Remote Perception Attacks against Camera-based Object Recognition Systems and Countermeasures

YANMAO MAN, Dept. of E.C.E., University of Arizona, USA
MING LI, Dept. of E.C.E., University of Arizona, USA
RYAN GERDES, Dept. of E.C.E., Virginia Tech, USA

In vision-based object recognition systems imaging sensors perceive the environment and then objects are detected and classified for decision-making purposes; e.g., to maneuver an automated vehicle around an obstacle or to raise alarms for intruders in surveillance settings. In this work we demonstrate how camera-based perception can be unobtrusively manipulated to enable an attacker to create spurious objects or alter an existing object, by remotely projecting adversarial patterns into cameras, exploiting two common effects in optical imaging systems, viz., lens flare/ghost effects and auto-exposure control. To improve the robustness of the attack, we generate optimal patterns by integrating adversarial machine learning techniques with a trained end-to-end channel model. We experimentally demonstrate our attacks using a low-cost projector on three different cameras, and under different environments. Results show that, depending on the attack distance, attack success rates can reach as high as 100%, including under targeted conditions. We develop a countermeasure that reduces the problem of detecting ghost-based attacks into verifying whether there is a ghost overlapping with a detected object. We leverage spatiotemporal consistency to eliminate false positives. Evaluation on experimental data provides a worst-case equal error rate of 5%.

CCS Concepts: • **Security and privacy → Hardware attacks and countermeasures**; • **Computing methodologies → Object detection**; • **Computer systems organization → Sensors and actuators**; *Robotics*.

Additional Key Words and Phrases: Sensor Attacks, Adversarial Examples, Autonomous Systems

1 INTRODUCTION

Object recognition (including localization and classification) have been widely adopted in autonomous systems, such as automated vehicles [77, 80] (e.g., for lane centering [12]) and unmanned aerial vehicles [1], as well as surveillance systems (e.g., smart home monitoring systems [63]). These systems first perceive the surrounding environment via sensors (e.g., cameras, LiDARs, and motion sensors) that convert analog signals into digital data, then try to understand the environment using object detectors and classifiers (e.g., recognizing traffic signs or unauthorized persons), and finally make a decision on how to interact with the environment (e.g., a vehicle may decelerate or a surveillance system raises an alarm).

While the cyber (digital) attack surface of such systems have been widely studied [10, 13], vulnerabilities in the perception domain are less well-known, despite perception being the first and critical step in the decision-making pipeline. That is, if sensors can be compromised then false data can be injected and the decision making process will indubitably be harmed as the system is not acting on an accurate view of its environment. Recent work has

Authors' addresses: Yanmao Man, Dept. of E.C.E., University of Arizona, Tucson, Arizona, USA, yman@arizona.edu; Ming Li, Dept. of E.C.E., University of Arizona, Tucson, Arizona, USA, lim@arizona.edu; Ryan Gerdes, Dept. of E.C.E., Virginia Tech, Arlington, VA, USA, rgerdes@vt.edu.

2378-962X/2023/5-ART $15.00
https://doi.org/10.1145/3596221

demonstrated false data injection against sensors in a remote manner via either electromagnetic (radio frequency) interference [67], laser pulses (against microphones [73], or LiDARs [7, 59, 70]), and acoustic waves [71]. These perception domain sensor attacks alter the data at the source, hence bypassing traditional digital defenses (such as crypto-based authentication or access control), and are subsequently much harder to defend against [19, 83]. These attacks can also be remote in that the attacker needn't physically contact/access/modify devices or objects.

Among the aforementioned sensors, cameras are more common/crucial for automated systems in the transportation and surveillance domains. Existing *remote* attacks against cameras are limited to, essentially, denial-of-service attacks [59, 78, 84], which are easily detectable (e.g., by tampering detection [62]). In this work, we consider attacks that cause a camera-based object recognition system to either misperceive actual objects or perceive non-existent objects by remotely injecting light-based interference into a camera, without blinding it. Formally, we consider *creation attacks* whereby a spurious object (e.g., a non-existent traffic sign, or obstacle) is seen to exist in the environment by a camera, and *alteration attacks*, in which an existing object in the camera view is changed into another attacker-chosen object (e.g., changing a STOP sign to YIELD or changing an intruder into a bicycle).

As it is not possible, due to optical principles, to directly project an image into a camera (without blocking the target object), we propose to exploit two common effects in optical imaging systems, viz., *lens flare effects* and *exposure control* to induce camera-based misperception. The former effect is due to the imperfection of lenses, which causes light beams to be refracted and reflected multiple times resulting in polygon-shape artifacts (a.k.a., *ghosts*) to appear in images [22]. Since ghosts and their light sources typically appear at different locations, an attacker can overlap specially crafted ghosts with the target object without having the light source blocking it. Auto exposure control is a feature common to cameras that determines the amount of light incident on the imager and is used, for example, to make images look more natural. An attacker can leverage exposure control to make the background of an image darker and the ghosts brighter, so as to make the ghosts more prominent (i.e., noticeable to the detector/classifier) and thus increase attack success rates. Fig. 1 presents an example of a creation attack, where we used a projector to inject an image of a STOP sign in a ghost, which is recognized as a STOP sign by YOLOv3 [61], a state-of-the-art object detector.

(a) Projector off (b) Projector on

Fig. 1. A STOP sign image was injected into a camera by a projector, which was detected by YOLOv3 [61].

However, the attack setup in Fig. 1 is ideal, in which the projector was close to the camera, and the attacker had direct access to the camera's output. In reality, the distance may be large (thus low attack pattern resolutions), and the attacker would have access to neither the hardware or firmware of the camera, nor to the image that the camera outputs (thus unaware of the appearance of ghosts in terms of their positions, colors, etc.). To improve the practicality of our attack, we first study an empirical projector-camera channel model by which the attacker predicts the resolution, location, distorted color, and brightness of injected ghost patterns, for a given projector-camera topology. Furthermore, the attacker formulates and solves an adversarial machine learning-based optimization problem [8, 76] to derive optimal attack patterns (of varying resolutions) to project, which will be classified/recognized by the object classifier/detector as the intended target class. The channel model is integrated within the optimization formulation, such that the attack patterns are resistant to channel effects and thus able to deceive the machine learning (ML) models under realistic conditions. We use self-driving and surveillance systems as two illustrative examples to demonstrate the potential impact of GhostImage attacks. Proof-of-concept experiments were conducted with different cameras, image datasets, neural network

architectures, and environmental conditions. Results show that our attacks are able to achieve attack success rates as high as 100% for image classification depending on the attack distance, and 88% for object detection.

To defend a camera-based object recognition system against GhostImage attacks, we analyze potential countermeasures across hardware and software: We find them either impracticable for the application, or vulnerable to a slightly more powerful attacker (e.g., to defeat fusion-based defenses, an adversary could compromise multiple, even heterogeneous, sensors simultaneously by incorporating our attack with other sensor attacks [7, 73]). Also, ML models may not be suitable for recognizing ghosts as a defense as they themselves are vulnerable to adversarial examples [76]. We propose a detection algorithm that first detects ghosts by calculating their locations based on the light source location. If a detected object overlaps with one of the calculated ghosts, that ghost is considered suspicious. False positives may arise because the ghost was actually caused by a natural light source. To differentiate adversarial ghosts from natural ones, we adopt the idea of spatiotemporal consistency: As a camera is moving (on a vehicle or simply sweeping), adversarial ghosts would likely keep overlapping with the object for multiple frames (because the attacker is tracking the camera), while natural ghosts likely shift or disappear. Evaluation shows the worst-case equal error rate (EER) is 5% derived from experimental data of measurement errors.

Our contributions are summarized as follows. **(a)** We are the first to study remote perception attacks against camera-based object recognition systems, whereby the attacker induces mis-recognition of objects by injecting light, conveying adversarially generated patterns, into the camera. **(b)** We leverage optical effects/techniques, namely, lens flare and auto-exposure control, to *realize* the attack and an adversarial machine learning-based optimization framework to find the optimal patterns that purposely mislead the ML models, thus *enhancing* the attack. **(c)** The efficacy of the attacks is demonstrated via experiments with varying neural network architectures, image datasets, cameras, across self-driving and surveillance scenarios, and in indoor/outdoor environments. Results show that GhostImage attacks are able to achieve attack success rates as high as 100% on image classifiers depending on the projector-camera distance, and 88% on a state-of-the-art object detector, YOLOv3. **(d)** A detection algorithm based on spatiotemporal consistency is developed, which reduces the attack detection problem to verifying whether a ghost overlaps with a detected object. It leverages knowledge of the lens configuration and object detector to distinguish adversarial ghosts from natural ones. Evaluation based on real-world error distribution indicates an EER of at most 5%.

Compared with our preliminary work [42], this version extends the applicability of our GhostImage attack to object detection (Sec. 4.4), and more importantly, introduces a countermeasure (Sec. 7). Evaluation results of both are presented in Secs. 5.3 and 7.6, respectively.

2 BACKGROUND

In this section, we will introduce optical imaging principles, including flare/ghost effects and exposure control, which we will exploit to realize GhostImage attacks. Then, we will discuss the preliminaries about neural networks and adversarial examples that we will use to enhance GhostImage attacks.

2.1 Optical Imaging Principles

Due to the optical principles of camera-based imaging systems, it is not feasible to simply point a projector directly at a camera and have projected patterns appear at the same location as image of the targeted object, as the projector has to obscure the object in order to make the two images overlap. Instead, we exploit lens flare effects and auto exposure control to inject adversarial patterns.

Lens flare effects [22, 79] refer to a phenomenon wherein one or more undesirable artifacts appear on an image because bright light is scattered or flared in a non-ideal lens system (Fig. 2). Ideally, all light beams should pass directly through the lens and reach the CMOS sensor. However, due to imperfections in the lens elements,

a small portion of light is reflected several times within the lens system and then reaches the sensor, forming multiple polygons (called "ghosts") on the image. The shape of polygons depends on the shape of the aperture. For example, if the aperture has six sides, there will be hexagon-shaped ghosts in the image. Normally ghosts are very weak and one cannot see them, but when a strong light source (such as the sun, a light bulb, a laser, or a projector) is present—unnecessarily captured by the CMOS sensor, though [24]—the ghost effects become visible. Fig. 2 shows only one reflection path, but there are many other paths, which is why there are usually multiple ghosts in an image.

