electronics

MDPI

Article

Exploring the Physical-World Adversarial Robustness of Vehicle Detection

Wei Jiang [1,†], Tianyuan Zhang [1,2,*,†], Shuangcheng Liu [2], Weiyu Ji [1], Zichao Zhang [1] and Gang Xiao [3]

[1] Information Science Academy, China Eletronics Technology Group Corporation, Beijing 100846, China
[2] State Key Lab of Software Development Environment, Beihang University, Beijing 100191, China
[3] National Key Laboratory for Complex Systems Simulation, Beijing 100190, China
* Correspondence: zty929601635@gmail.com
† These authors contributed equally to this work.

Abstract: Adversarial attacks can compromise the robustness of real-world detection models. However, evaluating these models under real-world conditions poses challenges due to resource-intensive experiments. Virtual simulations offer an alternative, but the absence of standardized benchmarks hampers progress. Addressing this, we propose an innovative instant-level data generation pipeline using the CARLA simulator. Through this pipeline, we establish the Discrete and Continuous Instant-level (DCI) dataset, enabling comprehensive experiments involving three detection models and three physical adversarial attacks. Our findings highlight diverse model performances under adversarial conditions. YOLO v6 demonstrates remarkable resilience, exhibiting just a marginal 6.59% average drop in average precision (AP). In contrast, the ASA attack yields a substantial 14.51% average AP reduction, twice the effect of other algorithms. We also note that static scenes yield higher recognition AP values, and outcomes remain relatively consistent across varying weather conditions. Intriguingly, our study suggests that advancements in adversarial attack algorithms may be approaching its "limitation". In summary, our work underscores the significance of adversarial attacks in real-world contexts and introduces the DCI dataset as a versatile benchmark. Our findings provide valuable insights for enhancing the robustness of detection models and offer guidance for future research endeavors in the realm of adversarial attacks.

Keywords: adversarial attack; virtual simulation; intelligent perception; autonomous driving

check for updates

Citation: Jiang, W.; Zhang, T.; Liu, S.; Ji, W.; Zhang, Z.; Xiao, G. Exploring the Physical-World Adversarial Robustness of Vehicle Detection. *Electronics* **2023**, *12*, 3921. https://doi.org/10.3390/electronics12183921

Academic Editor: Yannis Papaefstathiou

Received: 15 August 2023
Revised: 7 September 2023
Accepted: 13 September 2023
Published: 18 September 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, advancements in artificial intelligence (AI) technology, epitomized by deep neural networks (DNNs), have realized significant breakthroughs in areas such as computer vision [1–3], natural language processing [4], speech recognition [5], and autonomous driving. These strides have ignited a transformative wave, stimulating growth in societal productivity and catalyzing progress.

Nonetheless, these deep learning methodologies encounter formidable obstacles within the complexities of real-world application scenarios. These include environmental dynamics, input uncertainties, and even potential malevolent attacks, all of which expose vulnerabilities related to security and stability. Research has indicated that deep learning can be significantly influenced by adversarial examples; through the meticulous application of almost imperceptible noise, these models can be misguided into making high-confidence yet inaccurate predictions [6,7]. This emphasizes the inherent unreliability and uncontrollability in the current generation of deep learning models. In recent years, a proliferation of adversarial attack algorithms have been introduced [8–15], underscoring the threats posed by adversarial examples in the digital domain. It is worth noting that, although these adversarial attack methods are initially introduced within specific contexts, their underlying concepts hold a universal applicability that readily extends to other deep learning models. As the exploration of adversarial examples continues, it has become evident that

AI systems deployed in the physical world are also susceptible to these security challenges, potentially leading to catastrophic security incidents. Therefore, research studies on the adversarial security of deep learning in physical-world applications, the robustness testing of models, and ensuring the security and trustworthiness of AI systems have become an urgent imperative.

Unlike the controllable conditions in digital experiments, investigations into adversarial attacks and defenses in the physical world emphasize addressing real-world challenges due to the openness of the experimental scenarios and the variability of environmental conditions. Adversarial examples in the physical world refer to a unique type of samples created by various means such as stickers or paint that alter the features of real objects and can mislead deployed deep learning models post-sampling. For instance, in autonomous driving scenarios, a survey by the Rand Corporation [16] underscores the substantial challenge in establishing robust safety credentials for autonomous vehicles, necessitating the accumulation of an enormous 11 billion miles of test data. This formidable task aligns with the crucial need for empirical validation in the realm of autonomous systems. In the specific context of vehicle recognition and adversarial safety testing, the meticulous control of road conditions is paramount. Additionally, the modification of surface features, such as coatings and stickers, further complicates testing. Moreover, the reliability of sensor-captured data is susceptible to various factors, including weather, lighting, and angles, making this resource-intensive endeavor challenging to accurately replicate, and hampering the detection of security vulnerabilities.

To tackle the challenges outlined above, virtual simulation emerges as an effective solution. By harnessing simulation technology, it can significantly expedite the resource-intensive experiments mentioned earlier, all while providing the valuable advantage of reproducibility. This approach aligns with the broader efforts to enhance the credibility of assessing deep learning models in real-world application scenarios, as evidenced by several studies in adversarial attack and defense-leveraging simulation sandboxes [14,17,18]. By leveraging a physical simulation sandbox powered by a real physics engine, one can model physical scenes, and construct and combine real objects, thereby enabling research into adversarial attack and defense techniques in the physical world. Research grounded in simulation sandboxes can effectively circumvent the challenges of inconvenient testing, high replication difficulty, and excessive testing costs inherent in real-world physical environments. Despite the growing interest in simulation scenarios, a universally accepted benchmark to guide such research is still lacking. Therefore, it is imperative to construct a robust evaluation benchmark for simulation scenarios in the physical world. This leads to three research questions:

- RQ1: How to quickly generate high-fidelity data?
- RQ2: How to build a comprehensive dataset?
- RQ3: How to conduct extensive robustness evaluation?

In order to address this issue, we propose an instant-level scene generation pipeline based on CARLA and introduce the Discrete and Continuous Instant-level (DCI) dataset. This dataset comprises diverse scenarios with varying sequences, perspectives, weather conditions and textures, among other factors. Using the DCI dataset, we conducted extensive experiments to evaluate the effectiveness of our approach. The research framework is depicted in Figure 1. Our primary contributionscan be distilled as follows:

- We present the Discrete and Continuous Instant-level (DCI) dataset, a distinct contribution that sets a benchmark for assessing the robustness of vehicle detection systems under realistic conditions. This dataset facilitates researchers in evaluating the performance of deep learning models against adversarial examples, with a specific emphasis on vehicle detection.
- We perform a thorough evaluation of three detection models and three adversarial attack algorithms utilizing the DCI dataset. Our assessment spans various scenarios, illuminating the efficacy of these attacks under diverse conditions. This comprehensive

evaluation offers insights into the performance of these models and algorithms under a range of adversarial conditions, contributing to the ongoing quest to enhance the robustness and reliability of AI systems against adversarial attacks.

