(Extended Abstract) Road Decals as Trojans: Disrupting Autonomous Vehicle Navigation with Adversarial Patterns

Wei-Jia Chen[1], Chia-Yi Hsu[1], Chia-Mu Yu[1], Yu-Sung Wu[1], Ying-Dar Lin[1] and Wei-Bin Lee[2]
[1]National Yang Ming Chiao Tung University, [2]Hon Hai Research Institute

Abstract—**The emergence of autonomous vehicles (AVs), which rely heavily on advanced technologies such as object detection systems, 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, the development and optimization of object detection systems are essential to ensure the effectiveness and safety of autonomous vehicle operations. In this paper, we design a physical adversarial attack algorithm based on adversarial patches (APs) to deceive object detectors. Instead of using colored APs, our APs are monochrome, which makes it easier to implement on the road. In addition, to increase the robustness of the APs, we consider the Expectation Over Transformation (EOT) technique. Our experimental results show that our simple patches can effectively attack YOLO-V3, which consistently misidentifies the same object category for three consecutive frames.**

1. Introduction and Related Works

Object detection is the process of identifying semantic objects in images or video clips, with widespread applications in areas such as face detection, object tracking, and safety-critical tasks such as autonomous driving and intelligent video surveillance. In particular, in autonomous driving systems, object detectors play a critical role in tasks such as recognizing traffic signs, pedestrians, vehicles, traffic lights, and lanes. 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. In a more recent development, several studies [2]–[4] 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.

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 autonomous vehicles or surveillance systems. Recently, there have been many efforts to attack object detectors in the physical scenario. Most physical adversarial attacks involve creating adversarial patches (APs) and placing them on target objects to fool object detectors into detecting them as a wrong class, e.g., [5], [6]. Their main approach is to generate robust APs that are colorful by extending the range of image transformation. However, the application of colored APs on the road surface is highly challenging.

In this paper, our goal is to create robust adversarial patches to target state-of-the-art object detectors used in the real world, especially when considering speeds, wide angles, and diverse real-world scenarios. To facilitate the application of APs on the road surface, our APs will be of a single color for ease of implementation. Our contributions are summarized as follows: (1) We propose a novel method for attacking object detectors with monochrome APs. (2) We overcome several challenges such as speed and wobble.

2. Proposed Method

2.1. Overview

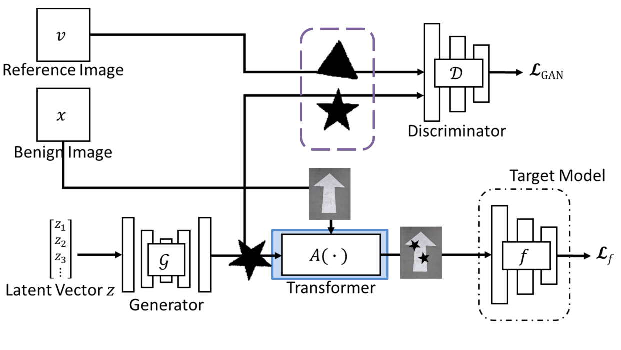

Figure 1. The framework of our method.

Inspired by [1], to attack victim objects, instead of one AP, we use several small-sized APs close to target objects. We use the Generative Adversarial Network (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.

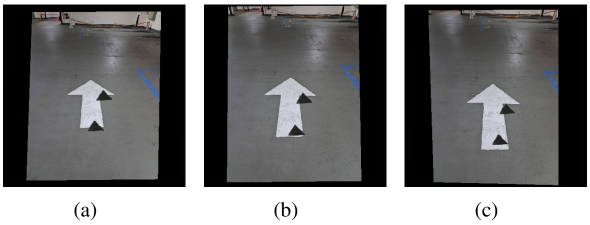

Figure 2. Visualization of 3 samples attaching APs with different angles in the same batch.

Overall, our attack framework consists of a GAN and an object detector, as shown in Fig. 1.

2.2. 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 Figure 2. We then perform EOT on the training images of this batch along with these APs. 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.

3. 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. 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:

$$\text{PWC} = \frac{\text{number of frames are classified to the target class}}{\text{total number of frames of the video}} \times 100\%. \tag{2}$$

CWC indicates whether the object detector has consistently misclassified the wrong object class for three consecutive frames.

3.1. 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 choose star-shaped ones because we found that APs with more angles give better attack results.

3.2. Experiment Results

We evaluate our attack on two different scenes: one is in a simulated environment, and another is in an underground parking lot. We use $N = 4$ and $k = 60$. Table 1 shows that our attack can overcome most challenges where PWCs are higher than 60%, and achieve CWCs except for the high speed in a simulated environment. We take screenshots from videos to show our simulated scenario.

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

TABLE 1. COMPARISON OF THE RESULTS UNDER THREE CHALLENGES IN SIMULATING A REAL-WORLD 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. However, we can still obtain CWCs in most settings.

	Rotation		Speed			Angles		
	fix	slight rotation	slow	normal	fast	$-15°$	$0°$	$+15°$
PWC	92%	80%	76%	44%	20%	26%	44%	28%
CWC	✓	✓	✓	✓	×	×	✓	×

TABLE 2. COMPARISON OF THE RESULTS UNDER THREE CHALLENGES IN A REAL-WORLD ENVIRONMENT.

References

[1] 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*, 2023.

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

[3] Chang-Sheng Lin, Chia-Yi Hsu, Pin-Yu Chen, and Chia-Mu Yu. Real-world adversarial examples via makeup. In *IEEE ICASSP*, 2022.

[4] Dinh-Luan Nguyen, Sunpreet S Arora, Yuhang Wu, and Hao Yang. Adversarial light projection attacks on face recognition systems: A feasibility study. In *IEEE/CVF CVPR Workshops*, pages 814–815, 2020.

[5] 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.

[6] 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.

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