Existing literature [22] on ghosts focuses on the simulation of ghosts given a detailed lens configuration, in which the algorithms simulate every possible reflection path. These models require white-box knowledge of the internal lens configuration, which is unrealistic for an attacker to possess. Rather, we assume that such information is only available to the defender who may be the manufacturer of the camera system or an owner supplied with such information. Accordingly, in Sections **??** and 4, we will introduce a lightweight end-to-end (black-box) channel model that an attacker trains to predict how ghosts would appear in the image in terms of locations, brightness, color, etc. In Section 7, on the other hand, we will assume a defender who possesses white-box knowledge of the lens configuration so that they can compute ghost locations instead of recognizing them using ML models. Note, that, this knowledge does not need to be kept secret from the attacker. In other words, the defense performs the same regardless of whether the attacker has the knowledge.

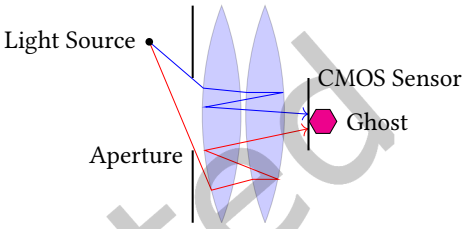


Fig. 2. Ghost effect principle

Exposure control mechanisms [31] are often equipped in cameras to adjust brightness by changing the size of the aperture or the exposure time. In this work we will model and exploit auto exposure control to manipulate the brightness balance between the targeted object and the injected attack patterns in ghosts to use as little attack power as possible.

2.2 Neural Nets and Adversarial Examples

Neural networks are used for object classfication and detection. We abstract a neural network-based image classifier [30] as a function $Y = f_\theta(x)$ (details are omitted due to the page limit and are, in any case, unnecessary to understand the attack). The input $x \in \mathbb{R}^{w \times h \times 3}$ (width, height and RGB channels) is an image, $Y \in \mathbb{R}^m$ is the output vector, and θ is the parameters of the network (which is fixed, thus we omit it for convenience). A softmax layer is usually added at the end of a neural network to make sure that $\sum_{i=1}^{m} Y_i = 1$ and $Y_i \in [0, 1]$. The classification result is then $C(x) = \text{argmax}_i Y_i$. The inputs to the softmax layer are called *logits*, $Z(x)$.

We use YOLOv3 [61] as an example of object detectors. Given an image x, the output of YOLO is a tensor in Shape $m \times m \times 3 \times 85$, where $m \times m$ is the number of cells of the grid that divides the image, 3 is the number of different shapes of anchors, and the last dimension contains 85 real numbers: The first four elements describe the bounding box, the fifth indicates the detection probability/confidence of whether the k-th anchor at the (i, j)-th cell contains an object at all, denoted by $P_x(i, j, k, x)$, and the remaining 80 elements represent the classification probability/confidence of a particular class c, denoted by $P_x^c(i, j, k, x)$.

An adversarial example [76] is denoted as y, where $y = x + \Delta$. Here, Δ is additive noise that has the same dimensionality as x. Given a benign image x and a target label t, an adversary wishes to find a Δ such that $C(x + \Delta) = t$ or $P_x^t(i, j, k, x + \Delta)$ is the highest, i.e., *targeted attacks*. Note that, in this paper, the magnitude of Δ is not constrained, since the images are usually not directly observed by human users; we still try to minimize Δ because it is proportional to attack power and, hence, attacker costs.

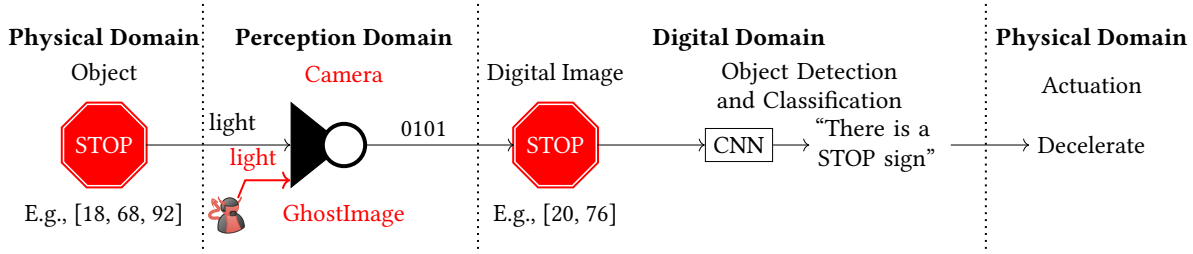


Fig. 3. Camera-based object recognition systems. GhostImage attacks target the perception domain, i.e., the camera, whose output is interpreted by (neural net-based) ML models in the digital domain. The ML recognition is fed back to the physical domain via actuation (e.g., decelerating).

3 SYSTEM AND THREAT MODEL

System and threat models are described, including two attack objectives and two types of attackers with incremental capabilities.

3.1 System Model

We assume an end-to-end camera-based object recognition system (Fig. 3) in which a camera captures an image of a scene with objects of interest. The image is then fed to a machine learning (ML) model to localize and classify the objects in it[1]. Autonomous systems increasingly rely on such ML models to make decisions and actions. If the recognition result is incorrect (e.g., modified by an adversary), wrong actions could be taken. For example, in a surveillance system, if an intruder is not detected, the house may be broken-in without raising an alarm.

3.2 Threat Model

We consider two different attack objectives. In **creation attacks** the goal is to inject a spurious (i.e., non-existent) object into the scene and have it be recognized (classified) as though it were physically present. For **alteration attacks** an attacker injects adversarial patterns over an object of interest in the scene that causes the object to be misclassified.

There are two types of attackers with differing capabilities: **Camera-aware attackers** possess knowledge of the victim's camera (i.e., they do not know the configuration of the lens system, nor post-processing algorithms, but they do possess the same type of camera used in the target system), from which they can train a channel model using the camera as a black-box. With such capabilities, they are able to achieve creation attacks and alteration attacks. **System-aware attackers** not only possess the capabilities of a camera-aware attacker but also know about the ML models including its architecture and parameters, i.e., black-box attacks on the camera but white-box attacks on the ML models (e.g., via side-channel attacks [9, 41, 87]). With such capabilities, they can mount creation attacks and alteration attacks, as well, but with higher attack success rates.

Both types of attackers are remote (unlike the lens sticker attack [32]), i.e., they do not have direct access to the hardware or the firmware of the victim camera, nor to the images that the camera captures. We assume that both attackers are able to track victim cameras and accurately aim light at them [7, 46, 78].

[1]Often, these two steps are achieved by a single object detector, e.g., YOLOv3 [61]. In this work, the term "object detection" is interchangeable with "object recognition".

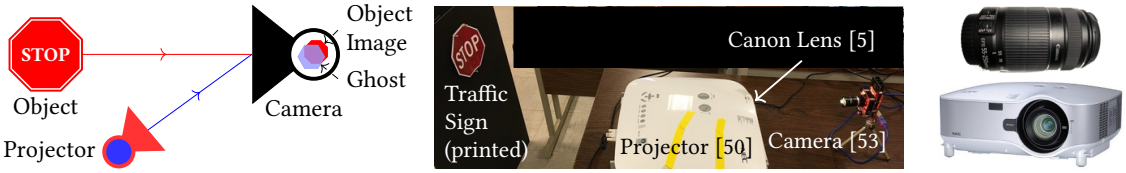


Fig. 4. (Left) Attack setup diagram. (Middle) In-lab experiment setup. (Right) Attack equipments: We replaced the original lens of the NEC NP3150 Projector [50] with a Canon EFS 55-250 mm zoom lens [5].

4 GHOSTIMAGE ATTACK

In this section, we first introduce camera-aware attacks where the attacker is able to inject arbitrary patterns in the perceived image of the victim camera using a COTS projector. Due to the drawbacks of camera-aware attacks, we improve GhostImage attack by introducing system-aware attackers who further take advantage of the vulnerability of ML models used for image processing. Note that, the latter inherits from the former all the attack capabilities, objectives, hence challenges as well.

4.1 Camera-aware Attack

Since we assume that the attacker does not have access to the images that the targeted camera captures, they will need to predict how ghosts appear in the image so as to align it with the object image. There are three main challenges.

(1) The locations of ghosts should be predicted given the relevant positions of the projector and the camera, so that the attacker can align the ghost with the image of the object of interest to achieve alteration attacks. *Solution:* Based on experiments, we propose a model that predicts the ghost location given the camera matrix, and the 3-D coordinates of both devices.
(2) Since a projector can inject shapes in ghost areas, the attacker needs to find out the maximum resolution of injectable shapes. *Solution:* We study a model that outputs the maximum resolution, given the projector-camera distance, the projector's throwing ratio and resolution, and the size of the camera's aperture.
(3) Realizing an attack derived from the position and resolution models above is challenging with a limited budget. *Solution:* We propose to replace the factory lens of our projector with a zoom lens made for DSLR cameras. See Fig. 4 for the experimental setup.

Details about the camera-aware attack and its experiments can be found in Sec. **??**[2].

4.2 System-aware Attack

Experimental results reveal some limitations of the camera-aware attack. First, increasing distances results in lower success rates because the classifier cannot recognize the resulting low-resolution images. Second, there are large gaps between digital domain results and perception domain results, as channel effects (which cause the inconsistency between the intended pixels and the perceived pixels) are not taken into account. In this section, we resolve these limitations and improve GhostImage attacks' success rates by proposing a framework which consists of a channel model that predicts the pixels perceived by the camera, given the pixels as input to the projector. We also include an optimization formulation based on which the attacker can solve for optimal attack patterns that cause mis-recognition by the target ML model.

[2]A reference that is prepended with an 'S' is a cross-reference to Supplementary Material.

Technical Challenges: First, the injected pixel values are often difficult to control as they exhibit randomness due to variability of the channel between the projector and the camera, thus the adversary is unable to manipulate each pixel deterministically. Second, to achieve optimal results, the attacker needs to precisely predict the projected and perceived pixels, thus channel effects must be modeled in an end-to-end manner, i.e., considering not only the physical channel (air propagation), but also the internal processes of the projector and the camera. Lastly, the resolution of attack patterns is limited by the attack distance and the projector lens (Eq. **??**), thus the ghost patterns must be carefully designed to fit the resolution with limited degrees of freedom.