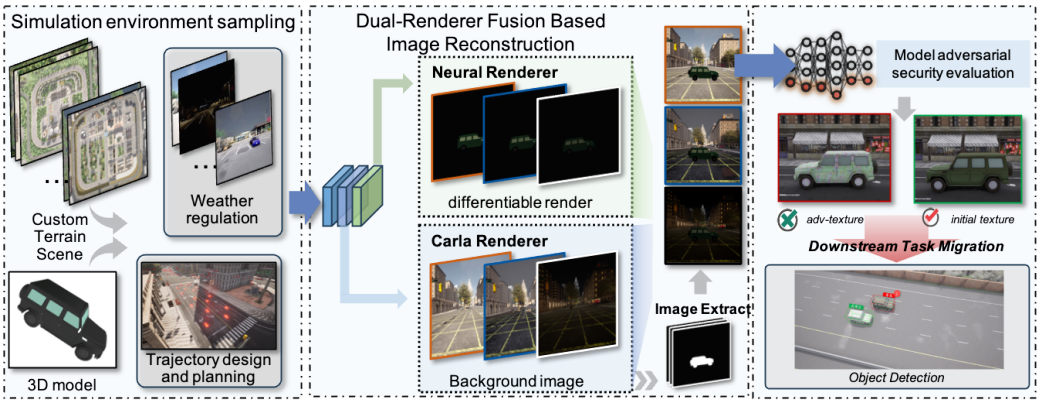


Figure 1. The framework of the entire research. This includes the process of sampling from a virtual environment, rendering with dual renderers, and model testing.

2. Related Work

2.1. Adversarial Attack in the Digital World

Adversarial samples are specially designed samples that are not easily perceived by humans but can lead to erroneous judgments in deep learning models. According to the scope of the attack, adversarial attacks can be divided into two types: digital-world attacks and physical-world attacks.

In the digital world, adversarial attacks directly manipulate image pixels. Szegedy et al. [6] initially proposed adversarial examples, generating them through the L-BFGS method. Capitalizing on target model gradients, Goodfellow et al. [7] introduced the Fast Gradient Sign Method (FGSM) for rapid adversarial example generation. Kurakin et al. [8] enhanced FGSM, developing iterative versions: the Basic Iterative Method (BIM) and the Iterative Least Likely Class Method (ILCM). Madry et al. [19] incorporated a "Clip" function to project and add random perturbations during initialization, culminating in the widely used Projective Gradient Descent (PGD) attack method. Lapid et al. [20] introduced a non-gradient-based adversarial attack method that demonstrated promising outcomes. Liu et al. [21] proposed a methodology for integrating adversarial attacks based on various norms, offering valuable insights for enhancing model robustness.

2.2. Adversarial Attack in the Physical World

Physical adversarial attacks often involve altering an object's visual attributes such as painting, stickers, or occlusion. They are broadly divided into two categories: (1) two-dimensional attacks and (2) three-dimensional attacks.

Two-dimensional attacks are typically executed via the application of distinct patterns or stickers to the targeted objects. Sharif et al. [22] deceived facial recognition systems by creating wearable eyeglass frames that mislead models in the physical world. Brown et al. [23] designed "adversarial patches", which are small perturbed areas that can be printed and pasted to effectively conduct an attack. Eykholt et al. [24] devised the Robust Physical Perturbation (RP2) method, which misguides traffic sign classifiers using occlusion textures. Thys et al. [25] demonstrated an attack on human detection models by attaching a two-dimensional adversarial patch to the human torso. Sato et al. [26] proposed the Dirty Road attack, misleading autonomous vehicles' perception modules by painting camouflages on lanes. Liu et al. [27] proposed X-adv, which implemented adversarial attacks against X-ray security inspection systems. Deng et al. [28] developed an adversarial patch generation framework inspired by rust style and utilizing style transfer techniques to

target detection models. Sun et al. [29] proposed ways to combat patch attacks in the field of optical remote sensing images (O-RSIs).

While two-dimensional physical attacks have proven to be effective, they are constrained by sampling angles and other conditions; thus, their success in general scenarios is not guaranteed. Three-dimensional physical attacks provide a solution. Athalye et al. [30] introduced the Expectation Over Transformation (EOT) framework to create adversarial attacks on 2D images and 3D objects. In contrast, Maesumi et al. [31] proposed a 3D-to-2D adversarial attack method, using structured patches from a reference mannequin, with adaptable human postures during training.

Moreover, three-dimensional attacks are also executed using simulation environments. Zhang et al. [14] and Wu et al. [17] utilized open-source virtual simulation environments for their optimized adversarial attacks. Wang et al. [32] introduced Dual Attention Suppression (DAS) to manipulate attention patterns in models. Zhang et al. [33] developed the Attention on Separable Attention (ASA) attack, enhancing the effectiveness of adversarial attacks.

Given the emergence of numerous adversarial attack and defense studies, the establishment of a benchmark for a comprehensive security analysis of these algorithms becomes imperative.

2.3. Adversarial Robustness Benchmark

Several physical-world adversarial example generation methods have been proposed and demonstrated to be effective [10,13,34–39]. However, they use different datasets for evaluation, which makes it difficult to conduct a comprehensive evaluation. In light of this challenge, benchmarks have emerged as crucial instruments for addressing these issues. Benchmarks, such as those introduced by Dong [40] and Liu [41], have been proposed to provide standardized evaluation frameworks. Tang [42] proposed the first unified Robustness Assessment Benchmark, RobustART, which provides a standardized evaluation framework for adversarial examples. Recently, there has been a notable emergence of a series of benchmarks [39,43–48] in different fields. These benchmarks serve the crucial function of unifying evaluation metrics, enabling a clear and systematic comparison of the strengths and weaknesses inherent in various methods.

In the virtual simulation environment, several adversarial attack algorithms for vehicle recognition scenarios have been proposed [14,32,49,50] and shown to be effective. The CARLA simulator [51] has been widely used in these studies due to its versatility and availability. However, the lack of a unified evaluation benchmark makes it difficult to compare and analyze the results. Establishing a benchmark is essential to promote the development of robust vehicle detection models.

2.4. Virtual Environment of Vehicle Detection

A series of vehicle detection-related simulators have been proposed. Simulators developed based on the Unity engine, such as LGSVL [52], and those developed based on the Unreal engine, such as Airsim [53] and CARLA [51], all support camera simulation. Among them, the Airsim simulator focuses more on drone-related research, while compared with LGSVL, current research on adversarial security is more focused on the CARLA simulator [32,49,50]. CARLA is equipped with scenes and high-precision maps made by RoadRunner and provides options for map editing. It also supports environment lighting and weather adjustments, as well as the simulation of pedestrian and vehicle behaviors.

Based on the above exploration, this study intends to use the CARLA autonomous driving simulator as the basic simulation environment to carry out research on the security analysis of autonomous driving intelligent perception algorithms.

3. DCI Dataset: Instant-Level Scene Generation and Design

The goal of this research is to construct a physical-world robustness evaluation benchmark. To achieve this goal, we propose the dual-renderer fusion-based image reconstruction method that integrates the advantages of the high fidelity of traditional image renderers

and easy optimization of neural renderers. Based on this image generation scheme, we refer to common application scenarios for vehicle detection in the physical world and design a DCI dataset from the perspectives of breadth and depth. This lays the foundation for subsequent physical-world robustness assessments, enabling researchers to evaluate the effectiveness of vehicle detection models in the physical world.

