Road Decals as Trojans: Disrupting Autonomous Vehicle Navigation with Adversarial Patterns

Wei-Jia Chen, Chia-Yi Hsu
NYCU

Wei-Bin Lee
Hon Hai (Foxconn) Research Institute

Chia-Mu Yu, Chun-Ying Huang
NYCU

***Abstract*—The emergence of autonomous vehicles (AVs) represents a significant breakthrough in transportation. These vehicles use object detection algorithms to sense and interpret their environment, enabling them to navigate and make decisions autonomously. Therefore, object detection systems are essential to ensure the effectiveness and safety of AV operations. However, recent studies have shown that object detection systems based on deep neural networks are susceptible to interference from intentionally designed objects containing adversarial perturbations. In this paper, we investigate the dependability of AVs by designing physical adversarial patches (APs) to fool object detectors. To ensure that the APs work in the real-world AVs, our APs have the following designs. First, we use the Expectation Over Transformation (EOT) technique to make APs adaptive to environmental challenges such as distance, angle, and shadow. Instead of using colored APs, our APs are monochrome and their shapes are more controllable, making them more stealthy on the road. Most importantly, an object is confirmed by AVs only after the object is detected for consecutive frames; however, most existing APs can only work in static cases. Our APs overcome the above challenges and ensure attack success in dynamic cases. Our experimental results show that our AP can effectively attack YOLOv3-tiny.**

***Index Terms*—Adversarial Patch, Autonomous Vehicles, DNN**

I. Introduction

In autonomous vehicles (e.g., TESLA and Hon Hai [19]), object detection stands as a cornerstone for autonomous navigation, enabling the car controller to anticipate obstacles in potential future trajectories. Consequently, there is a strong inclination towards object detection algorithms that exhibit high levels of accuracy. In recent years, numerous advanced object detection systems have seen outstanding improvements such as YOLO [32], Fast R-CNN [7], and Single Shot multibox Detector (SSD) [24]. However, in recent years, security concerns regarding object detectors have arisen due to the vulnerability of deep neural networks (DNNs) to adversarial examples (AEs). These are carefully crafted malicious inputs that can fool DNNs into making incorrect predictions. Early research focused primarily on studying adversarial examples against image classifiers in digital spaces, which involved computing perturbations, reintegrating them into original images, and feeding them directly into classification systems [1], [4], [27], [28], [40]. In a more recent development, several studies [13], [17], [22], [29], [37], [44] have demonstrated the feasibility of adversarial examples (AEs) against image classifiers in the physical world. They achieved this by capturing images of the AEs and feeding them directly into the classifier.

AEs for Object Detectors. Attacking object detectors is more challenging than attacking image classifiers, primarily because adversarial examples (AEs) must fool both label predictions and object existence predictions. In addition, object detectors operate in dynamic environments where the relative positions and movements of objects and detectors are constantly changing. This dynamic environment is evident in fast-moving AVs or surveillance systems. Recently, there have been many efforts to attack object detectors in the physical scenario. These efforts can generally be categorized into four types: pixel-wise [26], [48], patch-based [12], [20], [34], [38], [41], [42], [50], [52], wearable [14], [39], [46], and optical-based [25].

Technical Challenges. Here, we consider the use of road decals as adversarial patches (APs) to fool the object detector of real-world AVs. However, the design of such APs faces the following technical challenges. First, there are various issues such as distance, angle, and shadow between the APs and the camera on AVs. These factors can seriously affect the success rate of APs if not handled properly. Second, existing APs mainly use adversarial perturbations, which are complex, colorful, and not easily reproducible. Using such adversarial perturbations on roads would be too conspicuous and difficult to achieve. Third, an object is confirmed by AVs only after the object is detected for consecutive frames. However, most existing APs [2], [6], [18], [21], [35], [51] can only work in static cases; i.e., an AP attack is successful if object detection is misled in a single frame. APs must be carefully designed to account for the above real-world constraint; otherwise, APs will not work to fool AVs.

Our AP is designed to address the above challenges. First, our AP overcomes issues such as distance, angle, and shadow between the AP and camera on AVs through EOT [2] in AP generation. Second, our AP generation is based on a generative adversarial network (GAN) [9]. More specifically, a predefined set of shapes and colors is used to generate APs, making the shape and color of the generated APs more controllable. Third, during the training of our GAN, we explicitly include consecutive frames in a batch. Such a small change ensures that our AP can work in dynamic cases, i.e. an AP can mislead the object detector for consecutive frames.