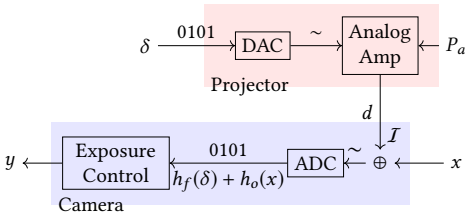


Fig. 5. Projector-camera channel model

4.2.1 System-aware Attack Overview. The system-aware attacker aims to find optimal patterns that can cause mis-recognition by the target model with high confidence by taking advantage of the non-robustness of neural networks [76]. We adopt an adversarial example-based optimization formulation into GhostImage attacks, in which the attacker tries to solve

$$\Delta^* = \arg\min_{\Delta} \quad \|\Delta\|_p + c \cdot \mathcal{L}_{\text{adv}}(y, t, \theta), \tag{1}$$

where Δ is the digital attack pattern as input to the projector, y is the perceived image of the object of interest under attacks, t is the target class/label, and θ represents the targeted neural network. $\|\cdot\|_p$ is an ℓ_p-norm that measures the magnitude of a vector, and $\mathcal{L}_{\text{adv}}$ is a loss function indicating how (un)successful Δ is. The actual form of $\mathcal{L}_{\text{adv}}$ depends on whether the target model is an image classifier (Eq. 3) or an object detector (Eq. 4). Here, we aim to minimize the power of the projector required for a successful attack, meanwhile maximizing the successful chance of attacks. These two objectives are balanced by a constant c.

More importantly, in (1) y is the final perceived image used as input to the neural network, which is estimated by our channel model in an end-to-end style (Fig. 5), in which δ [3] is the input to the projector, and y is the resulting image captured by the camera. The model can be formulated as

$$y = g\left(h_f(\Delta) + h_o(x)\right). \tag{2}$$

where $h_f(\Delta)$ is the ghost model that estimates the perceived adversarial pixel values in the ghost. For simplicity we let $h_o(x) = x$ because the attacker possesses the same type of the camera so that x can be obtained a priori, and $g(\cdot)$ is the auto exposure control that adjusts the brightness. Sec. **??** detail the design of y for spoofing image classification and object detection respectively.

We first briefly discuss Eq. 2 in Sec. 4.2.2, then discuss Eq. 1 for spoofing an image classifier in Sec. 4.3. Later in Sec. 4.4, we show how to defeat object detection.

4.2.2 Projector-Camera Channel Model. We consider the projector to camera channel model (Fig. 5) in which δ is an RGB value the attacker wishes to project which is later converted to an analog color by the projector. The attacker can control the power (P_a) of the light source of the projector so that the luminescence can be adjusted. The targeted camera is situated at a distance of d, which captures the light coming from both the projector and reflected off the object (x). The illuminance received by the camera from the projector is denoted as $\mathcal{I}$. The camera converts analog signals into digital ones, based on which it adjusts its exposure, with the final, perceived RGB value being y. An ideal channel would yield $y = x + \delta$ but due to channel effects, we need to find a way to

[3]Different than Δ which is a $w \times h \times 3$ tensor, δ is a single pixel with dimension 3×1 for the convenience of the analysis.

adjust the projected RGB value such that the perceived RGB value is as intended, i.e., to find the appropriate x given y. Due to the page limit, the modeling details are discussed in Sec. **??**.

4.3 Spoofing Image Classification

Here, we briefly discuss the formulation and results of defeating image classification.

Attack Loss: When targeting an image classifier, the attacker aims to minimize

$$\mathcal{L}_{\text{adv}}(y,t) = \max\left\{-\kappa, \max_{i:i\neq t}\{\mathbb{E}[Z_i(y)]\} - \mathbb{E}[Z_t(y)]\right\}, \tag{3}$$

where $\mathbb{E}[Z_i(y)]$ is the expectation of logits values of Class i given the input y. Term $\max_{i:i\neq t}\{\mathbb{E}[Z_i(y)]\}$ is the highest expected logits value among all the classes except the target class t, while $\mathbb{E}[Z_t(y)]$ is the expected logits value of t. Here, κ controls the logits gap between $\max_{i:i\neq t}\{\mathbb{E}[Z_i(y)]\}$ and $\mathbb{E}[Z_t(y)]$; the larger the κ is, the more confident that Δ is successful [8]. The attacker needs $\mathcal{L}_{\text{adv}}$ to be as low as possible so that the neural network would classify y as Class t (See Sec. **??** for details).

Evaluation Results: We tested system-aware attacks on image classification. Results show that system-aware attacks are roughly 100% more effective on average than camera-aware attacks; this is because we make use of adversarial ML optimization formulation with the consideration of channel effects. We also demonstrate the robustness of the attack on different datasets, NN architectures, cameras, and environmental conditions. Evaluation details are presented in Sec. 5.

4.4 Spoofing Object Detection

So far, we have introduced a system-aware GhostImage attacker who manipulates only the image classifier (Sec. 4.3). However, a more practical setting would be to compromise object detection models. You-only-look-once (YOLO) [61] is one example of such models. Due to its fast processing and high accuracy, YOLOv3 is being used in industrial self-driving car platforms, such as Apollo [3]. In this section, we use YOLOv3 as an example to demonstrate how a system-aware attacker is able to create an object that is misperceived by an object detection algorithm.

4.4.1 Challenges and Solutions. Compared with the functionality of image classification which is to assign a label for the entire image, object detection is to localize and classify (usually multiple) objects from the image. Due to such differences, it is still challenging to spoof YOLO even though we are able to spoof image classifiers. Here, we will discuss the technical challenges and how we address them. We still formulate an optimization problem similar to Eq. 1 but we need to update the objective function with additional constraints that capture the characteristics of YOLO and the projector-camera channel effects.

Shape and Contrast: According to our experiments, the shape of the injected patterns has a large impact on the attack effectiveness in the perception domain. For example, projecting a pattern in an octagon shape is more likely to be detected as a STOP sign by YOLO than a square-shaped pattern. Moreover, due to the channel effects (Sec. 4.2.2), the induced patterns do not typically have clear edges, thus indistinguishable from the background. In order to emphasize the shape, the edge of it needs to be of high contrast especially because the blurring effect is inevitable. As a result, when formulating the optimization problem, the attacker needs to add a shape constraint that represents the target class, and another constraint that increases the brightness of the pixels on the edge so that in the perceived image the pattern can easily be distinguished from the background (Fig. **??**).

Image Context: Since the official YOLOv3 model that we evaluate was trained as a regression problem where all objects (including their bounding boxes and labels) of the input image were fed to the optimizer, it implicitly learned the context of the scene [65, 92], e.g., YOLO might have learnt that "there is typically a pole under a STOP sign although the bounding box only includes the octagon." In our experiments for inducing a STOP sign,

placing a pole under the octagon pattern does increase the recognition confidence, which can be achieved by changing the shape constraint slightly (Fig. **??**). In addition, the background that surrounds the object also has an impact. In the case of our experiment setup, there is a blue-tinted ghost overlaying with the ghost that contains projected attack patterns, which distorts the attack pattern significantly. However, unless acquiring two or more light sources, the attacker would not be able to change the background (except its brightness). To overcome this, we instead update our channel model to simulate the second ghost as a constraint in the optimization (Fig. **??**), so that the optimizer finds optimal patterns anticipating the other ghost.

Robustness to Shift: As mentioned in Sec. **??**, only a small portion of the projected pixels are captured by the camera, which is denoted as S_f (Fig. **??**). As the relative movement between the projector and the camera occurs, S_f shifts within the projection plane S. Although the adversary is able to track the camera, it would still be better if we can relax the requirement of precisely controlling the position of S_f. To improve the attack practicality, we can equip adversarial patterns $\mathbf{\Delta}$ with shift-invariance. This means, wherever S_f is within S, the ghost that contains S_f could always deceive YOLO. To achieve shift-invariance, we tile the pattern $\mathbf{\Delta}$ into a canvas and apply masks on the canvas, where different masks reveal/uncover random portions (equivalent to S_f in a fixed size) of the canvas (Fig. 6). Finally, we ask the optimizer to find a pattern $\mathbf{\Delta}$ that can succeed under all masks. Another benefit of masks is that we can easily vary the pattern shape in order to mimic different target classes.

Loss Function: Recall that YOLOv3 outputs a tensor in Shape $m \times m \times 3 \times 85$ while an image classifier outputs simply a vector (Sec. 2.2), we need to redesign our loss function $\mathcal{L}_{\text{adv}}$ (Eq. 1) that measures how unsuccessful so far the pattern is. Generally, there are two terms that the attacker aims to manipulate. First, they aim to deceive YOLO that there is an object at (i, j, k) by increasing $P_x(i, j, k)$. Second, the attacker needs to make YOLO believe that the object is most likely of a specific class t by increasing $P_x^t(i, j, k)$. The attacker can either specify (i, j, k) at which the portion uncovered (by the mask) is, or simply using all permutations of (i, j, k). In our experience, the latter actually works faster because tensor slicing is not supported that well in TensorFlow.

4.4.2 Overview of Optimization. The formulation is the same as (1), with the loss function of creation attacks,

$$\mathcal{L}_{\text{adv}}(y, t, \theta) = -\sum_{M_f \in \mathcal{M}} \sum_{i,j,k} \mathbb{E}_y \left[P_y(i, j, k) + P_y^t(i, j, k) \right], \tag{4}$$

where $\mathcal{M}$ is a set of masks that reveal different portions of the canvas $\mathbf{\Lambda}$ to achieve shift-invariance. In each mask M_f, the values are either 0, 1, or $b > 1$. 0 means that pixel is covered and 1 means uncovered; b's represent the edge of the shape which is made brighter. With this setting, when we take an element-wise product (denoted as $\otimes$) between a canvas and a mask, i.e., $\mathbf{\Lambda} \otimes M_f$, a pattern with an arbitrary shape and bright edges can be formed (Fig. **??**).

Moreover, y is derived by our channel model (Eq. 2), from the background x, the second ghost of interest, and the masked adversarial pattern $\mathbf{\Lambda} \otimes M_f$. We are taking the expectation of y here because y is a random variable (Sec. **??**). See Fig. 6 for some examples of Pattern $\mathbf{\Delta}$, Canvas $\mathbf{\Lambda}$, the masked canvas, and the second ghost of interest. Details of the formulation can be found in Sec. **??** from the supplementary material.

5 SYSTEM-AWARE ATTACK EVALUATION

In this section, we consider camera-based object recognition systems, as used in self-driving vehicles and surveillance systems, to illustrate the potential impact of our attacks. We present proof-of-concept system-aware attacks in terms of *attack effectiveness*, namely how well system-aware attacks perform in the same setup as camera-aware attacks (Sec. **??**), and *attack robustness*, namely how well system-aware attacks are when being evaluated in different setups. Sec. 5.1 and 5.2 discuss the results of attacks against image classifers, while in Sec. 5.3, results on object detection are presented. The image classifier is trained on LISA [47] and yields 96% accuracy.