3.1. Neural 3D Mesh Render Technology

Neural 3D Mesh Renderer [33] is an image rendering technique based on deep learning that leverages trained neural networks to produce high-quality images. Traditional image rendering approaches typically require the manual definition of intricate rendering rules and optical models, using rasterization and shading techniques to generate realistic images. In contrast, neural rendering methods simplify this process by employing deep neural networks to automatically learn these rules and models. Additionally, these techniques ensure the traceability of the gradient of the training textures during the rendering process, facilitating the training and evaluation of adversarial attack and defense samples.

3.2. Dual-Renderer Fusion-Based Image Reconstruction

In this study, we have developed an instance-level scene generation approach that effectively combines the CARLA simulator and the neural renderer. This scene generation approach possesses several notable advantages: it exhibits rapid data generation, maintains a high level of fidelity, and facilitates straightforward optimization. The CARLA simulator, an image-based rendering tool, provides high fidelity and precision in detail, but is hindered by its inability to differentiate textures generated during rendering, which hampers adversarial sample generation and gradient-based optimization. In contrast, neural renderers bypass this limitation, preserving gradient traceability during the rendering process, which is essential for creating adversarial samples and enhancing adversarial attacks. This fusion of rendering tools not only facilitates the production of highly realistic scene imagery, but also supports the generation and optimization of adversarial attack methods by utilizing gradient information from the rendering process.

In previous studies, the only parameter that passed between the two renderers was the positional coordinate P_{co}. The positional coordinate P_{co} comprises several key elements, including the angle of the model, the elevation angle of the observation point, the azimuth angle of the observation point, and the size of the model. These parameters are critical in ensuring that the rendered 3D model aligns with expectations in terms of its angle, size, and shape, and that the textures on the model are displayed correctly.

While this method ensures consistency in the model's appearance pre- and post-rendering, it disregards the influence of environmental factors such as lighting changes on the final render quality. Consequently, in the synthesized image, the rendered 3D model appears with the same lighting effect under different environmental conditions (like sunny, rainy, night, etc.), which significantly impairs the image's realism. To address this issue, we introduced environmental parameters P_{en} as additional transfer parameters to minimize the difference in lighting between the two renderers, thus enhancing the realism of the render. By leveraging environmental parameters P_{en}, we can ensure that the rendered 3D model interacts with objects in the simulated environment, rather than existing as an independent entity. Specifically, environmental parameters capture information such as the lighting angle, intensity, and color of each capture point within Carla, which are essential in simulating similar lighting conditions using a neural renderer. With this information, we can create an interactive 3D model that is seamlessly integrated into the simulated environment. By integrating the CARLA simulator and the neural renderer in this manner, we have successfully built an instance-level scene generation method that guarantees scene realism while also supporting the training of adversarial textures. The parameters passed are shown in Table 1.

Table 1. Illustration of parameter transfer between renderers. The parameters fall into two categories: (1) position coordinates and (2) environmental parameters. Each category plays a vital role in enhancing the realism and fidelity of the rendered images.

Categories	Parameter Name
Positional coordinate P_{co}	$Model_{angle}$
	$Camera_{up}$
	$Camera_{direction}$
Environmental parameters P_{en}	$Intensity_{direction}$
	$Intensity_{directional}$
	$Color_{ambient}$
	$Color_{directional}$
	$Direction$

Specifically, we use the CARLA simulator to first generate the **background** image and obtain the position coordinates P_{co} and environment parameters P_{en} using the simulator's built-in sensor. Next, we transfer P_{co} and P_{en} to the neural renderer. The neural renderer then loads the 3D model and uses the received parameters to generate the ***Car*** image. During the rendering process, we adjust the relevant settings of the neural renderer according to the sampling environment in CARLA to narrow the gap between the two renderers. We then use a ***Mask*** to extract the background image and vehicle image, respectively. After completing the pipeline, we obtain an instant-level scene. The framework of scene generation is shown in Figure 2.

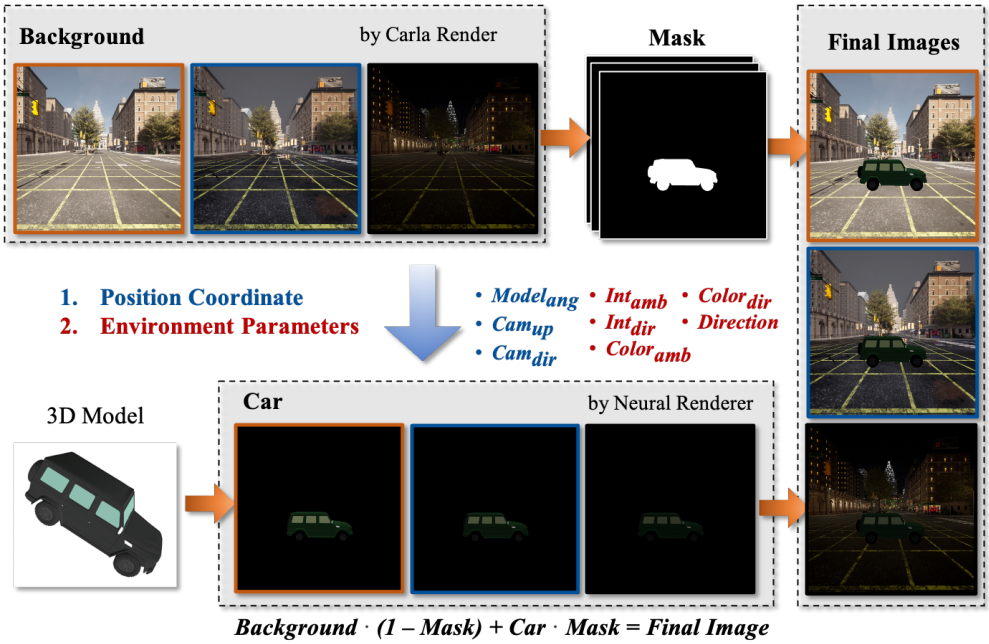


Figure 2. The pipeline of dual-renderer fusion-based image reconstruction.

By introducing environmental parameters P_{en}, we have successfully enhanced the overall quality of the scene generation process and maintained consistency between the renderers under varying lighting conditions. This approach lays a significant foundation for analyzing the robustness and security of deep learning models under various environmental conditions. We believe that this instance-level scene generation method can provide more comprehensive and reliable support for the adversarial safety testing and evaluation of autonomous driving systems.

3.3. Connected Graphs-Based Case Construction

When constructing scenario execution cases, understanding certain fundamental concepts is crucial. In the CARLA simulator, Actors refer to objects that can be arbitrarily positioned, set to follow motion trajectories, and perform actions. These include vehicles, pedestrians, traffic signs, traffic lights, sensors, and more. These Actors play various roles in the simulation scenario, and their interactions significantly influence the overall simulation process. Notably, CARLA's sensors, such as RGB cameras and instance segmentation cameras, can be attached to other Actors for data collection. By strategically positioning the Actor and determining its action trajectory, a broad range of scenarios and execution instances can be generated for testing various algorithms and models.

