual lane lines. Furthermore, the importance of detected lane line points should not be equal, i.e., closer points to the vehicle should be more important than the distanced points to control a vehicle. **(2)** The current metrics treat all lane lines in the driver's view equally, e.g., detection errors for the ego lane's left line are treated the same as the detection errors for the left lane's left line. However, the former is much more important to ALC systems than the latter, as the former can directly impact the downstream calculation of the lane center. For example, if a model cannot detect the left lane's left line but can still detect the ego lane's left line, it won't affect its use for ALC. However, if it cannot detect the latter but can detect the former, the accuracy metric remains the same but the downstream modules in ALC may consider the left lane's left line as ego lane's left line and thus mistakenly deviate to the left.

2.3. Automated Lane Centering

Automated Lane Centering (ALC) is a Level-2 driving automation technology that automatically steers a vehicle to keep it centered in the traffic lane [7]. Recently, ALC is widely adopted in various vehicle models such as Tesla [6] and thus one of the most popular downstream applications of lane detection. Typical ALC systems [4, 10, 33] operate in 3 modules: lane detection, lateral control, and vehicle actuation. More details of ALC are in the supplementary materials (Appendix G). While there is a line of research that designs end-to-end DNNs for ALC or higher driving automation [12, 14, 16], the current industry-standard solutions



(a) **E**nd-to-**E**nd **L**ateral **D**eviation Metric (**E2E-LD**) (b) **P**er-frame **S**imulated **L**ateral **D**eviation Metric (**PSLD**)

Figure 2. Overview of our driving-oriented metrics for lane detection models: E2E-LD and PSLD. X_t are camera frames from driver's view (lane detection model inputs). E2E-LD requires multiple (consecutive) camera frames, while PSLD only uses the current frame X_0.

adopt such a modular design to ensure accountability and safety. In the lateral control, ALC plans to follow the lane center as waypoints with Proportional-Integral-Derivative (PID) [20] or Model Predictive Control (MPC) [43].

Adversarial Attack on ALC. After researchers found DNN models generally vulnerable to adversarial attacks [24, 46], the following work further explored such attacks in the physical world [15, 23]. A recent study demonstrates that ALC systems are also vulnerable to physical-world adversarial attacks [44]. Their attack, dubbed Dirty Road Patch (DRP) attack, targets industry-grade DNN-based ALC systems, and is designed to be robust to the vehicle position and heading changes caused by the attack in the earlier frames. We use the DRP attack to simulate challenging but realizable scenarios in our evaluations.

3. Methodology

In this section, we motivate the design of 2 new downstream task-aware metrics to measure the performance of lane detection models in ALC. To evaluate the validity of the metrics even in challenging scenarios, we formulate attack objective functions to fairly generate adversarial attacks against the 4 major types of lane detection methods.

3.1. End-to-End Lateral Deviation Metric

As the name of ALC indicates, the performance of ALC should be evaluated by how accurately it can drive in the lane center, i.e., the *lateral* (left or right) deviation from the lane center. In particular, the maximum lateral deviation from the lane center in continuous closed-loop perception and control is the ultimate downstream-task performance metric for lane detection. Such deviation is directly safety-critical as large lateral deviations can cause a fatal collision with other driving vehicles or roadside objects. We call it *End-to-End Lateral Deviation metric* **(E2E-LD)**, shown in Fig. 2 (a). The E2E-LD at $t = 0$ is obtained as follows.

$$\max_{t \leq T_E}(|L_t - C_t|) \tag{1}$$

, where L_t is the lateral (y-axis) coordinate of the vehicle at t. C_t is the lane center lateral (y-axis) coordinate corresponding to the vehicle position at t. We use the vehicle coordinate system at $t = 0$. T_E is a hyperparameter to decide the time duration. If $T_E = 1$ second, the E2E-LD is the largest deviation within one second. To obtain L_t, it requires a closed-loop mechanism to simulate a driving by ALC, such as AD simulators [3, 22]. Starting from $t = 0$, the vehicle position and heading at $t = 1$ is calculated based on the camera frame at $t = 0$ (X_0): The lane detection model detects lane lines from the frame, the lateral control interprets it by a steering angle, and vehicle actuation operates the steering wheel. This procedure repeats until $t = T_e$. Hence, *multiple* (consecutive) camera frames $X_0, ..., X_{T_E}$ are required and they are dynamically changed based on the lane detection results in the earlier frames.