Fig. 6. An example of shift-invariance. (a) A shift-invariant pattern in 8×8 resolution (zoomed in for a better presentation). (b) A canvas that is full of (a) in 416×416 resolution which is the input size of YOLOv3. (c)(d) YOLOv3 detects it as a STOP SIGN at different locations. The light, blue circle is the simulated second ghost. (e) A perceived ghost conveying the same pattern is detected as a STOP sign (zoomed in).

We will again use attack success rates as our metric. We used the Adam Optimizer [25] to solve our optimization problems. There are two sets of results: *Emulation results* refer to the recognition results on emulated, combined images of benign images and attack patterns using our channel model (Equation **??**). Emulation helps us conduct scalable and fast evaluations of GhostImage attacks before conducting real-world experiments[4]. *Experimental results* refer to the recognition results on the images that are actually captured by the victim cameras when the projector is on.

5.1 Attack Effectiveness

To compare with camera-aware attacks, system-aware attacks are evaluated in a similar procedure, targeting a camera-based object classification system with the LISA dataset and its classifier. The system uses an Aptina MT9M034 camera [53] in an in-lab environment.

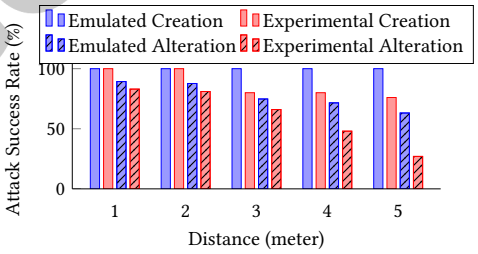


Fig. 7. System-aware creation and alteration

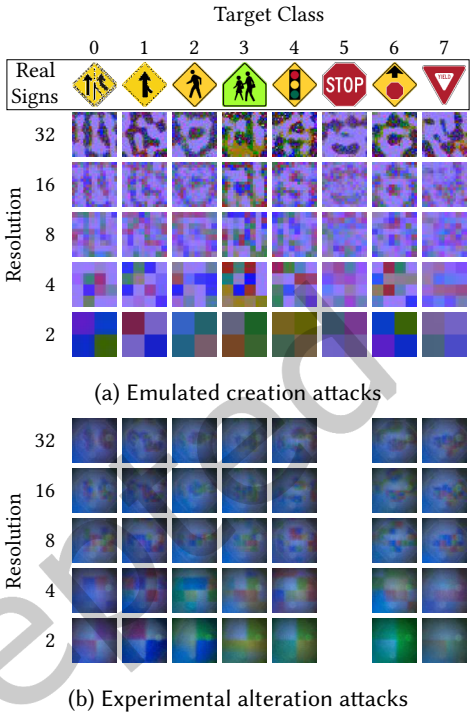
5.1.1 Creation attacks. For emulated creation attacks, all distances (or all resolutions) yield attack success rates of 100% (Fig. 7), which means that our optimization problem is easy to solve. In terms of computational overhead, we need roughly 30 s per image at 2×2-resolution, and 10 s at 4×4 or above (because of more degrees of freedom) using an NVIDIA Tesla P100 [52]. Fig. 8a shows examples of emulated attack patterns for creation attacks, along with the images of real signs on the top. Interestingly, high-resolution shapes do look like real signs. For example, we can see two vertical bars for ADDEDLANE, and also we can see a circle at the middle south for STOPAHEAD, etc. These results are consistent with the ones from the MNIST dataset [57] where we could also roughly observe the shapes of digits. Secondly, they are blue tinted because our channel model suggests that ghosts tend to be blue, thus the optimizer is trying to find "blue" attack patterns that are able to deceive the classifier.

Interestingly, the all k resulting patterns of solving the optimization problem targeting one class from k different (random) starting points look similar to the ones shown in Fig. 8a. However, CIFAR-10 [27] and ImageNet [14] yield much different results: Those patterns look rather random compared to the results from LISA or MNIST. The reason might be that in CIFAR-10 images in the same category are still very different, such as two different cats, but in LISA two images of STOP signs are not as different as two cats.

[4]Source code is available at https://github.com/harry1993/ghostimage

For the experimental results of creation attacks, we see that as distances increase, success rates decrease a little (Fig. 7), but much better than the camera-aware attacks (Table **??**), because the optimization formulation helped find those optimal attack patterns with high confidence. Overall, the emulated attacks perform better than the experimental attacks because we keep the channel model (Eq. 2) as simple as possible to reduce computation complexity during optimization (which sacrifices accuracy). While we evaluated up to five meters in our experiments, larger attack distances are possible. See Sec. **??** where we discuss how the ghost resolution depends on the throwing ratio and the attack distance.

5.1.2 Alteration attacks. The emulated and experimental results of alteration attacks are shown in Fig. 7. Compared with creation attacks, alteration attacks perform a bit worse, especially for large distances (three meters or further). This is because the classifier also "sees" the benign image in the background and tends to classify the entire image as the benign class. Moreover, the alignment of attack patterns and the benign signs is imperfect. However, when we compare Fig. 7 with Table **??** for camera-aware alteration attacks, we can see large improvements. Fig. 8b provides an example of system-aware alteration attacks in the perception domain, which were trying to alter the (printed) STOP sign into other signs: They look "blue" as the channel model predicted. The fifth column is not showing as it is trivial to alter a STOP into a STOP.



(a) Emulated creation attacks

(b) Experimental alteration attacks

Fig. 8. System-aware attack pattern examples.

5.2 Attack Robustness

We evaluate the robustness of our attacks in terms of different datasets, environments, and cameras.

5.2.1 Different image datasets. Here we evaluate our system-aware attacks on two other datasets, CIFAR-10 [27] and ImageNet [14], by emulation only because previous results show that our attack emulation yields similar success rates as experimental results.

CIFAR-10. The network architecture and model hyper parameters are identical to [8]. The network was trained with the distillation defense [58] so that we can evaluate the robustness of our attacks in terms of adversarial defenses. A classification accuracy of 80% was achieved. The evaluation procedure is similar to Sec. **??**. Results are shown in Fig. 9. The overall trend is similar to the LISA dataset, but the success rates are significantly higher. The reason might still be the large variation within one class (Section 5.1.1), so that the CIFAR-10 classifier is not as sure about one class as the LISA classifier is, hence is more vulnerable to GhostImage attacks.

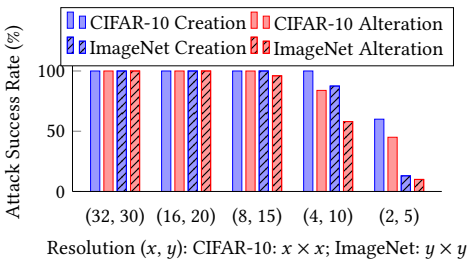


Fig. 9. System-aware attacks on CIFAR-10 and ImageNet

ImageNet. We used a pre-trained Inception V3 neural network [75] for the ImageNet dataset to evaluate the attack robustness against large networks. Since the pre-trained network can recognize 1000 classes, we did not iterate all of them [8]. Instead, for alteration attacks, we randomly picked ten benign images from the validation set, and twenty random target classes, while for creation attacks, the "benign" images were purely black. Results are given in Fig. 9.

For high resolutions ($\geq 15 \times 15$), the attack success rates were nearly 100%. But as soon as the resolutions went down to 10×10 or below, the rates decreased sharply. The reason might be that in order to mount successful *targeted* attacks on a 1000-class image classifier, a large number of degrees of freedom are required. 10×10 or lower resolutions plus three color channels might not be enough to accomplish targeted attacks. To verify this, we also evaluated untargeted alteration attacks on ImageNet. Results show that when the resolutions are 1×1 or 2×2, the success rates are 50% or 80%, respectively. But as soon as the resolutions go to 3×3 or above, the success rates reach 100%. Lastly, similar to CIFAR-10, system-aware attacks on ImageNet are more successful than on LISA due to high variation within one class.

5.2.2 Outdoor experiments. In order to evaluate system-aware attacks in a real-world environment, we also conducted experiments outdoor (Fig. 10), where the camera was put on the hood of a vehicle that was about to pass an intersection with a STOP sign. The attacker's projector was placed on the right curb, and it was about four meters away from the camera. The experiments were done at noon, at dusk and at night (with the vehicle's front lights on) to examine the effects of ambient light on attack efficacy. The illuminances were 4×10^4 lx, 4×10^3 lx, and 30 lx, respectively. The experiments at noon were unsuccessful due to the strong sunlight. Although more powerful projectors [4] could be acquired, we argue that a typical projector is effective in dimmer environments (e.g., cloudy days, at dawn, dusk, and night, or urban areas where buildings cause shades), which accounts for more than half of a day. See Sec. 6 for more discussion on ambient lighting conditions.

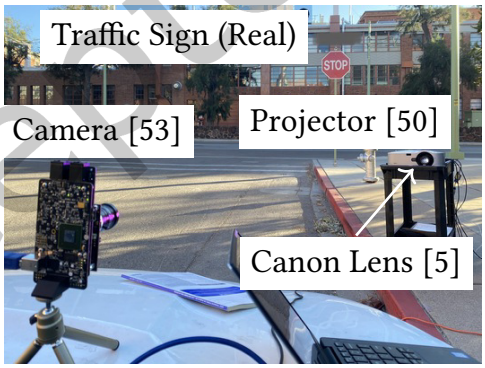


Fig. 10. Outdoor experiment setup

Results (Tab. 1) of the other cases show that the success rates are 30% lower than our in-lab experiments (the four-meter case from Fig. 7), because we used our in-lab channel model directly in the road experiments without retraining it, and also the environmental conditions are more unpredictable. Moreover, the attack rates on altering some classes (e.g., the STOP sign) into three other signs (e.g., YIELD) were 100%, which is critical as an attacker can easily prevent an autonomous vehicle from stopping at a STOP sign.

5.2.3 Different cameras. Previously, we conducted GhostImage attacks on Aptina MT9M034 camera [53] designed for autonomous driving. Here, we evaluate two other cameras, an Aptina MT9V034 [54] with a simpler lens design, and a Ring indoor security camera [63] for surveillance.

Table 1. Outdoor alteration success rates

Success rates of	Noon	Dusk	Night
Overall	0%	51%	42.9%
STOP → YIELD	0%	100%	100%
STOP → ADDEDLANE	0%	100%	100%
STOP → PEDESTRIAN	0%	100%	100%

Aptina MT9V034. We mounted system-aware creation attacks against the same camera-based object classification system as in Section 5.1 but we replaced the camera with the Aptina MT9V034 camera. Since this camera has a smaller aperture size and also a

simpler lens design than Aptina MT9M034, for a distance of one meter, only 16×16-resolution attack patterns could be achieved (previously we had 32×32 at one meter). We did not train a new channel model for this camera, so the attack success rate at one meter was only 75%, which is 25% lower than the Aptina MT9M034 camera. As the distances increased up to four meters, creation attacks yielded success rates as 46.25%, 33.75%, and 12.5%, respectively. Another reason why the overall success rate was lower is that even though the data sheet of Aptina MT9V034 [54] states that the camera also has the auto exposure control feature, we could not enable the feature in our experiments. In other words, system-aware creation attacks did not benefit from the exposure control. This, on the other hand, indicates the robustness of GhostImage attacks: Even without taking advantage of exposure control, the attacks were still effective, with attack success rates as high as 75%.