Overall, our contributions are summarized as follows:

- We propose a novel method to generate physical APs to misguide AVs. In contrast to the previous works, our APs are more controllable in terms of shape and color,

2833-292X/24/$31.00 ©2024 IEEE
DOI 10.1109/DSN-S60304.2024.00039

making the attack stealthier. Moreover, our APs work in misguiding AVs in dynamic cases.
- Extensive experiments were conducted to demonstrate the effectiveness of our APs in the real-world environment.

II. Related Works

In this section, we start by providing an overview of current object detection techniques. Following this, we present several physical adversarial attacks targeting image classifiers that are linked to our methodology, and we analyze the constraints of current physical adversarial attacks against object detectors.

A. Object Detection

There has been significant advancement in the field of object detection because of convolutional neural networks (CNNs). Current deep learning-based object detection methods can typically be classified into two categories, the one is the two-stage detector such as RCNN [8], Fast R-CNN [7], Faster R-CNN [33], and Mask R-CNN [11]. The other is the one-stage detector, such as YOLO [32] and SSD [24].

We choose the one-stage object detector, YOLO, to disrupt autonomous cars due to its well-known attributes of speed, effectiveness, and suitability for simulating self-driving scenarios. YOLO's main advantage lies in its ability to detect full images and webcam feeds in real-time. With a single feed-forward CNN, YOLO directly predicts class probabilities and bounding box offsets, resulting in faster processing speeds. Its exceptional efficiency and high accuracy render it an ideal choice for real-time processing systems, hence our decision to select it. In engineering, YOLOv3 is more widely used. However, we finally decide to use the YOLOv3-tiny instead of the standard YOLOv3. Compared to the standard YOLOv3 and its tiny versions, YOLOv3-tiny reduces the number of parameters, making it more suitable for applications with high-speed requirements.

B. Physical Adversarial Attacks

There have been many efforts to adversarial attacks against image classifiers. Early-stage research on adversarial examples exclusively concentrated on the digital domain, whereas there is currently a burgeoning interest in physical adversarial attacks against deep learning models. EOT [2], D2P [15] and ISPAttack [31] were earlier works belonging to pixel-wise attacks, where perturbations are distributed across the entire image to attack classifiers. Subsequently, there has been a shift towards creating adversarial perturbations in the form of patches or employing optical attacks. AdvPatch [3], AdvBug [47], ACOsAttack [23], and Tnt Attack [5] are all examples of creating APs to attack classifiers. ABBA [10], LightAttack [30], and LaserSpot [13] use projectors to craft adversarial perturbations against classifiers.

Apart from classifiers, researchers have also started conducting physical attacks against Facial Recognition (FR) and object detection systems. AdvEyeglasses [36] applies adversarial perturbations onto eyeglasses, leading to incorrect identification by FR systems when worn. AdvHat [17] creates adversarial perturbations as stickers and attaches them to hats, leading to misidentification in FR systems. Besides, AdvMakeup [22], SOPP [45], and AT3D [49] have the same objective that targets FR systems.

On the other hand, attacking object detection is more challenging due to the instabilities of the environment. Sava et al. [34] customize APs and employ EOT to enhance the robustness. Wang et al. [43] improved the resizing method in EOT for better attack outcomes by considering the realistic size variation of moving target objects. Jia et al. [16] proposed four types of attacks to generate APs targeting object detectors based on different objectives. Adversarial T-Shirt [46] printing adversarial perturbations onto clothing and overcoming the soft fabric's influence, successfully attacks object detectors. However, adversarial perturbations of existing methods are not easy to produce. For instance, when colorful adversarial perturbations are printed using a printer, it is necessary to consider the color range that the printer can reproduce, as well as the color discrepancies in the printed output, both of which can easily impact performance. In addition, colorful APs placed on the road easily attract attention and can be noticed as anomalies. Consequently, we are committed to designing simple APs, limiting them to only one color, and facilitating their creation. Moreover, placing these APs on the road will also make them less noticeable.

III. Methodology

Creating robust adversarial examples for real-world object detectors is challenging due to the dynamic interactions between objects and detectors, as well as the diverse environmental conditions. For example, as the AV advances, the car's camera captures images of road signs. As the distance to the target object decreases, the size of the object increases. Additionally, the angle varies accordingly. The relative movement between the object and the detector necessitates highly robust adversarial examples. Although the adversarial examples themselves are static, they must be able to withstand reasonable variations in size, shape, brightness, and other factors. In addition to robustness, we also take into account the complexity of the adversarial examples themselves because more complex examples are susceptible to additional errors when transitioning from digital to physical, thereby affecting the effectiveness of the attack. To create robust and feasible adversarial examples, we propose a simple adversarial patch attack. We utilize the generator of GAN to produce APs and enhance robustness using EOT techniques.