However, such AD simulations are too computationally expensive for large-scale evaluations. Thus, we simulate vehicle trajectories by following prior work [44], which combines vehicle motion model [41] and perspective transformation [25, 49] to dynamically synthesize camera frames from existing frames according to a driving trajectory.

3.2. Per-Frame Simulated Lateral Deviation Metric

The E2E-LD metric is defined as the desired metric based on the requirements of downstream task ALC. However, it is still too computationally intensive to be monitored during training of the upstream lane detection model. This overhead is mainly due to the camera frame inter-dependency that the camera frames are dynamically changed based on the lane detection results in the earlier frames. To address this limitation, we design the *Per-Frame Simulated Lateral Deviation metric* **(PSLD)**, which simulates E2E-LD only with a *single* camera input at the current frame (X_0) and the geometry of the lane center.

The overview of PSLD is shown in Fig. 2 (b). The calculation consists of two stages: ① update the vehicle position with the current camera frame at $t = 0$ (X_0) and its lane detection result, and ② apply the closed-loop simulation using the ground-truth lane center as waypoints from $t = 1$ to $t = T_p$. Note that we do not need camera frames in ② as the vehicle just tries to follow the ground-truth waypoints with lateral control, i.e., we bypass the lane detection assuming we know the ground-truth in $t \geq 1$. We then take the maximum lateral deviation from the lane center as a metric as with E2E-LD. For convenience, we normalize the maximum lateral deviation by T_p to make it a *per-frame* metric. The definition of PSLD is as follows:

$$\frac{1}{T_p} \max_{1 \le t \le T_p} (|\widetilde{L}_t - C_t|) \tag{2}$$

, where the $\widetilde{L}_t$ is the simulated lateral (y-axis) coordinate of the vehicle at t. For example, for $T_p = 1$, it is just a single-step simulation with the current lane detection result. The longer T_p can simulate the tailing effect of the current frame result in the later frames, but it may suffer from accumulated errors. In §4.3, we explore which T_p achieves the best correlation between PSLD and E2E-LD. More details are in the supplementary material (Appendix A).

3.3. Attack Generation

In this study, we utilize and extend physical-world adversarial attacks to evaluate the robustness of the lane detection system against challenging but realizable scenarios. To fairly generate adversarial attacks for all 4 major types of lane detection methods, we design an attack objective that can be commonly applicable to them. We name it the *expected road center*, 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 as follows.

Segmentation & row-wise classification approaches:

$$\frac{1}{L \cdot H} \sum_{l=1}^{L} \sum_{i=1}^{W} \sum_{j=1}^{H} i \cdot P_{ij}^{l} \tag{3}$$

, where H and W are the height and width of probability map, L is the number of probability maps (channels), and P_{ij}^{l} is the lane line existence probability of the pixel in the (i, j) element of the probability map.

Curve-fitting approach:

$$\frac{1}{L \cdot |\mathcal{H}|} \sum_{l=1}^{L} \sum_{j \in \mathcal{H}} [j^d, j^{d-1}, \cdots, j, 1] p_l \tag{4}$$

, where L is the number of detected lane lines, d is the degrees of polynomial ($d = 3$ used in PolyLaneNet [48]), $\mathcal{H}$ is a set of sampled y-axis values, and $p_l \in \mathbb{R}^{d+1}$ is the coefficient of detected lane line l.

Anchor-based approach:

$$\sum_{l \in \mathcal{A}} \left[\frac{1}{|\Delta^l|} \sum_{j \in \Delta^l} (a_j^l + \delta_j^l) \right] \cdot \pi^l \tag{5}$$

, where $\mathcal{A}$ is a set of the anchor proposals, Δ^l is an index set of y-axis value for anchor proposal l, π^l is the probability of anchor proposal l, and a_j^l and δ_j^l are the x-axis value and its offset of anchor proposal l at y-axis index j respectively.

We incorporate this expected road center functions into DRP attack [44] procedure to generate adversarial attacks that are effective for multiple frames.

Table 1. Target lane detection methods. *Acc.* is the accuracy of the TuSimple Challenge dataset [8] in the reference papers.

Approach	Selected Method	Acc.
Segmentation	SCNN [37]	96.53%
Row-wise classification	UltraFast (ResNet18) [39]	95.87%
Curve-fitting	PolyLaneNet (b0) [48]	88.62%
Anchor-based	LaneATT (ResNet34) [47]	95.63%

4. Experiments

We conduct a large-scale empirical study to evaluate the validity of the conventional metrics and our PLSD by comparing them with the ultimate downstream-task performance metric E2E-LD. We evaluate the 4 major types of lane detection approaches. We select a representative model for each approach as shown in Table 1. The pretrained weights of all models are obtained from the authors' or publicly available websites[2]. All pretrained weights are trained on the TuSimple Challenge training dataset [8].