Ring indoor security camera. We tested GhostImage untargeted attacks against a Ring indoor security camera [63] on the ImageNet dataset. To demonstrate that our attacks can be applied to surveillance scenarios, we assume the camera would issue an intrusion warning if a specific object type is detected by the Inception V3 neural network [75]. The attacker's goal is to change an object for an intruder class to a non-intruder class. However, we could not find "human", "person" or "people", etc. in the output classes, we instead used five human related items (such as sunglasses) as the benign classes. We found six images from the validation set of ImageNet, of which top-1 classification results are one of those five benign classes. The six images were displayed on a monitor. For each benign image, we calculated ten alternative 3×3 attack patterns (the highest resolution at one meter by the Ring camera). Results show that for all six benign images, system-aware attacks achieved untargeted attack success rates of 100% (Table 2).

Table 2. GhostImage untargeted alteration attacks against Ring camera on ImageNet dataset in perception domain

Index	Benign Class	Rate	Common Prediction
19992	fur boat	100%	geyser, parachute
21539	sunglasses	100%	screen, microwave
22285	sunglasses	100%	plastic bag, geyser
31664	sarong	100%	jellyfish, plastic bag
2849	sweatshirt	100%	laptop, candle
26236	puncho	100%	table lamp

5.3 System-aware Attacks on Object Detection

So far, we have tested GhostImage system-aware attacks against image classifiers trained with different datasets in different architectures. Here, we test system-aware attacks against YOLOv3 as a proof-of-concept experiments to demonstrate how an object objector can be fooled to detect an object created by an attacker.

5.3.1 Evaluation Methodology. The attack setup is the same as depicted in Sec. **??**: We use the NEC projector [50] with a replaced lens [5], and the victim camera is Aptina MT9M034 [53] that is 1 m away. We use a TensorFlow implementation [88] of the YOLOv3 architecture with the official network parameters [61], which was pre-trained on the COCO dataset [34] with 80 classes.

We iterate the attack resolution from 8×8 [5] due to the large capacity of YOLOv3. The attack success rates are calculated as the number of images that contain one or more objects with the targeted labels divided by the total number of images. For each label, we manually design the shape of the pattern based on the icons shown in the COCO dataset website [34] as they represent the typical shapes of differing classes. Since the camera outputs images in Resolution 2592×1944 [6] while YOLOv3 accepts 416×416 only, we first crop them to 1944×1944 and then downsample them to 416×416 in order for less distortion compared with direct downsampling.

5.3.2 Results. Table 3 lists the attack success rates of creation attacks against YOLOv3 on traffic-related classes mainly. Firstly, the average success rates on cars is the relatively lower than other traffic-related labels, maybe because in the dataset, the variation of cars is large, meanwhile stop signs and traffic lights are typically the same

[5]The projected resolution
[6]The camera pixel resolution

in most of the training images. In addition, the success rate of fire hydrants is also low because fire hydrants are generally darker than their background while the ghost patterns are brighter than their background.

Secondly, traffic lights yield the highest success rates among other traffic-related labels, probably because the texture of ghost patterns is similar to how a (led array-like) light source would look like from a camera. After all, the ghosts are induced by a projector that is also one type of light sources. To verify this, we also induce TV monitors, yet another type of light sources; results show that it is indeed as successful as the traffic lights from the experimental evaluation, but the emulated success rate is slightly lower because the emulated patterns do not contain the light-source-style texture.

Table 3. Success rates of creation attacks against YOLOv3 in Resolutions 8×8, 16×16, and 32×32 at one meter.

Label	Emulated			Experimental		
	8	16	32	8	16	32
stop sign	100%	100%	100%	82%	53%	13%
traffic light	100%	100%	100%	86%	50%	38%
fire hydrant	100%	100%	100%	33%	13%	13%
car	100%	100%	100%	87%	38%	13%
tv monitor	90%	100%	100%	88%	63%	50%

Thirdly, different than the results from image classification, the attacks do not benefit from high projected resolutions. Instead, as the resolution goes higher, the success rate becomes lower. This is because in the full-resolution images (1944×1944), a typical ghost pattern only occupies 320×320; when downsampling a 1944×1944 image to 416×416, the ghost pattern becomes only 68×68 with also noticeable blur effect. As a result, the projection pixels cannot be clearly seen in the downsampled images. To increase the attack success rate in this case, the attacker may choose to mount the attack at relatively large distances (which is actually stealthier), or be opportunistic for a camera system with large ghosts.

6 DISCUSSION ON ATTACK PRACTICALITY

We discuss the practicality of GhostImage attacks.

Target aiming and tracking: The overlap of ghosts and objects of interest in images must be nearly complete for the attacks to succeed. In the cases of a moving camera (e.g., one mounted to a vehicle), the attacker needs to be able to accurately track the movement of the targeted camera, otherwise the attacker can only sporadically inject ghosts. In fact, recent works on remote sensor attacks [7, 69, 73, 84] generally assume that the attacker is able to aim and/or track stationary/moving targets (so that the researchers could focus their effort on the sensor attack itself), e.g., the AdvLiDAR attack [7] assumes the attacker can achieve this via camera-based object detection and tracking. Nevertheless, there are existing works [46, 78] that have demonstrated the feasibility of tracking cameras and then neutralizing them. This paper's main goal is to propose a new category of camera attacks, which enables an attacker to inject arbitrary patterns, thus we leave the target aiming and tracking as future work.

Attack distances and intensity: Based on our model (Eq. **??**) and experiments (Tab. **??**), the illuminance on the camera from the projector would better be 4/3 of the part from ambient illuminance (to achieve a success rate of 100%). Since Illuminance $\propto$ Luminance $\cdot\, r_{\text{throw}}^2/d^2$, in order to carry out an attack during sunny days (typically with Illuminance 40×10^3 lx), a typical projector (e.g., [16] with Luminance 9×10^3 lm) should work with a telephoto lens [55] (with a throwing ratio 100) at a distance of one meter. For longer distances or brighter backgrounds, one can either acquire a more powerful projector (e.g., [4] with 75×10^3 lm), or combine multiple lenses to achieve much larger throwing ratios (e.g., two Optela lenses [55] yield 200, etc.), or both.

Conspicuousness: The light bursts around the light source in Figures 1 and **??** may raise stealthiness concerns about our attacks. However, according to our analysis in Sec. **??**, such bursts can actually be eliminated because the light source can be outside of view [24]. Even the light source has to be in the frame (due to the lens configuration), we argue that a camera-based object classification system used in autonomous systems generally make decisions without human input (for example, in a Waymo self-driving taxi [80], no human driver is required). Additionally,

the attack beam is so concentrated that only the victim camera can observe it while other human-beings (e.g., pedestrians) cannot (Fig. 10). Finally, the light source only needs to be on for a short amount of time (in the scale of milliseconds), as a few tampered frames can cause incorrect recognitions, decisions and potentially actions [49].

Knowledge of the targeted system: We assume that both types of attackers know about the camera matrix M_c and color calibration matrix H_c. We note that the attacks can still be *effective* without such knowledge but with it the attacks can be more *efficient*. For example, the attacker may choose to lower their attack success expectation but the probability of successful attack may still be too high for potential victims to bear (e.g., a success rate of only 10% might be unacceptable for reasons of safety in automated vehicles). This challenge can be largely eliminated if the attacker is able to purchase a camera of the same, or similar, model as used in the targeted system and use it to derive the matrices. Although the duplicate camera may not be exactly the same as the target one, the channel model would still be in the same form with approximate, probably fine-tuned parameters (via retraining), thanks to the generality of our channel model. Lastly, assuming white-box knowledge on sensors is widely adopted and accepted in the literature [7, 69, 73, 84]. Also, we assume white-box attacks on the neural network, though this assumption can be eliminated by leveraging the transferability of adversarial examples [56].

Attack Variations: Instead of flare effects, we can also leverage beamsplitting to merge the benign image and the adversarial one together. Rather than projectors, lasers can also be used. See our technical report [43] for more details.

7 MODEL-BASED GHOSTIMAGE ATTACK DETECTION

There are potentially several ways to defend against GhostImage attacks. One may leverage hardware defenses such as lens hoods (which however reduces the field of view) or liquid lenses (which is not commonly available yet). One may also improve the robustness of ML algorithms, although most existing approaches focus on norm-based perturbation [40, 58] which do not apply to GhostImage. While sensor fusion may be useful, recent work has shown the vulnerabilty of sensor fusion algorithms [69]. Finally, we tried to detect ghost/flare areas from an image using edge detction and shape recognition, but the detector didn't generalize well due to the transparency of the ghosts. The reader is refered to [42] for detailed discussion of potential defenses.

In this section, we propose a model-based algorithm that detects ghost via modeling the ghost locations, and eliminate false positives via modeling the movement of the natural ghosts. We first define the attack model in Sec. 7.1 (which is stronger than the one in Sec. 3), before introducing the general attack detection strategy (Sec. 7.2). We discuss how to detect attack existence, and eliminate false positives in Sec. 7.3 and Sec. 7.4, respectively. We present our security analysis considering measurement errors in Sec. 7.5, and evaluation results in Sec. 7.6. Sec. 7.7 presents discussion.

7.1 Attack Model

In an attack against the camera perception of an autonomous system, the attacker uses GhostImage techniques to spoof a spurious/incorrect object into the system, with the goal of controlling its actuation. For example, a self-driving car could be made to decelerate because it recognizes a non-existent stop sign created by the attacker. Depending on the actual design of an autonomous system, which includes the perception, planning, and control module, the adversarial object generally needs to be induced consistently for a period of time in order for the attacker to successfully induce undesired actuation.

In terms of image frames, let us assume that the attacker needs to consistently inject an adversarial object for at least k frames where $k \geq 2$ for the victim system to react as the attacker intends. For creation attacks, the attack is regarded as successful when the injected ghost has been recognized as the target class for at least

k consecutive frames. For alteration attacks, an attack is regarded as successful when the injected ghost has overlapped with the target object in the image and altered the classification result, to the targeted class, for at least k consecutive frames.