In this paper, we focus on white-box adversarial attacks, which entail accessing the target model, including its structure and parameters. In other words, attackers can compute gradients through the model's weights to craft effective adversarial examples. In the following, we begin with an overview of our attack's framework, briefly introducing the components employed. We then perform more details on how to generate APs by the Generative Adversarial Network (GAN) [9].

A. Overview

Inspired by [13], to attack victim objects, instead of one AP, we use several small-sized APs close to target objects.

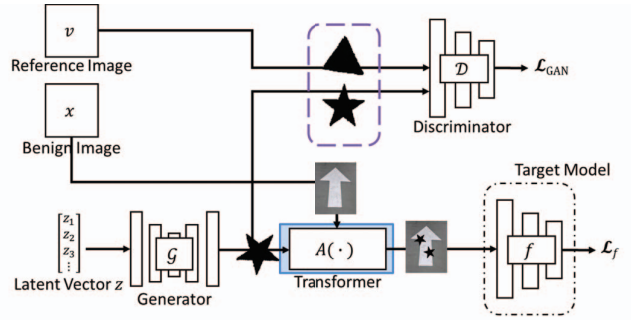


Fig. 1. The framework of our method.

(a) (b) (c)

Fig. 2. Three samples attaching APs with different angles in the same batch.

We use the GAN to generate APs. To achieve faster and more real-time image acquisition, we chose the YOLOv3-tiny architecture for our object detector. During the training process of the GAN, we simultaneously introduce the classification loss function corresponding to the object detector to ensure that APs can successfully cause misclassification by the object detector. Furthermore, we employ EOT techniques, denoted as $A(\cdot)$, to increase the robustness of the APs. Overall, our attack framework consists of a GAN and an object detector, as shown in Fig. 1.

B. Training Generator and Attack

While training the generator, we simultaneously attack the object detector. To attack a particular scene with N APs, each batch of training images during GAN training consists of **consecutive frames** related to that scene. We use the generator to synthesize an AP of size $k \times k$ and then copy $N \times$batch size times. We show an example of a batch of data with APs in Fig 2. We then perform EOT on the training images of this batch along with these APs. Regarding EOT, we use resize, rotation, gamma correction, and perspective. Perspective is used to simulate changes in the size of the target object as cars move, aiding successful attacks in three continuous frames. We also conduct additional experiments to demonstrate how different combinations of techniques in EOT affect the performance. After the EOT is completed, we remove the backgrounds from the APs and add N APs to each training image of this batch. Then we compute the loss function of the GAN. Note that the N APs in each image may have different rotation angles.

We formalize the loss of GAN as follows:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = \min_{\mathcal{G}}\max_{\mathcal{D}}&\mathbb{E}_{v\sim p_v}[\log\mathcal{D}(v)] \\ &+ \mathbb{E}_{z\sim p_z}[\log(1-\mathcal{D}(\mathcal{G}(z)))] \\ &+ \alpha\mathbb{E}_{z\sim p_z,\ \theta\sim p_\theta}[\mathcal{L}_f(A(\mathcal{G}(z),x,\theta),t)] \end{aligned} \tag{1}$$

where $\mathcal{G}$ and $\mathcal{D}$ are denoted as the generator and discriminator, respectively. v and x represent the reference and training images, respectively. z is the input to the generator and represents random noise. θ denotes transformation types, such as rotation. t denotes the target class into which we expect the object detector to classify the object. α controls the importance of the attack. In Eq. 1, the first two terms denote the conventional loss function utilized in training GANs, where the generator aims to synthesize images resembling those in the training set, while the discriminator is trained to differentiate between real and fake images. The third term in Eq. 1 is associated with the target attack and the complete equation is shown following:

$$\mathcal{L}_f = \ell(\mathcal{O}_\theta(A(\mathcal{G}(z)),x,\theta),t) \tag{2}$$

where ℓ is denoted as the cross-entropy function, while $\mathcal{O}_\theta$ represents parameters of the victim object detector. By incorporating $\mathcal{L}_f$ into the loss function of the generator, it aims to enforce the generated adversarial patches (APs) to be classified as the target class t when applied to the training sample x. In other words, minimizing $\mathcal{L}_f$ entails identifying APs with reduced cross-entropy with the target class t.