4.1. Conventional Evaluation on TuSimple Dataset

Evaluation Setup. We first evaluate the lane detection models with the conventional accuracy and F1 score metrics on the TuSimple dataset [8], which has 2,782 one-second-long video clips as test data. Each clip consists of 20 frames, and only the last frame is annotated and used for evaluation. We randomly select 30 clips from the test data. For each clip, we consider two attack scenarios: attack to the left, and to the right. Thus, in total, we evaluate 60 different attack scenarios. In each scenario, we place 3.6 m x 36 m patches 7 m away from the vehicle as shown in Fig. 1. To know the world coordinate, we manually calibrate the camera matrix based on the size of lane width and lane marking. To deal with the limitation (2) discussed in §2.2, we remove lane lines other than the ego-left and ego-right lane lines to evaluate the applicability to ALC systems more correctly. More details of each attack implementation and parameters are in the supplementary materials (Appendix B).

Results. Table 2 shows the accuracy and F1 score metrics in the benign and attacks scenarios. In the benign scenarios, LaneATT has the best accuracy (94%) and F1 score (88%). SCNN and UltraFast show also high accuracy and F1 score while UltraFast has the lowest F1 score (8%) in the attack scenarios. PolyLaneNet has lower accuracy and F1 score than the others in both benign and attack scenarios. These results are generally consistent with the reported performance as in Table 1. However, when we visually look into the detected lane lines under attack, we find quite some cases suggesting vastly different conclusions if used in AD as the downstream task. For example, as shown in Fig. 1,

[2] 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

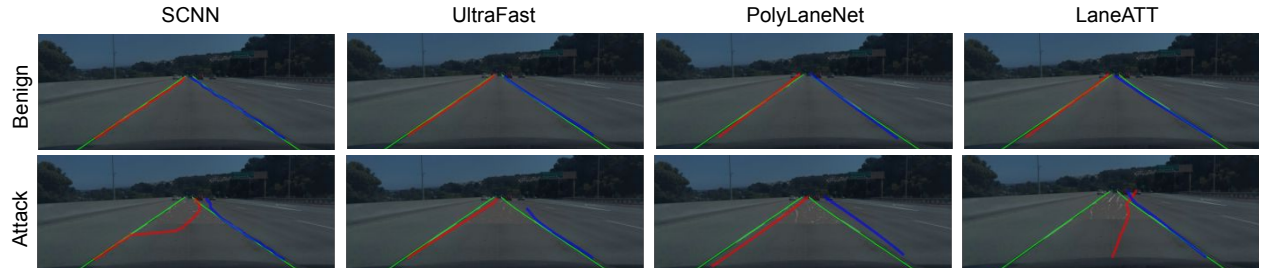


Figure 3. Examples of the benign and attack-to-the-right scenarios on the Comma2k19-LD dataset. The red, blue, and green lines are the detected left and right lines and the ground-truth lines respectively.

Table 2. Accuracy and F1 scores for attack and benign cases on the TuSimple Challenge dataset. The metrics are calculated only with ego left and right lanes. The **bold** and underlined letters mean the highest and lowest scores, respectively, among the 4 lane detection methods. The higher score means the higher performance.

	Accuracy		F1 Score	
	Benign	Attack	Benign	Attack
SCNN [37]	89%	**58%**	75%	28%
UltraFast [39]	87%	36%	77%	8%
PolyLaneNet [48]	72%	53%	50%	19%
LaneATT [47]	**94%**	51%	**88%**	**29%**

even though SCNN has the highest accuracy in all three scenarios, its detected lane lines are heavily curved by the attack. In contrast, the detection of PolyLaneNet looks like the most robust among the 4 models, as the detected lane lines are generally parallel to the actual lane lines. However, its accuracy (63%) is smaller than the one of SCNN (51%) in the attack to the right scenario. In the benign scenario, PolyLaneNet has a lower accuracy (16% margin) than the others, but it is hard to find meaningful differences for humans as the detected lines are well-aligned with actual lane lines. We provide more examples in the supplementary material (Appendix G). Hence, the conventional accuracy and F1 score-based evaluation may not be well suitable to judge the performance of the lane detection model in representative downstream tasks such as AD.