7.2 Overview of Detection Algorithm

Our detection algorithm reduces the problem of attack detection to verifying whether there is a ghost overlapping with an object (that is detected by an object detector): If so, the algorithm raises an alarm. This is based on the fact that if an attack has, or is about to, succeed, there must be at least one frame that contains such an overlap (Sec. 7.3). Note that the latter is only a necessary condition of the former, but not a sufficient one. For instance, a natural light source can produce a ghost that accidentally overlaps with the object, i.e., a false positive.

Algorithm 1: GhostImage Attack Detection

```
Function IsClose(F):
    s ← FindLightSource(F)
    G ← CalculateGhosts(s)
    B ← ObjectDetection(F)
    for G ∈ G do
        for B ∈ B do
            if dist. between G and B is less than T then
                return TRUE
    return FALSE
while a new frame F(t') comes do
    F(t) ← The frame captured Δt ago
    if IsClose (F(t)) and IsClose (F(t')) then
        Raise an alarm
```

In order to eliminate such false positives, we consider two frames taken from a moving camera: If both frames contain ghost-object overlaps, the algorithm can confidently claim an attack (Sec. 7.4). We would like to underline that only when considering false positives do we require two frames from a moving camera. If false positives for some objects are acceptable to the system designers (out of, say, an abundance of caution to always operate safely), one frame is sufficient and, additionally, the camera can be stationary.

The benefit of our detection approach is threefold. First, it allows for attack prevention since two frames are sufficient for detection; a decision can be made before the attack succeeds if $k > 2$ (Sec. 7.1). Second, it detects failed attacks, too, such as an attack attempt that had manipulated only k' frames where $2 < k' < k$, or an attack where the ghost was too weak to alter the object class. Third, it requires only the result from camera-based perception; it does not rely on the information from other sensors, nor involve other modules such as the planning or control, which minimizes the attack surface.

7.3 Ghost Object Detection

Our detection algorithm detects possible GhostImage attacks by the fact that all successful attacks must result in a detected object that overlaps or is contained in a ghost (a necessary condition). We find that in order to detect GhostImage attacks, the defender must detect the attack mechanism; viz., the introduction of objects through the lens flare effect, which indubitably introduces ghosts. Recall that the attacker aims to either create an image of a fake object (creation attacks), or alter an image of a real object into something else (alteration attacks). Either way, the compromised image must contain at least one (adversarial) object that is able to be detected by the object detection algorithm being used (otherwise the attack would not be considered successful, so there is nothing to defend against). Therefore, the defender only needs to pay attention to those detected objects.

Among those detected objects, the defender now needs to distinguish adversarial objects from legitimate ones. To achieve this, the defender can detect all ghosts and then check whether there is a ghost overlapping with an

object: If so, then this object is considered adversarial. To detect ghosts, we assume that the defender possesses the configuration of the lens system (white-box knowledge), which can be used to derive the pixel coordinates of all possible ghosts given the coordinates of the light source [22]. In this way, the defender avoids recognizing ghosts directly using existing CV algorithms or neural nets (which themselves are vulnerable to adversarial attacks [8]), but rather recognizes light sources, which is easier and more robust [64].

In fact, even in the case where such knowledge is not available, the defender can always follow the attacker's end-to-end, black-box-style analysis (Sec. **??**) to estimate the pixel coordinates of ghosts. In any case, we may assume that the attacker is in possession of white-box knowledge of the lens system, as it does not affect the performance of our defense algorithm because *it is the physical law (instead of a secret) that reveals the attack.*

Our detection algorithm is summarized in Alg. 1. The defender uses a light source detection algorithm [64] (Function FindLightSource(F) where F is the image) to find the coordinates of the light source s, then the defender calculates a set of (center) coordinates of potential ghosts $\mathcal{G}$ (Function CalculateGhosts(s)). The object detection algorithm (e.g., YOLO) returns a set of (center) coordinates of detected objects $\mathcal{B}$ (Function ObjectDetection(F)). With $\mathcal{G}$ and $\mathcal{B}$, the defender raises an alarm if and only if there exists a pair of $G \in \mathcal{G}$ and $B \in \mathcal{B}$ whose Euclidean distance is less than a threshold $\mathcal{T}$.

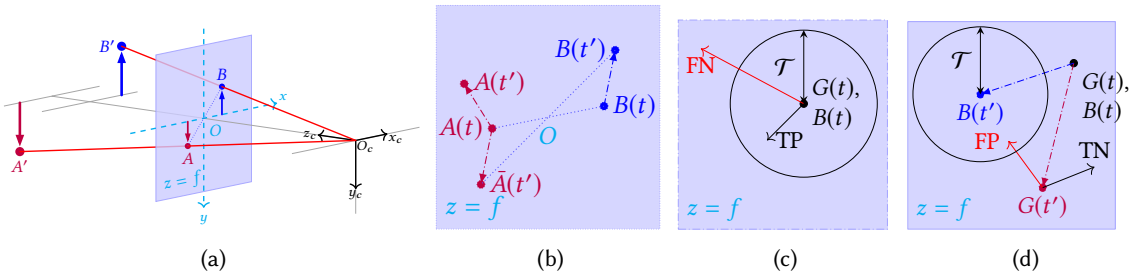
7.4 Elimination of False Positives

Unfortunately, using the above strategy alone may raise false positives. For example, a natural light source, such as the sun or streetlights, may induce a ghost that coincidentally overlaps with an object. To eliminate such false positives, we can utilize the principle of spatiotemporal consistency: Only when two frames from a moving camera yield an overlap between ghosts and objects would the detector raise an alarm (Alg. 1). A moving camera captures the dynamic of the scene which enables us to verify the spatiotemporal consistency that an attack would break. This is a modeling-based approach (modeling the movement of natural ghosts) that can be regarded as a consistency check on the continuous overlap of a ghost with a detected object.

The assumption of a moving camera is reasonable for the systems under consideration and doesn't produce a burden on their designers. For example, when autonomous vehicles are in motion so are their cameras. When vehicles are parked or otherwise stationary, we would argue that the consequence of an attack is negligible, therefore the importance of defending against an attack in such cases is low. Similarly, the cameras in surveillance systems [17, 81] are usually able to pan and/or tile themselves (e.g., to track an object). They can either keep moving, or move themselves only when necessary (e.g., when detect a ghost-object overlay). Last but not least, we note that only when we need to eliminate this type of false positive do we require a moving camera. If the system could tolerate a slightly higher false positive rate, such an assumption might not need to be made as such false positive cases are rare.

In the rest of this section, we show that *it is impossible for a stationary, natural light source to cause a ghost that overlaps with the image of a stationary, actual object for more than one frame taken by a moving camera, when the light source and the object are at different 3-D locations.* We analyze a forward-moving vehicle to show how spatiotemporal consistency can be utilized on a moving camera to eliminate false positives caused by natural ghosts. In particular, we show that a system of three equations will eventually reveal the real-world location (3-D) of a light source, and that an object must be at least very close, due to measurement error (Sec. 7.5), to this location in order for the ghost to overlap with the object image consistently in more than one frame. As a natural light source is unlikely to produce two or more frames all with ghost-object overlaps, we can confidently state that a GhostImage attack is the cause of the apparent object.

Specifically, let us define the real-world coordinates of the natural light source and the natural object at time t as $A'(t) = (x_{A'(t)}, y_{A'(t)}, z_{A'(t)})^\top$ and $B'(t) = (x_{B'(t)}, y_{B'(t)}, z_{b'(t)})^\top$ respectively (Fig. 11a). We assume both are stationary in the real-world. The corresponding pixel coordinates of them in the camera-captured image are



(a) (b) (c) (d)

Fig. 11. **(a)** The world 3-D coordinate system is defined by (x_c, y_c, z_c) and is centered at $O_c = (0, 0, 0)$. The camera (at O_c) with focal length of f is facing towards $(0, 0, 1)$. The plane $z = f$ is the image/frame plane, which may be seen as the CMOS image sensor (the actual CMOS sensor is at $z = -f$). The origin of the image 2-D coordinate system is at its center, which is defined by (x, y). The object (e.g., a STOP sign) locates at B' in the world (3-D), and is projected to B on the image (2-D). Similarly, the light source locates at A' and A. The line AB crosses O. **(b)** The frame plane at two timestamps. At time t, the images of the light source and the object was at $A(t)$ and $B(t)$ respectively. The ghost caused by $A(t)$ was also at $B(t)$ (i.e., an object-ghost overlap). Due to the movement of the vehicle (towards $(0, 0, 1)$), the object moved to $B(t')$. In order to have object-ghost overlap again in the second frame, the light source of an attacker should move to $\bar{A}(t')$, but a natural stationary light source would move to $A(t')$ instead, because of the geometry caused by the vehicle's movement. **(c)** shows TP and FN in a single time step. **(d)** shows two time steps since FP detection utilizes spatiotemporal consistency across time steps.

denoted as $A(t) = (x_{A(t)}, y_{A(t)})^\top$ and $B(t) = (x_{B(t)}, y_{B(t)})^\top$ respectively. The image center is at O. Given the camera matrix M_c, at time t, we can derive the pixel coordinates of $A(t)$ based on its real-world coordinates $A'(t)$ by the well-known homogeneous transformation (Eq. **??**). Similarly, we can derive $B(t)$, as well as $A(t')$ and $B(t')$ at time t' where $t' > t$ (Fig. 11b). We define $\Delta t = t' - t$ (Alg. 1), which will be discussed in Sec. 7.7.

To simplify the analysis, we assume that the camera orientation is $(0, 0, 1)^\top$ for all time. In addition, because the camera is always the center of the coordinate system (Fig. 11a), as the vehicle is moving forward in the real-world, the object and the light source are both moving backward by the same distance. Let us define the camara displacement as $D = (\Delta x, \Delta y, \Delta z)^\top$ where $\Delta y = 0$ (altitude) for simplicity. With that, we have

$$A'(t') = A'(t) + D, \quad B'(t') = B'(t) + D. \tag{5}$$

For a false positive to occur the ghost G caused by the light source A overlaps with the image of the object B, i.e., $B = G$ (otherwise it is not a positive detection). At time t, the image of the light source $A(t) = (x_{A(t)}, y_{A(t)})^\top$, the image of the object $B(t) = (x_{B(t)} y_{B(t)})^\top$ and the frame center $O = (0, 0)^\top$ must be on the same straight line (Fig. 11b). The same principle applies to time t' as well. Thus, we have our first two equations in (6).

$$\frac{x_{A(t)}}{y_{A(t)}} = \frac{x_{B(t)}}{y_{B(t)}}, \quad \frac{x_{A(t')}}{y_{A(t')}} = \frac{x_{B(t')}}{y_{B(t')}}, \quad \frac{x_{A(t')}}{x_{B(t')}} = \frac{x_{A(t)}}{x_{B(t)}}. \tag{6}$$

In addition, recall that the source-ghost ratio $r = \overline{OB}/\overline{OA}$ must be identical at any time (Sec. **??**). Thus, for the case of two frames we arrive at the third equation in (6).