The original approach to generating and setting Actors in the CARLA simulator involves manually determining each Actor's position, speed, displacement distance, and steering angle. However, this method has issues such as slow generation speed and a lack of realistic simulation effects, thereby necessitating improvements. As a solution, we utilized an optimization method based on Connected Graph generation. This method automatically generates information such as the location, number, and action track of Actors via a program, enabling the quick construction of numerous execution instances. Specifically, the CARLA built-in map contains several "spawn points". By setting these spawn points on the running path and incorporating the A* shortest path generation algorithm, we can quickly generate and realistically simulate Actor trajectories. The algorithm's pseudocode is presented in Algorithm 1.

Algorithm 1 Connected Graphs-based Case Construction

/* Initialization */
Initialize empty set $OpenList$ and add $startNode$ to it
while $OpenList \neq \varnothing$ **do**
    $currentNode \leftarrow \arg\min_{node \in OpenList} f(node)$
    $childSet \leftarrow$ {Children of $currentNode$ that are valid and not visited}
    for each $childNode$ in $childSet$ **do**
        Calculate $f(childNode)$ considering $currentNode$ as parent
        if $f(childNode)$ can be improved **then**
            Update parent of $childNode$ to $currentNode$
        end if
    end for
    Remove $currentNode$ from $OpenList$
    if $endNode \in childSet$ **then**
        break
    end if
end while
$shortestPath \leftarrow$ Trace back from $endNode$ to $startNode$
return $shortestPath$

3.4. Composition of the DCI Dataset

Based on the data generation scheme mentioned earlier, we designed the Discrete and Continuous Instant-level (DCI) dataset to evaluate the performance of vehicle detection models in diverse scenarios. It can be divided into two parts that focus on different aspects. Figure 3 illustrates various components of the DCI dataset. The process of scenario selection pertains to identifying real-world conditions that have a substantial likelihood of occurrence and are susceptible to security-related concerns [54].

The **continuous** part of the DCI dataset comprises seven typical scenes, each describing a real-life scenario that is widely used. To address the issue of irregular data distribution, we employed a fixed-viewpoint approach to address issues of uneven data distribution and insufficient scene representation. This approach includes the driver's perspective and monitor view. The driver's viewpoint simulates the field of view of an on-road driver,

while the drone viewpoint offers a comprehensive bird's-eye view of the scene. Lastly, the surveillance viewpoint resembles that of a fixed surveillance camera. This multi-viewpoint strategy broadens our data collection scope, significantly enhancing the dataset's quality and diversity. To expand the coverage, we chose three different weather conditions to generate the dataset: *ClearNoon*, *ClearNight*, and *WetCloudySunset*. This part of the dataset involves seven angles, distances, and more than 2000 different positions. The instance-based scene generator images are shown in Figure 4.

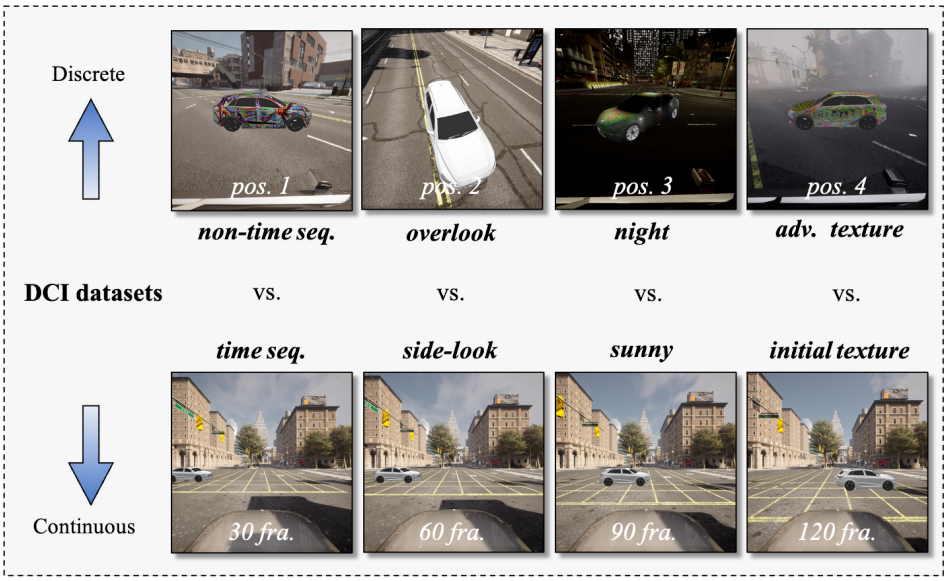


Figure 3. The Discrete and Continuous Instant-level Dataset (DCI): the discrete part aims to provide all-around coverage, while the continuous part is designed to test specific scenarios in greater depth.

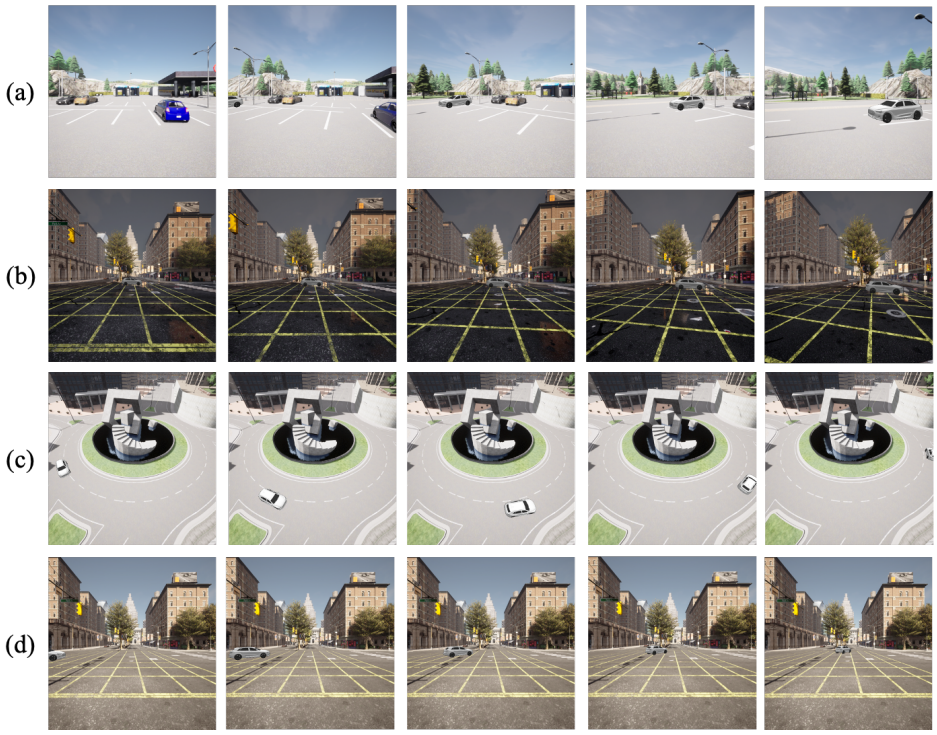


Figure 4. Instance-based scene generator images under different scenes: (**a**) Parking Lot. (**b**) Turning A. (**c**) Traffic Circle. (**d**) Straight A.