4.2. Consistency of TuSimple Metrics with E2E-LD

To more systematically evaluate the consistency of the conventional accuracy and F1 score with the performance in AD as the downstream tasks, we conduct a large-scale empirical study on our newly-constructed dataset.

New Dataset: Comma2k19-LD. To evaluate both the conventional metrics and the downstream task-centric metrics E2E-LD and PSLD on the same dataset, we need both lane line annotations and driving information (e.g., position, steering angle, and velocity). Unfortunately, there is no existing dataset that satisfies the requirements to our best knowledge. Thus, we create a new dataset, coined *Comma2k19-LD*, in which we manually annotate the left and right lane lines for 2,000 frames (100 scenarios of 1-second clips at 20 Hz). The selected scenarios are randomly selected from the scenarios with more than 30 mph ($\approx$ 48 km/h) in the original Comma2k19 dataset [45]. Fig 3 shows the example frames of the Comma2k19-LD dataset. These frames are the first frames of the scenario. The following 20 frames are also annotated and the same patch is used for each attack. More details are in our supplementary materials (Appendix C). The Comma2k19-LD dataset is published on our website [11].

Evaluation Setup. We conduct the evaluation on the Comma2k19-LD dataset. For the attack generation, we attack to the left in randomly selected 50 scenarios and attack to the right in the other 50 scenarios. For the lateral control, we use the implementation of MPC [43] in OpenPilot v0.6.6, which is an open-source production ALC system. For the longitudinal control, we used the velocity in the original driving trace. For the motion model, we adopt the kinematic bicycle model [32], which is the most widely-used motion model for vehicles [2, 32, 51]. The vehicle parameters are from Toyota RAV4 2017 (e.g., wheelbase), which is used to collect the traces of the comma2k19 dataset. To make the model trained on the TuSimple dataset work on the Comma2k19-LD dataset, we manually adjust the input image size and field-of-view to be consistent with the TuSimple dataset. We place a 3.6 m x 36 m patch at 7 m away from the vehicle at the first frame. For the E2E-LD metric, we use T_E = 20 frames (1 second). It follows the result that the average attack success time of the DRP attack is nearly 1 sec [44]. More setup details are in the supplementary materials (Appendix B, D, and G).

Results. Table 3 shows the evaluation results of conventional accuracy and F1 score and E2E-LD. We calculate the Pearson correlation coefficient r and its p value. As shown, there are *substantial inconsistencies between the downstream-task performance (from the heavy-weight E2E-LD metric) and the conventional metrics*. In the benign scenarios, SCNN has the highest accuracy (0.59) and F1 score (0.84) under the original parameters ($\alpha = 20, \beta = 0.85$). However, SCNN is one of the methods with the *lowest* E2E-LD (0.21), and instead UltraFast has the highest E2E-LD (0.18). In the attack scenarios, the inconsistency is more obvious: PolyLaneNet has the highest E2E-LD (0.38), but PolyLaneNet achieves the 2nd lowest accuracy (0.59) and the highest F1 score (0.13) with the original parameters. Hence, the E2E-LD draws quite different conclusions

Table 3. Evaluation results of the E2E-LD and the conventional metrics, accuracy and F1 in the benign and attack scenarios. For each metric, the corresponding Pearson correlation coefficient with E2E-LD in the bottom rows. The original parameters are the ones used in the TuSimple challenge. The best parameters are those that have the highest correlation between E2E-LD with respect to F1 score. The **bold** and underlined letters indicate the highest and lowest performance or correlation, respectively.

		Benign					Attack				
			Original Parameters ($\alpha = 20, \beta = 0.85$)		Best Parameters ($\alpha = 5, \beta = 0.9$)			Original Parameters ($\alpha = 20, \beta = 0.85$)		Best Parameters ($\alpha = 50, \beta = 0.65$)	
		E2E-LD [m]	Accuracy	F1	Accuracy	F1	E2E-LD [m]	Accuracy	F1	Accuracy	F1
Metric	SCNN [37]	0.21	**0.93**	**0.84**	**0.59**	0.03	0.48	**0.68**	**0.31**	**0.83**	0.76
	UltraFast [39]	**0.18**	0.92	0.81	0.55	**0.10**	0.58	0.60	0.21	0.82	**0.77**
	PolyLaneNet [48]	0.20	0.78	0.50	0.44	0.01	**0.38**	0.59	0.13	0.81	0.76
	LaneATT [47]	0.21	0.89	0.75	0.54	0.06	0.72	0.51	0.14	0.66	0.48
Corr.	SCNN [37]	-	-0.65***	-0.60***	**-0.33*****	-0.13^{ns}	-	-0.13^{ns}	**-0.06**^{ns}	-0.14^{ns}	-0.06^{ns}
	UltraFast [39]	-	-0.58***	-0.59***	-0.38***	-0.24*	-	-0.24*	-0.14^{ns}	-0.20*	-0.13^{ns}
	PolyLaneNet [48]	-	-0.60***	**-0.55*****	-0.46***	**0.10**^{ns}	-	-0.27**	-0.28**	-0.06^{ns}	**0.01**^{ns}
	LaneATT [47]	-	**-0.57*****	-0.58***	-0.34***	-0.14^{ns}	-	**0.08**^{ns}	-0.09^{ns}	**0.11**^{ns}	0.12^{ns}