IV. Experiments

We conducted our attack in a white-box setting, focusing on targeted attacks to evaluate the results. We collected our own dataset of road images, consisting of 1000 images for training and 71 images for testing. Furthermore, we fine-tune the pre-trained object detector (pre-trained weights are from darknet53.conv.74) on our dataset with five labels such as person, word, mark, car, and bicycle, respectively. Regarding APs, we select the Four Shapes dataset[1] including star, circle, square, and triangle. It is worth noting that the shape is black with a white background in each image. To evaluate the performance of our attack, we use two indicators: the Percentage of Wrong-Class (PWC) and the Continuous Detection with Wrong-Class (CWC). PWC is computed by the following equation:

$$\text{PWC} = \frac{\text{number of frames are classified to the target class}}{\text{total number of frames of the video}} \times 100\%. \tag{3}$$

CWC indicates whether the object detector has consistently misclassified the wrong object class for three consecutive frames. According to our investigation, self-driving cars respond accordingly when they detect specific results in consecutive frames.

On the other hand, we evaluate the performance of our attack under three different challenges: rotation, speed, and angles. For rotation, we stand stationary and gently shake the camera to capture the results of the targeted object. In various velocity settings, we aimed the camera at the targeted object

[1]https://paperswithcode.com/dataset/shapes-1

	Rotation		Speed			Angles		
	fix	slight rotation	slow	normal	fast	$-15°$	$0°$	$+15°$
w/o Attack	0% / ✗	0% / ✗	0% / ✗	0% / ✗	0% / ✗	0% / ✗	0% / ✗	0% / ✗
Ours (w/ 3 consecutive frames)	**92%** / ✓	**80%** / ✓	**78%** / ✓	**45%** / ✓	**26%** / ✓	**70%** / ✓	**78%** / ✓	**74%** / ✓
Ours (w/o 3 consecutive frames)	62% / ✓	56%/ ✓	53% / ✓	38% / ✓	20% / ✗	58% / ✓	53% / ✓	53% / ✓
[34]	46% / ✓	38% / ✗	34% / ✓	19% / ✗	10% / ✗	22% / ✗	34% / ✓	30% / ✓

TABLE I
COMPARISON OF THE RESULTS UNDER THREE CHALLENGES IN A REAL-WORLD ENVIRONMENT.

and approached it at different speeds. We use three different speeds: slow, which is 15 km/hr; normal, which is 25 km/hr; and fast, which is 35 km/hr. Regarding different angles, the target object will be positioned on the left, center, or right side of the camera frame shown in Fig 3, while we move forward to simulate scenes of the car in motion.

(a) $-15°$ (b) $0°$ (c) $+15°$

Fig. 3. Visualization of the different-angle setting.

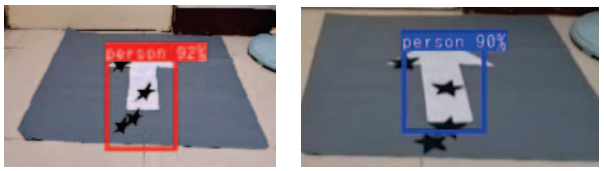
A. Experiment Setup

We performed targeted attacks in a white-box setting, where the attackers had access to all parameters of the object detector. For the coefficients of $\mathcal{L}_f$ we set $\alpha = 0.5$. For training the GAN, we choose Adam as the optimizer and set the batch size, learning rate, and epochs to 18, 10^{-4}, and 800. Regarding the APs, we select **star-shaped** ones because we found that APs with more angles give better attack results. We also explore the impact of various shapes, N and k on performance in Section IV-C. All experiments are run on NVIDIA Tesla V100 GPUs.

B. Experiment Results

We evaluate our attack on two different scenes: one is in a simulated environment, and another is in an underground parking lot. To carry out a physical attack, firstly, we ensure that APs attached to the images can successfully misclassify in the digital world. Secondly, we created physical APs and positioned them in real-world settings to replicate the successful digital attack scenarios. For all experimental results, we conduct three runs and average the results.

Simulated Environment We use $N = 4$ and $k = 60$ in a simulated environment. We used gray paper to simulate the road and white arrows to replicate real-life scenarios for filming in the bedroom. Fig 4 (a) and (b) present attacking successful results in the digital and real world, respectively. Table II shows that our attack are effective against rotation and various angles where PWCs are higher than 64%, and achieve all CWCs. However, achieving the robustness of APs at fast speeds presents a more challenging task in different



(a) Digital (b) Simulated

Fig. 4. Attack outcomes in both the digital and simulated realms.

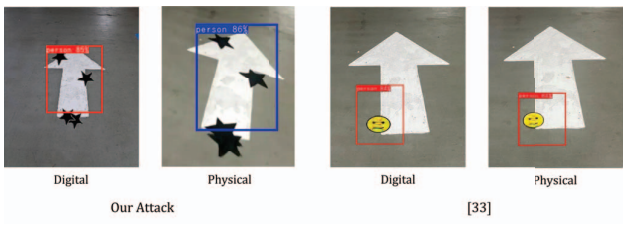


Fig. 5. Attack outcomes in both the digital and real-world realms.

speed settings. While configurations for different speeds pose challenges, we still achieve PWCs of over 85% in slow and normal settings, with some reaching CWCs.