The **discrete** part of the DCI dataset aims to extend coverage by widely selecting parameters such as map locations, sampling distances, pitch angles, and azimuth angles,

encompassing various road types and topological structures. We traverse road locations on the map while fine-tuning lighting angles and intensities to simulate variations in illumination under different times and weather conditions. Moreover, we adjust environmental conditions like haze and particle density, thereby enhancing the dataset's authenticity and diversity. This segment includes 40 angles, 15 distances, and over 20,000 distinct locations. The composition of the DCI dataset is shown in Table 2.

Table 2. Overview of the DCI dataset.

Scene Name	Scene Description (Perspective)
Overall [1]	Random (Random)
Traffic Circle	Driving in the center of the road (Monitor)
Parking Lot	Exiting from a parking lot (Driver)
Stationary A	Stationary observation (Driver)
Straight A	Driving straight on a road (Driver)
Turning A	Turning at an intersection (Driver)
Stationary B	Stationary observation (Driver)
Straight B	Driving straight on a road (Driver)

[1] This is the discrete part, while the others are all continuous parts.

4. Experiments and Evaluations

4.1. Experiment Settings

Adversarial Attack Algorithm. Following Zhang et al. [49], we intentionally selected strategies grounded in divergent conceptual frameworks. Specifically, our choices encompassed methods that leverage the model's attention mechanism, exemplified by DAS [32] and ASA [49], as well as those predicated on the model's intrinsic loss functions, as epitomized by FCA [50]. These are the commonly adopted adversarial attack methods in the physical world. These algorithms were carefully chosen based on their proven effectiveness in generating adversarial examples and their compatibility with our proposed method. The adversarial texture was trained on the discrete dataset mentioned earlier, utilizing 1 epoch, a batch size of 1, and an iteration step size of 1×10^{-5}.

Vehicle 3D Model. Building upon the approach established by Wang et al. [32], we employed the Audi E-Tron, a frequently utilized 3D model in prior research, for our experimental investigations. The model comprises 13,449 vertices, 10,283 vertex normals, 14,039 texture coordinates, and 23,145 triangles.

Vehicle Detection Algorithm. Target detection models are typically categorized into two primary types: single-stage and two-stage models. Additionally, it is important to note that even within the same type of detection model, there may exist various architectures. In pursuit of enhanced coverage, we evaluated the proposed method on three popular object detection algorithms: YOLO v3 [2]; YOLO v6 [55]; and Faster R-CNN [56]. By selecting both single-stage and two-stage typical algorithms, we investigated the capability of the attack algorithm in the real world. The target class we chose is the car. We used the average precision (AP) as the evaluation metric to measure the performance of the detection algorithm on the test dataset.

4.2. Analysis of Experimental Results in Discrete Part

We initially selected the overall coverage scenario for analysis, which allows for a comprehensive performance assessment under various conditions. We use the "@" symbol to represent the corresponding model, as shown in Table 3. Under original texture conditions, the YOLO v6 model exhibited the highest AP value, reaching 73.39%. The YOLO v6 model was closely followed by the YOLO v3 model, with an AP of 65.37%. Meanwhile, the Faster RCNN model had the lowest AP value, at just 56.81%.

Table 3. Accuracy of vehicle detection (AP) in discrete scenarios. Bold text displays the highest value in each row.

Texture Type	AP@YOLO v3 (%)	AP@YOLO v6 (%)	AP@FRCNN (%)
initial texture	65.37	**73.39**	56.81
ASA adv-texture	41.59	**65.68**	44.76
DAS adv-texture	57.39	**65.66**	50.49
FCA adv-texture	456.8	**68.44**	47.21

Thus, under conditions free from adversarial attacks, the order of **detection accuracy rates** is as follows: **YOLO v6 > YOLO v3 > Faster RCNN**.

Switching vehicle textures to adversarial forms resulted in notable shifts in model detection accuracy. Under ASA adversarial texture, the performance of the YOLO v3 model was notably diminished, registering an AP of 41.59%, less than the Faster RCNN model's 44.76% AP. This anomaly may stem from the significant impact of the ASA adversarial texture on the YOLO v3 model.

Contrastingly, in DAS and FCA adversarial scenarios, the YOLO v3 model outperformed Faster RCNN, recording APs of 57.39% and 56.8%, compared to Faster RCNN's 50.76% and 47.21%, respectively. This highlights YOLO v3's relative resilience and stability under these adversarial conditions.

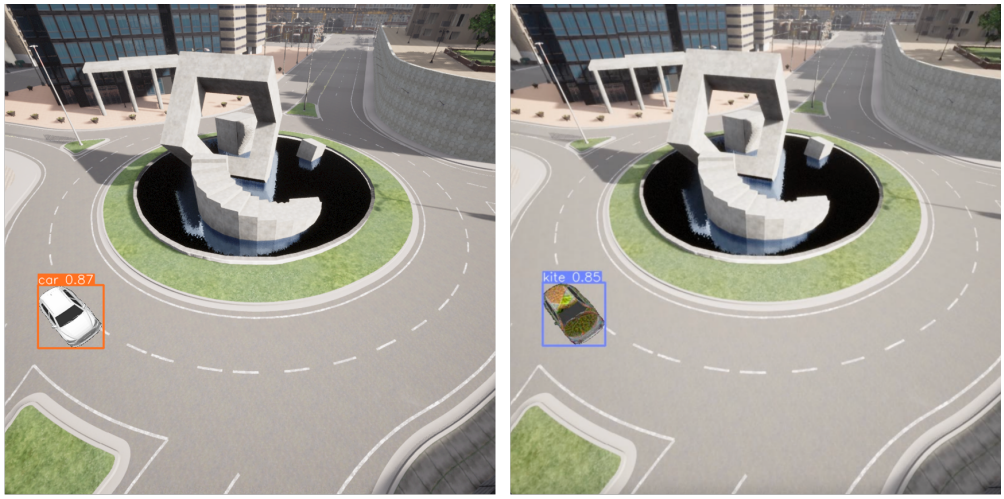
To assess the effectiveness of adversarial texture attacks, merely observing the average precision (AP) values can be insufficient as these can be influenced by a myriad of factors. To more precisely evaluate the attack effects, we considered the decline in AP. Thus, we calculated the average AP drop rates under adversarial texture conditions for various object detection models, as presented in Table 4.

Table 4. Accuracy decline in vehicle detection (AP) in discrete scenarios. Bold text displays the highest value in each row.

Texture Type	AP@YOLO v3 (%)	AP@YOLO v6 (%)	AP@FRCNN (%)
ASA adv-texture	**23.78**	7.71	12.05
DAS adv-texture	**7.98**	7.74	6.32
FCA adv-texture	8.57	4.95	**9.6**

Following the implementation of adversarial perturbations, the mean decrease in AP was 13.44%, 6.79%, and 9.32% for YOLO v3, YOLO v6, and Faster RCNN models, respectively. Notably, the YOLO v6 model demonstrates the highest resilience with the least AP decrease, while the YOLO v6 model is the weakest, with the most significant drop. The Faster RCNN model presents good robustness but is slightly behind the YOLO v6 model. To visually demonstrate attack effects, we utilized the YOLO v3 model on a frame rendered with original and FCA adversarial textures (Figure 5).