From a high level, Eqs. 6 capture the optical principle of the lens flare effects and overlaps, while Eqs. 5 capture the moving camera assumption. The homogeneous transformation captures the optical imaging principle. Together these produce the conditions under which spatiotemporal consistency exists. That is, when we put them together we arrive at $A'(t) = B'(t)$, which means the light source has to be placed at the same real-world location as the object, or at least very close. The chance that there is a naturally occurring source-ghost pair with a ratio of one is rare, let alone the ghost would be overwhelmed by the image of the light source (burst effect) in this case .

In conclusion, a natural light source is unlikely to cause two or more frames all with ghost-object overlaps. Therefore, if two or more frames indicate such an overlap, the defender can confidently claim a detection of GhostImage attacks with a low chance of false positives.

7.5 Security Analysis

Because there will be errors in determining the pixel coordinates of objects, light sources, ghosts, etc., we evaluate the True Positive Rate (TPR) and False Positive Rate (FPR) under different thresholds $\mathcal{T}$. We again use the forward-moving vehicle scenario to study the impact of different camera displacements, and different real-world configurations of objects and light sources.

First of all, we estimate the error distribution using the same data that we collected when we moved around the flashlight in front of the camera (See Sec. **??**), for which we run a light source detection algorithm that is based on edge detection and shape recognition [64]. As we do not have detailed information on the lens configuration, we estimate the ghost locations using the source-ghost ratio (Eq. **??**) approximated in Sec. **??** for our setup. Results show that the error follows a bivariate Gaussian distribution, i.e., $\epsilon \sim \mathcal{N}(\boldsymbol{\mu}_\epsilon, \Sigma_\epsilon)$, where

$$\boldsymbol{\mu}_\epsilon = \begin{bmatrix} \mu_\epsilon \\ \mu_\epsilon \end{bmatrix}, \quad \Sigma_\epsilon = \begin{bmatrix} \sigma_\epsilon^2 & 0 \\ 0 & \sigma_\epsilon^2 \end{bmatrix}.$$

For our experimental setup, we have $\mu_\epsilon = 0.014$, $\sigma_\epsilon^2 = 0.004$ with the image size normalized to one.

For true positives (Fig. 11c), we first derive the TPR r_{TP} for a single frame; the TPR for two frames, assuming the error distribution is i.i.d. across frames, is then r_{TP}^2. For simplicity, we analyze the worst case scenario where the attacker is able to track the camera perfectly, which means the center of the adversarial ghost G is always at the center of the object image B, i.e., $B = G$. In our analysis we note that the measured pixel location of ghosts may be incorrect (Fig. 11c). On one hand, the estimated ghost may fall out of the circle $G(t)$ with Radius $\mathcal{T}$, causing a false positive (the red arrow); on the other hand, the error may be small and the ghost is still within the circle, thus a true positive. The derivation of r_{TP} is the integral of the probability density function (PDF) of the error ϵ over the circle that centers at $B(t)$ with Radius $\mathcal{T}$ (Fig. 11c). For TP, the integral is a special case of the offset circle probability [15] with zero offset.

The worst case scenario of FPRs is different: A natural ghost $G(t)$ and an object $B(t)$ located at the same pixel coordinates in the first frame and in the next frame they move to $G(t')$ and $B(t')$, respectively, due to the displacement of the camera (Fig. 11d). A larger error may result in a false positive (falling into Circle $B(t')$), while a smaller error may produce a true negative. The calculation of FPR is different than TPR, as the TPR's derivation is based solely on the error distribution because the attacker's capability of tracking the camera compensates for the impact of the 3-D location of the adversarial light source A', and the camera displacement D. For the worst-case FPR, we need to consider different combinations of the natural light source locations and camera displacements because their variation produces different distances between $G(t')$ and $B(t')$, denoted as $d_{GB} = \overline{G(t')B(t')}$, that further produces differing FPRs. We use offset circle probability [15] to calculate the FPR r_{FP}, which is the circular integral of the error's PDF, where the offset circle centers at $B(t')$ with Radius $\mathcal{T}$ (Fig. 11d). For FP, the error is zero at $G(t')$ hence the offset is d_{GB}.

7.6 Evaluation

Setup: We ran a numerical evaluation with settings described as follows. For simplicity, we assume a pinhole camera (Fig. 11a) with a camera matrix

$$M_c = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

where $f = 30\,\text{mm}$ is the focal length, and $p_x = p_y = 0$ which means the camera origin is at the image origin. The size of the CMOS sensor is $7\,\text{mm} \times 6\,\text{mm}$.

At time t, let us suppose the object is at $B' = (1\,\text{m}, 2\,\text{m}, 20\,\text{m})^\top$ and we iterate the natural light source's real-world location A' around B' to show the impact of configuration on detection performance. Because we are interested in the worst-case FPR, for each instance of A', we assign the source-ghost ratio r with a value that would result in perfect ghost-object overlays at the first frame, i.e., $r = \overline{OB}(t)/\overline{OA}(t)$. We estimated the light source detection error from an image dataset collected from our experiment setup (Sec. 7.5).

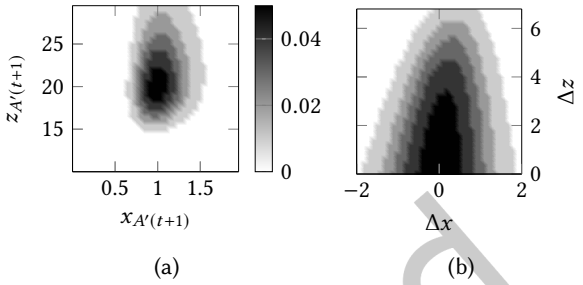


Fig. 12. Equal error rate (EER) distribution under (a) varying real-world locations of the natural light source, with a fixed camera displacement $(0.75, 0, 3)^\top$, and (b) varying displacements of the camera, with a fixed natural light source location $A'(t) = (1.25, 2.5, 22)^\top$. Both figures share the same color bar. For both figures, the object is at $B'(t) = (1, 2, 20)^\top$, and the unit of all axes is meter. The highest EER is 0.05.

Results: We present the distribution of equal error rates (EERs, which considers both false positive rate and false negative rate) in terms of different real-world locations of the natural light source in Fig. 12a. There are only two degrees of freedom, $x_{A'(t)}$ and $z_{A'(t)}$, that decide the 3-D location of the natural light source because the it needs to be on the plane, OBB' (Fig. 11a), otherwise there would not be a ghost-object overlap in the first frame. We take the object to be located at the center of the plot, i.e., $B'(t)$. We set $\Delta z = 3\,\text{m}$ and $\Delta x = 0.75\,\text{m}$. We iterate $x_{A'(t)}$ from 0 m to 2 m, and $z_{A'(t)}$ from 10 m to 30 m. We see that a higher EER occurs when A' is close to B', but it decreases to zeros as A' is placed away from B'. This supports the theoretical conclusion made in Sec. 7.4 that to have ghost-object overlaps in two frames, the natural light source needs to be placed at or close to the object in the real world.

To illustrate the impact of the camera movement, we test different camera displacements by varying Δx and Δz (Fig. 12b). Note that varying Δz is essentially changing the vehicle speed. The natural light source is at $A'(t) = (1.25, 2.5, 22)^\top$. We make three observations: First, larger Δz (e.g., higher car speeds, or lower sampling rates) yields a lower (better) EER. This is because when the vehicle is moving faster (i.e., Δz is larger), the ghost is further away from the object image in the second frame (i.e., d_{GB} is larger). Therefore the integral over Circle $B(t')$ of the error distribution becomes smaller (Fig. 11d), which results in a lower false positive rate. Meanwhile the true positive rate remains constant because it depends only on an error distribution that is i.i.d. Second, driving slightly towards the left or the right results in a lower EER because such movements change the real-world relative, geometrical configuration (of the camera, the light source, and the object) more significantly than moving only forward. Third, movement towards to the left yields a lower EER than to the right because of the configuration as well. To explain this we note that if $A'(t) = (1.25, 2.5, 18)^\top$ or $A'(t) = (0.9, 1.8, 22)^\top$ (i.e., the natural light source is closer to the camera than the object along either the x or z axis), the inclination of EER will be on the other side.

7.6.1 Moving Natural Light Sources. Here, we relax the assumption of stationary, natural light sources and use a real-world dataset to test whether a moving natural light source such as the head light of an incoming vehicle can produce false positives.

Setup: From the BDD100K dataset [86], we randomly select the 100 traces of the headlights and 100 traces of traffic signs and the average length of these traces is three seconds at 5 Hz (15 frames per trace). For each pair of a headlight trace and a sign trace, we aim to test the ghost created by the headlight is close enough (determined by the threshold $\mathcal{T}$) to the sign for two or more consecutive frames (which is our detection criterion, otherwise it is not a false positive). Because they may start to appear at different time, we shift one trace by various steps.

Once a combination of two traces, $G(t), 1 \leq t \leq M$ and $B(t), 1 \leq t \leq N$, and the shift step k has been decided, we compute the Euclidean distances between $G(t)$ and $B(t+k)$, and between $G(t+1)$ and $B(t+k+1)$: If both distances are less than $\mathcal{T}$, we mark this combination as a false positive. Therefore, given a threshold $\mathcal{T}$, we can calculate the false positive rate. The TPR calculation is the same as in Section 7.5.

Results: We plot the ROC curve in Figure 13. We can see that the moving headlights rarely cause false positive rates. The reason is that the dynamics of the ghosts created by the headlight of an incoming vehicle is significantly different than the dynamics of a traffic sign. They may overlap for one frame, but continuous overlapping for more than two frames is rare as their directions are different. See Figure 11b for an illustration.

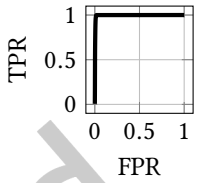


Fig. 13. The ROC curve of moving headlights.

7.7 Discussion

The results in Figs. 12a and 12b are based on a simple light source detection algorithm [64] and an approximated source-ghost model (Sec. **??**); if the defender uses a more sophisticated light source detection algorithm with higher accuracy [36], and/or a more precise source-ghost model derived from white-box lens configuration, an even lower EER can be achieved.