	Rotation		Speed			Angles		
	fix	slight rotation	slow	normal	fast	$-15°$	0	$+15°$
PWC	100%	100%	100%	87%	40%	64%	87%	68%
CWC	✓	✓	✓	✓	✗	✓	✓	✓

TABLE II
COMPARISON OF THE RESULTS UNDER THREE CHALLENGES IN SIMULATING A REAL-WORLD ENVIRONMENT.

Real-Word Environment. We run our attack in the real world with $N = 6$ and $k = 60$. As the scenario shifts to a real-world environment, the effectiveness of the attack decreases somewhat due to the greater variability in environmental factors. We compare our attack with [34] and implement it by ourselves due to a lack of official codes. We chose to compare with [34] because their target model is also YOLOv3, and they also utilized many EOT techniques to generate colored APs. We confirm the success of all digital attacks and show pairs of digital and physical results in Fig 5, but when [34] is transferred to the real world, the performance of the attack significantly declines. In Table I, we perform the performance of w/o attack, our attack with and without

3 consecutive frames, and [34] with each data representing PWC / CWC. In this paragraph, we are solely focusing on our attack by taking account of 3 consecutive frames and will discuss the importance of consecutive frames later. As shown in Table I, the performance of the YOLOv3-tiny is quite stable when no attacks are encountered. When we place APs on the target object, it is highly likely that YOLOv3-tiny will misclassify it as a wrong category by our attack. However, the attack performance of [34] is quite poor in a real-world setting. Regarding rotation, our PWCs are at least 42% higher than [34]. In terms of different speeds, the PWCs differ by up to 40%, and we can achieve CWCs while [34] cannot. At different angles, [34] still exhibits a significant difference from our attack, with PWCs approximately lower than ours by 44%. The significant difference in performance may be attributed to slight discrepancies between the colors of the printed APs and their digital counterparts. However, such subtle differences may have a significant impact on the effectiveness of adversarial perturbations. This is also why we limit the colors of APs.

Importance of consecutive frames. As shown in Table I, we present the results of our attack in two settings: with and without considering three consecutive frames. It's evident to demonstrate that considering consecutive frames during the generation of APs, instead of a statically single frame, has a significant impact on the effectiveness of the attack. In the setting of rotation, PWCs without considering consecutive frames are reduced by up to 30%. At different speeds, PWCs differ by up to 25%, and the CWC cannot be achieved at fast speeds without continuous frames. Similarly, without considering consecutive frames, PWCs are also reduced by around 20% at different angles. The reason for such gaps in attack performance is that we evaluate the attack effectiveness by simulating vehicle behaviors to capture videos for calculating PWCs and CWCs. This is a dynamic process, and considering consecutive frames during the attack helps maintain successful attacks while the camera is in motion.

C. Ablation Study

In this paragraph, we conduct additional ablation studies to demonstrate how we set these hyperparameters in Section IV-B. These hyperparameters include the number of APs (N), the shapes of APs, APs of various sizes (k), and different combinations of tricks in EOT. In this section, all experiments are conducted in real-world settings, specifically in underground parking lots, for physical adversarial attacks.

Different N. To demonstrate the impact of different values of N, we maintain a constant total area for all APs and utilize star-shaped APs. All hyperparameters are followed in Section IV-A. Table III presents the outcomes for various values of N, indicating that $N = 4$ or $N = 6$ results in superior performance. Particularly, when considering various angles, their PWCs remain above 70%. Regarding the speed, the difference can be up to 10% and 8% at most under the settings of slow and fast, respectively. Given the fixed total area of all APs, it becomes evident that reducing the number of APs while increasing their individual sizes, or vice versa, has a detrimental effect on the outcome. Therefore, in Section IV-B, we use N as 4.

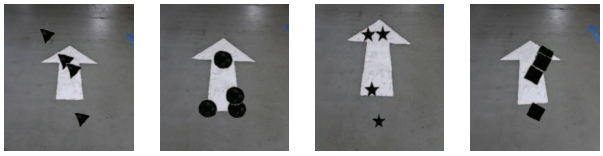
(a) $N = 2$ (b) $N = 4$ (c) $N = 6$ (d) $N = 8$

Fig. 6. Different numbers of physical adversarial patches.