In summary, when subjected to adversarial texture attacks, the YOLO v6 model exhibits superior robustness, while YOLO v3 presents less resilience. The **robustness ranking** is as follows: **YOLO v6 > Faster RCNN > YOLO v3**. Hence, in view of practical deployment scenarios, it becomes imperative to opt for models characterized by both high accuracy and robustness. Among the models scrutinized in our experiments, YOLO V6 demonstrates superior performance in this regard.



(a)initial texture: 87% car (b)FCA adv-texture: 85% kite

Figure 5. Illustration of YOLO v3 model's performance under original and FCA adversarial textures. The model correctly identifies a car with 87% confidence under original texture but misclassifies it as a kite with 85% confidence under the FCA adversarial texture.

4.3. Analysis of Experimental Results in Continuous Part

Upon analysis of the overall coverage scenario, we further delve into various subdivided scenarios to examine the models' recognition performance in diverse environments, with specific results presented in Table 5. Through these more granulated scenario experiments, we have observed some differing results.

Primarily, the YOLO v6 model still exhibits the highest recognition accuracy across most scenarios. This indicates that the YOLO v6 model has superior performance and can maintain high accuracy across a multitude of subdivided scenarios.

Table 5. Accuracy of vehicle detection (AP) in continuous scenarios. Bold text displays the highest value in each row.

Scene	AP@YOLO v3 (%)	AP@YOLO v6 (%)	AP@FRCNN (%)
Traffic Circle	63.36	**86.95**	19.43
Parking Lot	23.03	32.33	**33.64**
Stationary A	66.27	**68.99**	67.89
Straight Through A	80.93	**82.49**	81.86
Turning Left A	48.02	**64.34**	18.26
Stationary B	98.81	**100**	100
Straight Through B	76.72	**78.68**	76.57

Interestingly, within the overall poorer-performing Faster RCNN model, we found that it achieved the highest accuracy among the three models in the "Parking Lot" and "Stationary B" scenarios. This implies that while the Faster RCNN model can perform well in specific scenarios, it tends to be unstable in others.

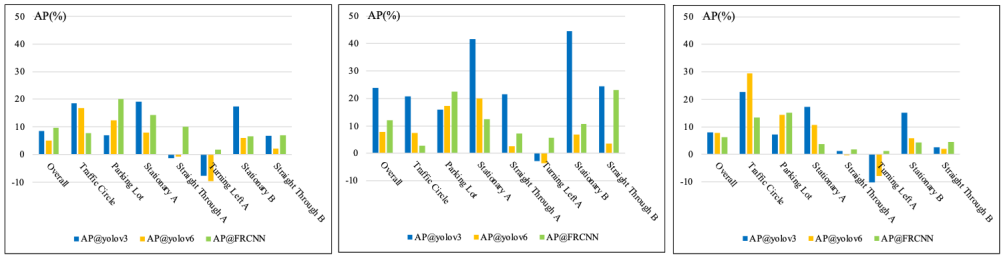
In our scene-specific tests, we evaluated the models' AP decline under adversarial attacks, revealing the variance in algorithm performance across diverse scenes (Table 6). The YOLO v3 model demonstrated the most significant AP decline, often exceeding 20%. Conversely, YOLO v6 and Faster RCNN showed a more stable AP decline, consistently under 20%. This implies that model robustness varies across scenes, with YOLO v3 particularly needing additional optimization for complex environments, while YOLO v6 and Faster RCNN display superior robustness.

Table 6. Accuracy decline in vehicle detection (AP) in continuous scenarios. Bold text displays the highest value in each row.

Scene	AP@YOLO v3 (%)	AP@YOLO v6 (%)	AP@FRCNN (%)
Traffic Circle	**13.44**	6.80	9.32
Parking Lot	**20.66**	17.89	7.95
Stationary A	10.02	14.70	**19.23**
Straight Through A	**25.98**	12.82	10.19
Turning Left A	**7.15**	0.46	6.36
Stationary B	−7.41	−7.01	**2.86**
Straight Through B	**25.72**	6.21	7.12

Upon a detailed examination of various adversarial attack algorithms, as illustrated in Figure 6, we observe a pronounced drop in the AP for the YOLO v3 model under the ASA attack, underlining its weakest robustness against this specific adversarial scenario. Interestingly, the AP decrease across other adversarial attack methods does not show significant discrepancies for the remaining models. This observation suggests that the impact of different adversarial attacks on target detection models varies significantly. In particular, the YOLO v3 model exhibits a substantial decrease in performance under ASA, resulting in a considerable drop in AP. However, in other adversarial scenarios, all three models showcase comparable levels of robustness, indicating a relatively strong resistance to adversarial texture attacks.

Thus, under adversarial attacks in specific scenarios, the robustness of the three-object detection algorithms is ranked as follows: **YOLO v6 > Faster RCNN > YOLO v3**, which aligns with the results observed in overall scenarios. Taking practical application into account, the observed performance of models across various scenarios underscores the limitations of relying on a single model to address the demands of all complex scenarios. Instead, it becomes evident that employing models with distinct characteristics, tailored to specific scenarios, is a more effective strategy.



(a) FCA (b) ASA (c) DAS

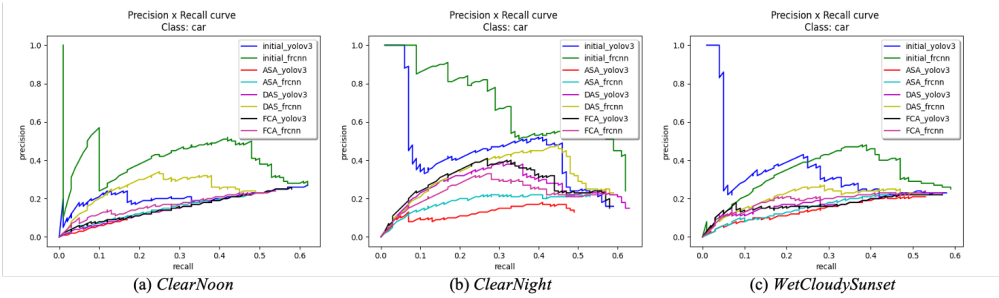
Figure 6. Average decrease in AP for detection models under attack in various scenarios.

4.4. Analysis of Specific Scenarios

We decided to analyze the Parking Lot scenario further. Figure 7 presents the Precision–Recall curves corresponding to different adversarial textures under the YOLO v3 and Faster RCNN models. As expected, the two lines with the highest values correspond to the initial textures. Interestingly, in the context of adversarial texture conditions within the same scene, it is noteworthy that while there exist numerical distinctions in the data distribution of PR images, these variations manifest similar trends and patterns.

Considering that the attack magnitude was unrestricted, this implies that there may be a common "limiting factor" among different attacks, rendering their effects similar to a certain extent. It is conceivable that there exists a lower threshold for the Precision–Recall (PR) graph of the same detection model when subjected to diverse adversarial attacks. Hence, in the pursuit of novel adversarial attack methods, a fruitful approach might entail commencing from the vantage point of data distribution and seeking more efficacious attack techniques by progressively approaching the discernible "limit". This finding has

significant implications for understanding the nature of adversarial attacks and their impact in practical applications, and could guide future research direction in the realm of object detection and adversarial attacks.