^{ns} Not Significant ($p > 0.05$),* $p \leq 0.05$,** $p \leq 0.01$,*** $p \leq 0.001$

from the conventional metrics. If we adopt the conventional metrics, SCNN should be preferred as the best performant model. This is consistent with the results in Table 1 and §4.1 since SCNN, UltraFast, and LaneATT show close performance in the conventional metrics (SCNN may have slight advantages in Comma2k19-LD). On the other hand, if we adopt E2E-LD, PolyLaneNet should be preferred since there is only a slight difference between the 4 lane detection methods in the benign scenarios and PolyLaneNet clearly outperforms the other methods in the attack scenarios.

The inconsistency between the E2E-LD and the conventional metrics can be more systematically quantified using Pearson correlation coefficient r. Generally, the E2E-LD and the conventional metrics have strongly *negative* correlations ($r \leq$-0.55) with high statistical significance ($p \leq 0.001$), meaning that some recent improvements in the conventional metrics may not have led to improvements in AD, but rather may have made it worse by overfitting to the metrics. SCNN, the segmentation approach, is the only one that does not use domain knowledge, e.g., lane lines are smooth lines (§2.1). This high degree of freedom in the model may lead to overfitting of the human annotations with noise.

Finally, we evaluate the parameters in the conventional metrics: α for the accuracy and β for F1 score. For α, we explore every 5 pixels from 5 pixels to 50 pixels. For β, we explore every 0.05 from 0.5 to 0.9. In the benign scenarios, ($\alpha = 20, \beta = 0.85$) has the best correlation between the E2E-LD and F1 score. In the attack scenarios, ($\alpha = 50, \beta = 0.65$) has the best correlation between the E2E-LD and F1 score. However, the results are still similar to those using the original parameters: SCNN shows the highest accuracy; UltraFast has a higher F1 score than the others, but the correlation is still negative. Thus, such a naive parameter tuning does not resolve the limitations of the conventional metrics.

4.3. Consistency of E2E-LD with PSLD

In this section, we evaluate the validity of PSLD as a *per-frame* surrogate metric of E2E-LD.

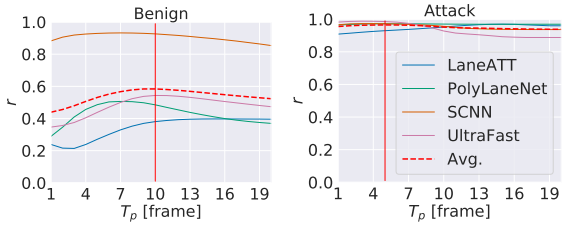


Figure 4. Pearson correlation coefficient r between E2E-LD and PSLD when T_p is varied from 1 to 20 in the benign and attack scenarios. The red vertical lines are T_p with the largest average r.

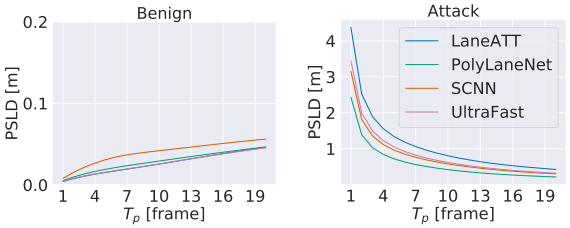


Figure 5. PSLD for the 4 major lane detection models when T_p is varied from 1 to 20 frames in the benign and attack scenarios.