	Speed			Angles		
N	slow	normal	fast	$-15°$	$0°$	$+15°$
2	68% / ✓	44% / ✓	12% / ✗	62% ✓	68% / ✓	66% / ✓
4	**78%** / ✓	45% / ✓	**26%** / ✓	70% / ✓	**78%** / ✓	**74%** / ✓
6	76% / ✓	**48%** / ✓	18% / ✗	**72%** / ✓	76% / ✓	70% / ✓
8	68% / ✓	40% / ✓	18% / ✗	60% / ✓	66% / ✓	59% / ✓

TABLE III
COMPARISON OF THE PERFORMANCE BASED ON DIFFERENT AMOUNTS OF APS UNDER TWO CHALLENGES IN A REAL-WORLD ENVIRONMENT.

Different Shapes. Interestingly, the shape of APs has a significant impact on performance. The Four Shapes dataset includes triangles, circles, stars, and squares. We follow the same hyperparameters mentioned in Section IV-A and set $N = 4$, $k = 60$. Fig. 7 illustrates samples featuring different shapes of APs that were actually used during our attacks. Table V clearly demonstrates that the performance of star-shaped APs significantly surpasses that of others, while the circular APs exhibit the poorest performance. The performance of star-shaped APs differs by at least 18% compared to other shapes in terms of PWC across different speeds, while maintaining PWCs of over 70% across various angles. Besides, star-shaped APs can achieve CWCs with all settings of speed and angles. From the experimental results, it can be observed that shapes with more angles, such as star-shaped APs, tend to yield better results, whereas shapes with fewer angles, such as circles that have no angles, exhibit the poorest performance.



(a) Triangle (b) Circle (c) Star (d) Square

Fig. 7. Visualization of physical adversarial patches with different shapes.

Different k. The size of individual APs also influences the attack results. We use the same hyperparameters as in Section IV-B and only adjust the value of k. Table VI shows that the best performance is $k = 60$. It's intuitive that APs that are too small are less likely to succeed in attacks. In Table VI, with k=20 being quite small, there is hardly any noticeable

	Speed			Angles		
Tricks	slow	normal	fast	$-15°$	$0°$	$+15°$
(1)+(2)+(3)+(5)	64% / ✓	42% / ✓	14% / ✗	62% / ✓	64% / ✓	58% / ✓
(1)+(2)+(4)+(5)	**78%** / ✓	45% / ✓	**26%** / ✓	70% / ✓	**78%** / ✓	**76%** / ✓
(2)+(3)+(4)+(5)	**76%** / ✓	44% / ✓	26% / ✗	**73%** / ✓	76% / ✓	71% / ✓
(1)+(3)+(4)+(5)	72% / ✓	**48%** / ✓	26% / ✗	72% / ✓	72% / ✓	70% / ✓
(1)+(2)+(3)+(4)	45% / ✓	18% / ✗	10% / ✗	45% / ✓	45% / ✓	35% / ✗
All	**78%** / ✓	45% / ✓	26% / ✗	70% / ✓	78% / ✓	74% / ✓

TABLE IV
COMPARISON OF THE PERFORMANCE BASED ON DIFFERENT COMBINATION OF TRICKS IN EOT UNDER TWO CHALLENGES IN A REAL-WORLD ENVIRONMENT.

	Speed			Angles		
Shapes	slow	normal	fast	$-15°$	$0°$	$+15°$
triangle	36% / ✓	20% / ✗	11% / ✗	33% / ✓	36% / ✓	36% / ✓
circle	27% / ✓	13% / ✗	8% / ✗	24% / ✓	27% / ✓	27% / ✓
Star	**78%** / ✓	**45%** / ✓	**26%** / ✓	**70%** / ✓	**78%** / ✓	**76%** / ✓
Square	34% / ✓	19% / ✓	10% / ✗	34% / ✓	34% / ✓	11% / ✓

TABLE V
COMPARISON OF THE PERFORMANCE BASED ON DIFFERENT SHAPES OF APS UNDER TWO CHALLENGES IN A REAL-WORLD ENVIRONMENT.

effect in the attack, as both the PWCs are only around 10% and the CWCs cannot be achieved. However, having APs that are too large is also detrimental to the attack, as oversized APs may obstruct too much of the target object, essentially preventing the detection of the object altogether. Therefore, the PWCs between k=80 and k=40 differ significantly, with the maximum difference reaching up to 40%. Based on the findings presented in Table VI, we opt for $k = 40$ to enhance attack effectiveness in Section IV-B.

(a) $k = 20$ (b) $k = 40$ (c) $k = 60$ (d) $k = 80$

Fig. 8. Visualization of physical adversarial examples with different k.