(a) *ClearNoon* (b) *ClearNight* (c) *WetCloudySunset*

Figure 7. The Precision–Recall chart illustrates the Parking Lot scenario in three different weather conditions, demonstrating a similar distribution of values.

5. Discussion

As previously introduced, our study addresses three fundamental research questions pertinent to the evaluation of real-world robustness. In this chapter, we offer concise and well-defined responses to these initial research inquiries through the presentation of our proposed methodologies and the outcomes of our experimental investigations.

RQ1: The Dual-Renderer Fusion-Based Approach. To tackle the challenge of data generation, we implemented an innovative image generation methodology rooted in dual-renderer fusion. This approach acts as a crucial bridge between the CARLA simulator and neural simulation, enabling the rapid, highly realistic, and efficiently optimized generation of data. By leveraging the synergy between these two rendering methods, our approach significantly enhances the authenticity and efficiency of data generation, a critical aspect of assessing model robustness.

RQ2: DCI Dataset Generation. To mitigate concerns related to data coverage, we formulated a methodology rooted in the generation of real-world scenario data. This approach significantly enhances the scope and granularity of scenario representation within our dataset, encompassing a diverse spectrum of everyday situations, both discrete and continuous. By encompassing such a wide array of scenarios, our DCI dataset not only facilitates the evaluation of model performance, but also serves as a valuable asset for training and benchmarking physical-world applications.

RQ3: Exploring the Robustness. To ensure a comprehensive evaluation, we adopted a multifaceted approach. This involved the utilization of detection models characterized by distinct architectural features and the implementation of diverse adversarial attack techniques. By employing a variety of detection models and attack strategies, our study provides a nuanced and in-depth analysis of model robustness. This multifaceted approach enhances the effectiveness of our testing procedures, ultimately contributing to a more thorough and rigorous evaluation framework.

In conclusion, our study significantly advances the understanding of physical-world model robustness evaluation by addressing these three fundamental research questions. Our methodologies, datasets, and experimental findings collectively contribute to the ongoing development of robustness evaluation in the context of physical-world applications.

6. Conclusions

Our study contributes to the realm of benchmarks for evaluating the robustness of physical-world systems. Primarily, we introduce a novel dual-renderer fusion-based image reconstruction approach that synergizes the merits of conventional image renderers and neural renderers. This innovative method not only ensures superior fidelity, but also facilitates streamlined optimization processes. Additionally, we present the Discrete and Continuous Instant-level (DCI) dataset, meticulously crafted to encompass a diverse array

of scenarios characterized by dynamic sequences, varied perspectives, diverse weather conditions, and intricate textures. This comprehensive dataset offers unparalleled breadth and depth, forming a solid foundation for the comprehensive assessment of vehicle detection models under authentic conditions.

Our experimental endeavors yield noteworthy insights. In the experiment, YOLO v6 showed the strongest resistance to attacks with an average AP drop of only 6.59%. ASA was the most effective attack algorithm, reducing the average AP by 14.51%, which is twice that of other algorithms. Static scenes had a higher recognition AP, and the results in the same scene under different weather conditions were similar. Further improvement of adversarial attack algorithms may be approaching the "limitation".

Nevertheless, it is imperative to acknowledge the limitations inherent in our study. The DCI dataset, while relatively comprehensive, may not provide a wholly exhaustive representation of the intricacies characterizing real-world scenarios. Furthermore, it is worth noting that the dataset might not encompass scenarios involving extreme environmental conditions, thereby suggesting potential avenues for refinement and augmentation in future research endeavors.

In the trajectory of future research, we propose delving into more sophisticated adversarial attack algorithms while also gauging the efficacy of alternative defense mechanisms. Furthermore, we advocate for an in-depth exploration of the impact of environmental factors, ranging from fluctuating lighting conditions and diverse weather phenomena to intricate traffic patterns, on the performance of vehicle detection models. By embracing these undertakings, we envision a progressive evolution in the realm of benchmarks for evaluating real-world robustness, thereby augmenting the performance of vehicle detection models within authentic scenarios.

Author Contributions: Conceptualization, W.J. (Wei Jiang) and T.Z.; methodology, W.J. (Wei Jiang) and T.Z.; validation, W.J. (Weiyu Ji) and Z.Z.; data curation, S.L. and W.J. (Weiyu Ji); writing—original draft preparation, T.Z. and S.L.; writing—review and editing, W.J. (Wei Jiang) and G.X.; visualization, Z.Z. and G.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The data are not publicly available due to future research plans.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *60*, 84–90. [CrossRef]
2. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
3. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
4. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
5. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
6. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
7. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
8. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018; pp. 99–112.
9. Evtimov, I.; Eykholt, K.; Fernandes, E.; Kohno, T.; Li, B.; Prakash, A.; Rahmati, A.; Song, D. Robust physical-world attacks on machine learning models. *arXiv* **2017**, arXiv:1707.08945.
10. Liu, A.; Wang, J.; Liu, X.; Cao, B.; Zhang, C.; Yu, H. Bias-based universal adversarial patch attack for automatic check-out. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XIII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 395–410.
11. Wei, X.S.; Cui, Q.; Yang, L.; Wang, P.; Liu, L. RPC: A large-scale retail product checkout dataset. *arXiv* **2019**, arXiv:1901.07249.