Evaluation Setup. We follow the same setup as in §4.2. We generate the DRP attacks for 100 scenarios in the Comma2k19-LD dataset with the same parameters. For the PSLD, we obtain the ground truth waypoints by the following procedure. We generate a trajectory with the bicycle model and OpenPilot's MPC by using the human driving trajectory as waypoints. We then use the generated trajectory as a ground-truth road center. While we can directly use the human driving trajectory as ground truth, human driving sometimes is not smooth and this approach can cancel the effect of motion models, which have differences from real vehicle dynamics. For the benign scenarios, we calculate the PSLD for each frame in the original human driving. For the attack scenarios, we use the frames synthesized by the method described in 3.1 instead of the original frames because the attacked trajectory and its camera frames are largely changed from the original human driving. For example, to obtain the PSLD at frame $t = N$, we simulate the trajectory until $t = N - 1$ and we then calcu-

Table 4. Evaluation results of the E2E-LD and PSLD in the benign and attack scenarios. The format is the same as Table 3.

		Benign		Attack	
		E2E-LD [m]	PSLD [m]	E2E-LD [m]	PSLD [m]
Metric	SCNN [37]	0.21	0.04	0.48	0.58
	UltraFast [39]	**0.18**	**0.03**	0.58	0.62
	PolyLaneNet [48]	0.20	**0.03**	**0.38**	**0.42**
	LaneATT [47]	0.21	**0.03**	0.72	0.80
Corr.	SCNN [37]	-	**0.93*****	-	0.96***
	UltraFast [39]	-	0.54***	-	0.93***
	PolyLaneNet [48]	-	0.49***	-	**0.97*****
	LaneATT [47]	-	0.38***	-	0.95***

$^{n.s}$ Not Significant ($p > 0.05$),* $p \leq 0.05$,** $p \leq 0.01$,*** $p \leq 0.001$

late the PSLD with the synthesized frame at $t = N$.

Results. Fig. 4 shows the Pearson correlation coefficient r between E2E-LD and PSLD when T_p is varied from 1 to 20 frames. As shown, the E2E-LD, PSLD has strong positive correlations in both benign and attack scenarios. In particular, there are significant correlations (>0.8) in the attack scenarios. This is because the direction of lateral deviation generally coincides with the attack direction. By contrast, in the benign scenarios, the vehicle drives around the road center with overshooting, and thus the direction of lateral deviation heavily depends on the initial states. Nevertheless, the PSLD has always high positive correlations with E2E-LD (>0.2). In particular, SCNN has strong correlations (>0.8) with E2E-LD in all T_p. We consider that the high correlation can be due to the segmentation approach, which is the only method among the 4 methods that do not use the domain-specific knowledge the lane lines are generally smooth (§2.1). The detection of SCNN at the same location tends to be consistent across different frames, i.e., SCNN is less dependent on global information.

Finally, we explore the best T_p for PSLD to proxy E2E-LD. As shown in Fig. 4, the average of the correlation coefficients of the 4 methods achieves the maximum at $T_p = 10$ in the benign scenarios and $T_p = 5$ in the attack scenarios respectively. We list the E2E-LD and PLSD with $T_p = 10$ and the corresponding r in Table 4. As shown, there are strong, statistically significant ($p \leq 0.001$) positive correlations (≥ 0.38) between E2E-LD and PSLD in both cases. The results strongly support the fact that PSLD can measure the performance of lane detection in ALCs based solely on the single camera frame and ground-truth road center geometry. We note that the PSLD is not so sensitive to the choice of T_p. As shown in Fig. 5, the magnitude relation of the 4 methods is generally consistent for all T_p.

5. Discussion

Alternative Metric Design. To improve the existing metrics, we explored other possible design choices. One of the most intuitive approaches is the $\mathcal{L}_1$ or $\mathcal{L}_2$ distance in the bird's eye view. We evaluated the designs and confirmed that these metrics are still leading to erroneous judgment on downstream AD performance similar to the conventional metrics. Details are in the supplementary materials (Appendix F). We note that our metrics are specific to AD, the main downstream task of lane detection. For other downstream tasks, other metric designs can be more suitable.

Domain Shift. In this work, we use lane detection models pretrained on the TuSimple dataset and evaluate them on the Comma2k19-LD. To evaluate the impact of domain shift, we conduct further evaluation and confirm that our observations are generally consistent. Detailed results and discussions are in the supplementary material (Appendix E).

Closed-loop Simulation. To obtain driving-oriented metrics, there are multiple parameters and design choices in the closed-loop simulation. In this study, we follow the parameters in the Comma2k19 datasets and select simple and popular designs, e.g., bicycle model and MPC. Meanwhile, we think that such design differences should only have minor effects on our observations because ALC, Level-2 driving automation, just follows the lane center line, which is designed to be smooth on normal roads.