	Speed			Angles		
k	slow	normal	fast	$-15°$	$0°$	$+15°$
20	12% / ✗	8% / ✗	0% / ✗	10% / ✗	12% / ✗	11% / ✗
40	66% / ✓	40% / ✓	12% / ✗	60% / ✓	66% / ✓	63% / ✓
60	**78%** / ✓	**45%** / ✓	**26%** / ✓	**70%** / ✓	**78%** / ✓	**74%** / ✓
80	32% / ✓	12% / ✗	5% / ✗	36% / ✓	32% / ✓	32% / ✓

TABLE VI
COMPARISON OF THE PERFORMANCE BASED ON DIFFERENT COMBINATION OF $k \times k$ OF APS UNDER TWO CHALLENGES IN A REAL-WORLD ENVIRONMENT.

Techniques in EOT. We use EOT to enhance the robustness of APs and there are many tricks that you can choose such as rotation and brightness. However, in the process of generating APs, simply adding more tricks in EOT does not necessarily result in the best APs. Therefore, in this paragraph, we delve into which tricks exert a more pronounced influence on the effectiveness of our attack. We select five common tricks: (1) resize, (2) rotation, (3) brightness, (4) gamma, and (5) perspective. We demonstrate the effects of different combinations of these five tricks in EOT. We follow the hyperparameters used in Section IV-B. Table IV presents the comparison of the performance with different combinations of tricks in EOT. The best choice is (1)+(2)+(4)+(5), which is what we used in Section IV-B. According to Table IV, (5), perspective, has the most significant impact on the results of the attack. Without using (5), the PWCs decrease significantly, dropping by up to 33%. It can also be observed that compared to (3), (4) is more important. Reviewing the first two rows of the Table IV, it becomes apparent that the absence of (4) resulted in the most substantial decline in PWCs, up to 14%. While both (3) and (4) are related to brightness, (4) involves non-linear adjustments to brightness, which could be the reason why using (4) yields better results than (3).

V. CONCLUSION

In this paper, we propose a simple adversarial patch attack and consider more practical environmental factors such as speed. Placing APs on the road can lead to incorrect judgments by the object detector regarding road signs, potentially resulting in erroneous responses. This highlights the risks associated with autonomous vehicles and poses a threat to road safety. Differing from existing approaches, we impose stricter constraints on APs to mitigate potential large disparities during physical APs creation and to enhance their inconspicuousness when deployed on the road. The experimental results demonstrate that effective attacks can be achieved using simple APs under three different settings that represent practical scenarios where autonomous vehicles may encounter various driving behaviors. For future work, we aim to enhance the robustness of APs against varying speeds, particularly at higher speeds. Overall, we have designed a simple and effective patch-based attack, considering more realistic scenarios to evaluate the attack results, while also revealing the potential risks associated with AVs.

REFERENCES

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pages 484–501. Springer, 2020.

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[3] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[5] Bao Gia Doan, Minhui Xue, Shiqing Ma, Ehsan Abbasnejad, and Damith C Ranasinghe. Tnt attacks! universal naturalistic adversarial patches against deep neural network systems. *IEEE Transactions on Information Forensics and Security*, 17:3816–3830, 2022.

[6] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1000–1008, 2020.

[7] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[10] Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Jian Wang, Bing Yu, Wei Feng, and Yang Liu. Watch out! motion is blurring the vision of your deep neural networks. *Advances in Neural Information Processing Systems*, 33:975–985, 2020.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[12] Shahar Hoory, Tzvika Shapira, Asaf Shabtai, and Yuval Elovici. Dynamic adversarial patch for evading object detection models. *arXiv preprint arXiv:2010.13070*, 2020.

[13] Chengyin Hu, Yilong Wang, Kalibinuer Tiliwalidi, and Wen Li. Adversarial laser spot: Robust and covert physical-world attack to dnns. In *Asian Conference on Machine Learning*, pages 483–498. PMLR, 2023.

[14] Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7848–7857, 2021.

[15] Steve TK Jan, Joseph Messou, Yen-Chen Lin, Jia-Bin Huang, and Gang Wang. Connecting the digital and physical world: Improving the robustness of adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 962–969, 2019.

[16] Wei Jia, Zhaojun Lu, Haichun Zhang, Zhenglin Liu, Jie Wang, and Gang Qu. Fooling the eyes of autonomous vehicles: Robust physical adversarial examples against traffic sign recognition systems. *arXiv preprint arXiv:2201.06192*, 2022.

[17] Stepan Komkov and Aleksandr Petiushko. Advhat: Real-world adversarial attack on arcface face id system. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 819–826. IEEE, 2021.