12. Duan, R.; Ma, X.; Wang, Y.; Bailey, J.; Qin, A.K.; Yang, Y. Adversarial camouflage: Hiding physical-world attacks with natural styles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1000–1008.
13. Liu, A.; Huang, T.; Liu, X.; Xu, Y.; Ma, Y.; Chen, X.; Maybank, S.J.; Tao, D. Spatiotemporal attacks for embodied agents. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XVII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 122–138.
14. Zhang, Y.; Foroosh, H.; David, P.; Gong, B. CAMOU: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
15. Huang, L.; Gao, C.; Zhou, Y.; Xie, C.; Yuille, A.L.; Zou, C.; Liu, N. Universal physical camouflage attacks on object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 720–729.
16. Kalra, N.; Paddock, S.M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A Policy Pract.* **2016**, *94*, 182–193. [CrossRef]
17. Wu, T.; Ning, X.; Li, W.; Huang, R.; Yang, H.; Wang, Y. Physical adversarial attack on vehicle detector in the carla simulator. *arXiv* **2020**, arXiv:2007.16118.
18. Xiao, C.; Yang, D.; Li, B.; Deng, J.; Liu, M. Meshadv: Adversarial meshes for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6898–6907.
19. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
20. Lapid, R.; Sipper, M. Patch of invisibility: Naturalistic black-box adversarial attacks on object detectors. *arXiv* **2023**, arXiv:2303.04238.
21. Liu, A.; Tang, S.; Liu, X.; Chen, X.; Huang, L.; Qin, H.; Song, D.; Tao, D. Towards Defending Multiple ℓ_p-Norm Bounded Adversarial Perturbations via Gated Batch Normalization. *Int. J. Comput. Vis.* **2023**, 1–18.
22. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proceedings of the 2016 ACM Sigsac Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1528–1540.
23. Brown, T.B.; Mané, D.; Roy, A.; Abadi, M.; Gilmer, J. Adversarial patch. *arXiv* **2017**, arXiv:1712.09665.
24. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; Song, D. Robust physical-world attacks on deep learning visual classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1625–1634.
25. Thys, S.; Van Ranst, W.; Goedemé, T. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
26. Sato, T.; Shen, J.; Wang, N.; Jia, Y.; Lin, X.; Chen, Q.A. Dirty road can attack: Security of deep learning based automated lane centering under Physical-World attack. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Vancouver, BC, Canada, 11–13 August 2021; pp. 3309–3326.
27. Liu, A.; Guo, J.; Wang, J.; Liang, S.; Tao, R.; Zhou, W.; Liu, C.; Liu, X.; Tao, D. X-adv: Physical adversarial object attacks against X-ray prohibited item detection. *arXiv* **2023**, arXiv:2302.09491.
28. Deng, B.; Zhang, D.; Dong, F.; Zhang, J.; Shafiq, M.; Gu, Z. Rust-Style Patch: A Physical and Naturalistic Camouflage Attacks on Object Detector for Remote Sensing Images. *Remote Sens.* **2023**, *15*, 885. [CrossRef]
29. Sun, X.; Cheng, G.; Pei, L.; Li, H.; Han, J. Threatening patch attacks on object detection in optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2023**. [CrossRef]
30. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing robust adversarial examples. In Proceedings of the International Conference on Machine Learning. PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 284–293.
31. Maesumi, A.; Zhu, M.; Wang, Y.; Chen, T.; Wang, Z.; Bajaj, C. Learning transferable 3D adversarial cloaks for deep trained detectors. *arXiv* **2021**, arXiv:2104.11101.
32. Wang, J.; Liu, A.; Yin, Z.; Liu, S.; Tang, S.; Liu, X. Dual attention suppression attack: Generate adversarial camouflage in physical world. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8565–8574.
33. Kato, H.; Ushiku, Y.; Harada, T. Neural 3d mesh renderer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3907–3916.
34. Liu, A.; Liu, X.; Fan, J.; Ma, Y.; Zhang, A.; Xie, H.; Tao, D. Perceptual-sensitive gan for generating adversarial patches. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 1028–1035.
35. Wang, J.; Liu, A.; Bai, X.; Liu, X. Universal adversarial patch attack for automatic checkout using perceptual and attentional bias. *IEEE Trans. Image Process.* **2021**, *31*, 598–611. [CrossRef] [PubMed]
36. Wang, J.; Yin, Z.; Hu, P.; Liu, A.; Tao, R.; Qin, H.; Liu, X.; Tao, D. Defensive patches for robust recognition in the physical world. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2456–2465.

37. Liu, S.; Wang, J.; Liu, A.; Li, Y.; Gao, Y.; Liu, X.; Tao, D. Harnessing perceptual adversarial patches for crowd counting. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, CA, USA, 7–11 November 2022; pp. 2055–2069.
38. Liu, A.; Tang, S.; Liang, S.; Gong, R.; Wu, B.; Liu, X.; Tao, D. Exploring the Relationship between Architecture and Adversarially Robust Generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.
39. Guo, J.; Bao, W.; Wang, J.; Ma, Y.; Gao, X.; Xiao, G.; Liu, A.; Dong, J.; Liu, X.; Wu, W. A Comprehensive Evaluation Framework for Deep Model Robustness. *Pattern Recognit.* **2023**, *137*, 109308. [CrossRef]
40. Dong, Y.; Fu, Q.A.; Yang, X.; Pang, T.; Su, H.; Xiao, Z.; Zhu, J. Benchmarking adversarial robustness on image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 321–331.
41. Liu, A.; Liu, X.; Yu, H.; Zhang, C.; Liu, Q.; Tao, D. Training robust deep neural networks via adversarial noise propagation. *IEEE Trans. Image Process.* **2021**, *30*, 5769–5781. [CrossRef] [PubMed]
42. Tang, S.; Gong, R.; Wang, Y.; Liu, A.; Wang, J.; Chen, X.; Yu, F.; Liu, X.; Song, D.; Yuille, A.; et al. Robustart: Benchmarking robustness on architecture design and training techniques. *arXiv* **2021**, arXiv:2109.05211.
43. Zhang, T.; Xiao, Y.; Zhang, X.; Li, H.; Wang, L. Benchmarking the Physical-world Adversarial Robustness of Vehicle Detection. *arXiv* **2023**, arXiv:2304.05098.
44. Yu, K.; Tao, T.; Xie, H.; Lin, Z.; Liang, T.; Wang, B.; Chen, P.; Hao, D.; Wang, Y.; Liang, X. Benchmarking the robustness of lidar-camera fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 3187–3197.
45. Ali, S.; Sahoo, B.; Zelikovsky, A.; Chen, P.Y.; Patterson, M. Benchmarking machine learning robustness in COVID-19 genome sequence classification. *Sci. Rep.* **2023**, *13*, 4154. [CrossRef]
46. Li, S.; Zhang, S.; Chen, G.; Wang, D.; Feng, P.; Wang, J.; Liu, A.; Yi, X.; Liu, X. Towards Benchmarking and Assessing Visual Naturalness of Physical World Adversarial Attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12324–12333.
47. Xiao, Y.; Liu, A.; Zhang, T.; Qin, H.; Guo, J.; Liu, X. RobustMQ: Benchmarking Robustness of Quantized Models. *arXiv* **2023**, arXiv:2308.02350.
48. Xiao, Y.; Liu, A.; Li, T.; Liu, X. Latent Imitator: Generating Natural Individual Discriminatory Instances for Black-Box Fairness Testing. In Proceedings of the 32th ACM SIGSOFT International Symposium on Software Testing and Analysis, Seattle, WA, USA, 17–21 July 2023.
49. Zhang, Y.; Gong, Z.; Zhang, Y.; Li, Y.; Bin, K.; Qi, J.; Xue, W.; Zhong, P. Transferable physical attack against object detection with separable attention. *arXiv* **2022**, arXiv:2205.09592.
50. Wang, D.; Jiang, T.; Sun, J.; Zhou, W.; Gong, Z.; Zhang, X.; Yao, W.; Chen, X. FCA: Learning a 3D full-coverage vehicle camouflage for multi-view physical adversarial attack. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington DC, USA, 7–14 February 2022; Volume 36, pp. 2414–2422.
51. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the Conference on Robot Learning, PMLR, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.
52. Rong, G.; Shin, B.H.; Tabatabaee, H.; Lu, Q.; Lemke, S.; Možeiko, M.; Boise, E.; Uhm, G.; Gerow, M.; Mehta, S.; et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. In Proceedings of the 2020 IEEE 23rd International conference on intelligent transportation systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–6.
53. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics: Results of the 11th International Conference*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 621–635.
54. Feng, S.; Yan, X.; Sun, H.; Feng, Y.; Liu, H.X. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nat. Commun.* **2021**, *12*, 748. [CrossRef] [PubMed]
55. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
56. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *1*, 91–99. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.