Evaluation on Other Datasets. Our metrics are applicable to any dataset set that contains position data (e.g. GPS) and its camera frames, but ideally, velocity and ground-truth lane centers should be available. Such information is available in relatively new datasets such as [9, 17]. However, lane annotations are not directly available in the datasets and require considerable effort to obtain from map data and camera frames. To our knowledge, our Comma2k19-LD is so far the only dataset with both lane line annotation and driving information. We hope our work will facilitate further research to build datasets including them.

6. Conclusion

In this work, we design 2 new lane detection metrics, E2E-LD and PSLD, which can more faithfully reflect the performance of lane detection models in AD. Throughout a large-scale empirical study of the 4 major types of lane detection approaches on the TuSimple dataset and our new dataset Comma2k19-LD, we highlight critical limitations of the conventional metrics and demonstrate the high validity of our metrics to measure the performance in AD, the core downstream task of lane detection. In recent years, a wide variety of pretrained models have been used in many downstream application areas such as AD [1], natural language processing [19], and medical [18]. Reliable performance measurement is essential to facilitate the use of machine learning responsibly. We hope that our study will help the community make further progress in building a more downstream task-aware evaluation for lane detection.

Acknowledgments

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

References

[1] Baidu Apollo. https://github.com/ApolloAuto/apollo. 8

[2] Introduction to Self-Driving Cars. https://www.coursera.org/learn/intro-self-driving-cars. 6

[3] LGSVL Simulator: An Autonomous Vehicle Simulator. https://github.com/lgsvl/simulator/. 4

[4] OpenPilot: Open Source Driving Agent. https://github.com/commaai/openpilot. 3

[5] Super Cruise - Hands Free Driving — Cadillac Ownership. https://www.cadillac.com/world-of-cadillac/innovation/super-cruise. 1, 3

[6] Tesla Autopilot. https://www.tesla.com/autopilot. 1, 3

[7] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *SAE International,(J3016)*, 2016. 2, 3

[8] TuSimple Lane Detection Challenge. https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection, 2017. 1, 2, 3, 5

[9] Waymo Open Dataset: An Autonomous Driving Dataset. https://www.waymo.com/open, 2019. 8

[10] 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. 3

[11] Comma2k19 LD. https://www.kaggle.com/tkm2261/comma2k19-ld, 2022. 6

[12] Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. In *RSS*, 2019. 3

[13] Karsten Behrendt and Ryan Soussan. Unsupervised Labeled Lane Marker Dataset Generation Using Maps. In *IEEE International Conference on Computer Vision*, 2019. 1, 3

[14] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to End Learning for Self-Driving Cars. *arXiv preprint arXiv:1604.07316*, 2016. 3

[15] Tom Brown, Dandelion Mane, Aurko Roy, Martin Abadi, and Justin Gilmer. Adversarial Patch. *arXiv preprint arXiv:1712.09665*, 2017. 4

[16] Sergio Casas, Abbas Sadat, and Raquel Urtasun. MP3: A Unified Model to Map, Perceive, Predict and Plan. In *CVPR*, 2021. 3

[17] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D Tracking and Forecasting with Rich Maps. In *CVPR*, 2019. 8

[18] Sihong Chen, Kai Ma, and Yefeng Zheng. Med3d: Transfer Learning for 3D Medical Image Analysis. *arXiv preprint arXiv:1904.00625*, 2019. 8

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019. 8

[20] Richard C Dorf and Robert H Bishop. *Modern Control Systems*. Pearson, 2011. 4

[21] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *CoRL*, 2017. 2

[22] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *CoRL*, pages 1–16, 2017. 4

[23] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical Adversarial Examples for Object Detectors. In *WOOT*, 2018. 4

[24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572*, 2014. 4

[25] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2003. 4

[26] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent Progress in Road and Lane Detection: A Survey. *Machine vision and applications*, 25(3):727–745, 2014. 1

[27] Yuenan Hou, Zheng Ma, Chunxiao Liu, Tak-Wai Hui, and Chen Change Loy. Inter-Region Affinity Distillation for Road Marking Segmentation. In *CVPR*, 2020. 3

[28] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In *CVPR*, 2019. 2, 3

[29] Yen-Chang Hsu, Zheng Xu, Zsolt Kira, and Jiawei Huang. Learning to Cluster for Proposal-Free Instance Segmentation. In *IJCNN*, 2018. 2