[18] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[19] Jane Lanhee Lee and Debby Wu. Hon hai aims to use ai to expand its push into electric vehicles. https://www.bloomberg.com/news/articles/2023-10-18/hon-hai-aims-to-use-ai-to-expand-its-push-into-electric-vehicles, Oct 2023.

[20] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019.

[21] Juncheng Li, Frank Schmidt, and Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *International conference on machine learning*, pages 3896–3904. PMLR, 2019.

[22] Chang-Sheng Lin, Chia-Yi Hsu, Pin-Yu Chen, and Chia-Mu Yu. Real-world adversarial examples via makeup. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2854–2858. IEEE, 2022.

[23] Aishan Liu, Jiakai Wang, Xianglong Liu, Bowen Cao, Chongzhi Zhang, and Hang Yu. Bias-based universal adversarial patch attack for automatic check-out. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 395–410. Springer, 2020.

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[25] Giulio Lovisotto, Henry Turner, Ivo Sluganovic, Martin Strohmeier, and Ivan Martinovic. {SLAP}: Improving physical adversarial examples with {Short-Lived} adversarial perturbations. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1865–1882, 2021.

[26] Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv preprint arXiv:1712.02494*, 2017.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[28] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[29] Dinh-Luan Nguyen, Sunpreet S Arora, Yuhang Wu, and Hao Yang. Adversarial light projection attacks on face recognition systems: A feasibility study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 814–815, 2020.

[30] Nicole Nichols and Robert Jasper. Projecting trouble: Light based adversarial attacks on deep learning classifiers. *arXiv preprint arXiv:1810.10337*, 2018.

[31] Buu Phan, Fahim Mannan, and Felix Heide. Adversarial imaging pipelines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16051–16061, 2021.

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[34] Paul Andrei Sava, Jan-Philipp Schulze, Philip Sperl, and Konstantin Böttinger. Assessing the impact of transformations on physical adversarial attacks. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*, pages 79–90, 2022.

[35] Athena Sayles, Ashish Hooda, Mohit Gupta, Rahul Chatterjee, and Earlence Fernandes. Invisible perturbations: Physical adversarial examples exploiting the rolling shutter effect. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14666–14675, 2021.

[36] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. 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*, pages 1528–1540, 2016.

[37] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security (TOPS)*, 22(3):1–30, 2019.

[38] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*, 2018.

[39] Jia Tan, Nan Ji, Haidong Xie, and Xueshuang Xiang. Legitimate adversarial patches: Evading human eyes and detection models in the physical world. In *Proceedings of the 29th ACM international conference on multimedia*, pages 5307–5315, 2021.

[40] Yusuke Tashiro, Yang Song, and Stefano Ermon. Diversity can be transferred: Output diversification for white-and black-box attacks. *Advances in neural information processing systems*, 33:4536–4548, 2020.

[41] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection.

In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[42] Derui Wang, Chaoran Li, Sheng Wen, Qing-Long Han, Surya Nepal, Xiangyu Zhang, and Yang Xiang. Daedalus: Breaking nonmaximum suppression in object detection via adversarial examples. *IEEE Transactions on Cybernetics*, 52(8):7427–7440, 2021.

[43] Ningfei Wang, Yunpeng Luo, Takami Sato, Kaidi Xu, and Qi Alfred Chen. Does physical adversarial example really matter to autonomous driving? towards system-level effect of adversarial object evasion attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4412–4423, 2023.

[44] Zhibo Wang, Hongshan Yang, Yunhe Feng, Peng Sun, Hengchang Guo, Zhifei Zhang, and Kui Ren. Towards transferable targeted adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20534–20543, 2023.

[45] Xingxing Wei, Ying Guo, Jie Yu, and Bo Zhang. Simultaneously optimizing perturbations and positions for black-box adversarial patch attacks. *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[46] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 665–681. Springer, 2020.

[47] Hiromu Yakura, Youhei Akimoto, and Jun Sakuma. Generate (non-software) bugs to fool classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1070–1078, 2020.

[48] Kaichen Yang, Tzungyu Tsai, Honggang Yu, Tsung-Yi Ho, and Yier Jin. Beyond digital domain: Fooling deep learning based recognition system in physical world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1088–1095, 2020.

[49] Xiao Yang, Chang Liu, Longlong Xu, Yikai Wang, Yinpeng Dong, Ning Chen, Hang Su, and Jun Zhu. Towards effective adversarial textured 3d meshes on physical face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4119–4128, 2023.

[50] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 1989–2004, 2019.

[51] Yiqi Zhong, Xianming Liu, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15345–15354, 2022.

[52] Alon Zolfi, Moshe Kravchik, Yuval Elovici, and Asaf Shabtai. The translucent patch: A physical and universal attack on object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15232–15241, 2021.