[30] Ashesh Jain, Luca Del Pero, Hugo Grimmett, and Peter Ondruska. Autonomy 2.0: Why Is Self-Driving Always 5 Years Away? *arXiv preprint arXiv:2107.08142*, 2021. 2

[31] Pengfei Jing, Qiyi Tang, Yuefeng Du, Lei Xue, Xiapu Luo, Ting Wang, Sen Nie, and Shi Wu. Too Good to Be Safe: Tricking Lane Detection in Autonomous Driving with Crafted Perturbations. In *USENIX Security*, 2021. 2

[32] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. In *IV*, 2015. 6

[33] Jin-Woo Lee and Bakhtiar Litkouhi. A Unified Framework of the Automated Lane Centering/Changing Control for Motion Smoothness Adaptation. In *ITSC*, 2012. 3

[34] Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-CNN: End-to-End Traffic Line Detection with Line Proposal Unit. *ITSC*, 2019. 3

[35] Lizhe Liu, Xiaohao Chen, Siyu Zhu, and Ping Tan. CondLaneNet: A Top-To-Down Lane Detection Framework Based on Conditional Convolution. In *ICCV*, 2021. 2, 3

[36] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards End-to-End Lane Detection: An Instance Segmentation Approach. In *IV*, 2018. 2

[37] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as Deep: Spatial CNN for Traffic Scene Understanding. In *AAAI*, 2018. 1, 2, 3, 5, 6, 7, 8

[38] Jonah Philion. FastDraw: Addressing the Long Tail of Lane Detection by Adapting a Sequential Prediction Network. In *CVPR*, 2019. 3

[39] Qin, Zequn and Wang, Huanyu and Li, Xi. Ultra Fast Structure-Aware Deep Lane Detection. In *ECCV*, 2020. 3, 5, 6, 7, 8

[40] Zhan Qu, Huan Jin, Yang Zhou, Zhen Yang, and Wei Zhang. Focus on Local: Detecting Lane Marker From Bottom Up via Key Point. In *CVPR*, 2021. 3

[41] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011. 4

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 3

[43] Richalet, J. and Rault, A. and Testud, J. L. and Papon, J. Paper: Model Predictive Heuristic Control. *Automatica*, 14(5):413–428, Sept. 1978. 4, 6

[44] Takami Sato, Junjie Shen, Ningfei Wang, Yunhan Jia, Xue Lin, and Qi Alfred Chen. Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack. *USENIX Security Symposium*, 2021. 2, 4, 5, 6

[45] Harald Schafer, Eder Santana, Andrew Haden, and Riccardo Biasini. A Commute in Data: The comma2k19 Dataset. *arXiv preprint arXiv:1812.05752*, 2018. 6

[46] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing Properties of Neural Networks. In *ICLR*, 2014. 4

[47] Lucas Tabelini, Rodrigo Berriel, Thiago M. Paix ao, Claudine Badue, Alberto Ferreira De Souza, and Thiago Oliveira-Santos. Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection. In *CVPR*, 2021. 2, 3, 5, 6, 7, 8

[48] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane Estimation via Deep Polynomial Regression. In *ICPR*, 2021. 2, 3, 5, 6, 7, 8

[49] Shiho Tanaka, Kenichi Yamada, Toshio Ito, and Takenao Ohkawa. Vehicle Detection Based on Perspective Transformation Using Rear-View Camera. *Hindawi Publishing Corporation International Journal of Vehicular Technology*, 9, 03 2011. 4

[50] Jigang Tang, Songbin Li, and Peng Liu. A Review of Lane Detection Methods Based on Deep Learning. *Pattern Recognition*, 111:107623, 2021. 1, 3

[51] Daniel Watzenig and Martin Horn. *Automated Driving: Safer and More Efficient Future Driving*. Springer, 2016. 6

[52] Seungwoo Yoo, Hee Seok Lee, Heesoo Myeong, Sungrack Yun, Hyoungwoo Park, Janghoon Cho, and Duck Hoon Kim. End-to-End Lane Marker Detection via Row-Wise Classification. In *CVPR Workshops*, 2020. 3

[53] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *CVPR*, 2020. 2

[54] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. RESA: Recurrent Feature-Shift Aggregator for Lane Detection, 2020. 2

[55] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. RESA: Recurrent Feature-Shift Aggregator for Lane Detection. *AAAI*, 2021. 2