Recall that $\Delta t = t' - t$ is the time duration between two moments when the attack detection algorithm samples two images. The higher the sampling rate ($1/\Delta t$), the quicker a detection decision can be made; therefore, high sampling rates may seem preferable. However, when the vehicle is slow a high sample rate results in a small camera displacement, which means a higher EER. As such, the sampling rate should be proportional to the speed of the vehicle (i.e., $\Delta t \propto 1/v$) because a high sampling rates at high vehicle speeds still gives large enough camera displacements (thus lower EERs), and more importantly, it is more critical to detect attacks quickly in high speed scenarios for applications such as autonomous vehicles. Baidu Apollo's sampling rate is 10 Hz [3], in which case our algorithm needs only 0.2 second to detect an attack.

An attacker may combine GhostImage attacks with adversarial patches against object detection [65] where the label of the target object can be altered by adding a patch that does not need to overlay with the object. Such attacks can be detected and mitigated, by removing the patch, and then checking whether the object label is changed or not: If so, then the patch is considered adversarial, and the ground-truth label is the one after removal. The defender can locate the patch because it was conveyed via ghosts whose locations can be estimated (See Sec. 7.3).

8 RELATED WORK

Here, we review both sensor attacks and adversarial examples, as well as their countermeasures.

8.1 Sensor attacks

Perception in autonomous and surveillance systems occurs through sensors, which convert analog signals into digital ones that are further analyzed by computing systems. Recent work has demonstrated that the sensing mechanism itself is vulnerable to attack and that such attacks may be used to bypass digital protections [19, 83]. For example, microphones have been subject to inaudible voice, light, and electromagnetic (EM) based attacks [28, 73, 89], and light sensors can be influenced via EM interference to report lighter or darker conditions [67].

Existing remote attacks against cameras [59, 78, 84] are denial-of-service attacks and do not seek to compromise object recognition as our GhostImage attacks do. Those attacks that do target object recognition [11, 18, 92] are either digital or physical domain attacks (i.e., they need to modify the object of interest, in this case a traffic sign or road pavement, physically or after the object has been captured by a camera) rather than perception domain attacks [19, 83]. Similarly, several light-based attacks [37, 49, 51, 66, 94] fall within the domain of physical

attacks, as opposed to our perception domain attack, because these approaches indirectly attack the cameras by illuminating the object of interest or the environment with visible or infrared light, and they are less robust to the changes in color and texture of the object and its background. Although both works exploit rolling shuttle effect, Sayles et al [66] propose to shine light on objects hence it is perceptional, meanwhile Kohler's et al [26] direct laser to the camera, thus being a physical attack. We did not consider infrared noise in our attacks as it can be easily eliminated from visible light systems using infrared filters. Li et al.'s attacks on cameras require attackers to place stickers on lenses, to which is generally hard to get access [32]. Ji et al. send acoustic signals to compromise a camera's stabilizer which introduces only texture changes [23]. Attacks on LiDAR systems [7, 59, 70] are also related, but they are considerably easier to carry out than our visible light-based attacks against cameras because attackers can directly inject adversarial laser pulses into LiDARs without worrying about blocking the object of interest from the sensor. The most similar attack is DoubleStar attacks [93] which also exploit ghost effects, but they target depth estimation rather than object recognition, which make it not suitable for baseline comparison. On the other hand, our detection algorithm would be able to detect DoubleStar attacks since they also rely on ghost effects, similar to how our algorithm is able to detect GhostImage attacks.

Defenses against sensor attacks are mostly specific to the sensor and/or the attack [7, 28, 59, 70, 73, 74, 91], therefore cannot be directly adopted in our case. A recent work [74] proposes to detect LiDAR attacks using the occluding relationship among objects, which is also based on spatial consistency, but they do not consider temporal aspects as we do. Sensor fusion algorithms can generally be used as a defense [35] but it has been shown vulnerable to attackers who are able to compromise multiple sensors simultaneously [6, 48], or even one sensor (at a time) [39, 48, 69]. For example, Zhang et al. [90] propose a fusion-based detection algorithm that verifies the depth estimation from three cameras, assuming the attacker can induce noise at only random locations, which makes their defense vulnerable to GhostImage attacks that can induce noise at attacker-chosen locations. See [19, 83] for a review of analog sensor attacks and defenses.

8.2 Adversarial Perturbation and Defenses

State-of-the-art adversarial examples can be categorized as digital (e.g., [8, 76]), or physical domain attacks (e.g., [18, 29, 68]) in which objects of interest are physically modified to cause misclassification. The latter differs from GhostImage attacks in that we target the sensor (camera) without needing to physically modify any real-world object. Another line of work focuses on unrestricted adversarial examples (so as ours), such as [72], though they are limited in the digital domain. In terms of defending neural networks from adversarial examples, be they physical or digital, schemes include modifying the network to be more robust [40, 58], while other defenses have focused on either detecting adversarial inputs [38] or transforming them into benign images [45], most of which are under the general assumption of bounded perturbations, hence are inapplicable to our attacks [65]; while others could also be bypassed by being taken as constraints in the optimization formulation. Our attack detection scheme does not rely on the digital characteristics of adversarial examples, but on the physical characteristics of adversarial ghosts, therefore it is robust to optimization-based adversarial attacks.

Consistency-based defenses have recently become the focus as countermeasures against adversarial attacks. While AdvIT [82] leverages optical flow for spatio-temporal consistency check, it cannot detect GhostImage attacks because ghosts are similar to adversarial patches that can maintain the consistency. Gurel et al. [21] proposed to the consistency of an object's attributes using prior knowledge such as a STOP sign is mainly red and in an octagon shape, however their approach cannot apply to the temporal domain which is continuous and high-dimensional with more uncertainty. The co-existence among multiple objects in a scene can reveal the spurious object [33, 85]; however it does require a scene with multiple objects. Moving target defenses have been shown effective against adversarial examples, though with a norm-bounded assumption [2] as well. PercepGuard

[44] uses an LSTM to classify a sequence of bounding boxes into an object class, which is then cross-checked with the object classification result for misclassification detection.

Different from the aforementioned defenses that are data-driven which requires a large amount of data to train ML models, our detection algorithm follows the model-based approach which leverages the domain knowledge of the problem for consistency checks. Unlike data-driven approaches where FPs/FNs can occur due to corner cases (induced by incomplete training data and uncertainty in the data distribution), model-based approaches leverage domain knowledge and are easily explainable when FP occurs, which enables white-box testing. However, model-based approaches are usually more specific to the problem [60]. For example, while the idea of using spatio-temporal consistency can be applied to a wide range of problems, the instance of this idea proposed in this paper is more suitable for ghost-based camera attacks [43, 93] since it relies on the physical characteristics of camera ghost effects. Nevertheless, model-based and data-driven approaches are complimentary to each other to improve the robustness of a cyber-physical system.

9 CONCLUSION

In this work we presented GhostImage attacks against camera-based object recognition systems, and its countermeasures. Using common optical effects, viz. lens flare/ghost effects, an attacker is able to inject arbitrary adversarial patterns into camera images using a projector. To increase the efficacy of the attack, we proposed a projector-camera channel model, and leverage adversarial machine learning for optimal attack patterns. We evaluated the effectiveness and robustness of GhostImage attacks via experiments; results show attack success rates as high as 100% depending on the attack distance, demonstrating potential impact on autonomous systems, such as self-driving cars and surveillance systems. Lastly, we developed an attack detection algorithm that yields worst-case EERs as 5%. The algorithm depends on whether there are ghost-object overlaps in images to detect the attacks, meanwhile eliminates false positives via model-based consistency check.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation (NSF Grant CNS-1801611) and Army Research Office (ARO Grant W911NF-21-1-0320).

REFERENCES

[1] Amazon. 2021. Prime Air Delivery.
[2] Abderrahmen Amich and Birhanu Eshete. 2021. Morphence: Moving Target Defense Against Adversarial Examples. In *Annual Comp. Sec. Applications Conf.*
[3] Baidu. 2020. Apollo. https://github.com/ApolloAuto/apollo.
[4] Barco. 2021. XDL-4K75.
[5] Canon. 2021. Telephoto Zoom EF-S 55-250mm.
[6] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. Chen, M. Liu, and B. Li. 2021. Invisible for both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks. In *2021 2021 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 1302–1320. https://doi.org/10.1109/SP40001.2021.00076
[7] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. 2019. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2267–2281.
[8] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 39–57.
[9] Hervé Chabanne, Jean-Luc Danger, Linda Guiga, and Ulrich Kühne. 2021. Side channel attacks for architecture extraction of neural networks. *CAAI Transactions on Intelligence Technology* 6, 1 (2021), 3–16.
[10] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, et al. 2011. Comprehensive experimental analyses of automotive attack surfaces.. In *USENIX Security Symposium*, Vol. 4. San Francisco, 447–462.

[11] Alesia Chernikova, Alina Oprea, Cristina Nita-Rotaru, and BaekGyu Kim. 2019. Are Self-Driving Cars Secure? Evasion Attacks against Deep Neural Networks for Steering Angle Prediction. In *IEEE SP Workshop on IoT*. IEEE.
[12] commaai. 2021. openpilot. https://github.com/commaai/openpilot.
[13] Andrei Costin, Jonas Zaddach, Aurélien Francillon, and Davide Balzarotti. 2014. A large-scale analysis of the security of embedded firmwares. In *23rd USENIX Security Symposium (USENIX Security 14)*. 95–110.
[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 248–255.
[15] A. R. DiDonato and M. P. Jarnagin. 1961. Integration of the general bivariate Gaussian distribution over an offset circle. *Math. Comp.* 15 (1961), 375–382.
[16] Epson. 2021. Pro L1490U WUXGA 3LCD Laser Projector.
[17] eufy. 2021. Indoor Cam 2K Pan and Tilt.
[18] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1625–1634.
[19] Ilias Giechaskiel and Kasper Bonne Rasmussen. 2020. Taxonomy and Challenges of Out-of-Band Signal Injection Attacks and Defenses. *IEEE Communication Surveys & Tutorials* (2020).
[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[21] Nezihe Merve Gürel, Xiangyu Qi, Luka Rimanic, Ce Zhang, and Bo Li. 2021. Knowledge Enhanced Machine Learning Pipeline against Diverse Adversarial Attacks. In *International Conference on Machine Learning*.
[22] Matthias B. Hullin, Elmar Eisemann, Hans-Peter Seidel, and Sungkil Lee. 2011. Physically-Based Real-Time Lens Flare Rendering. *ACM Trans. Graph. (Proc. SIGGRAPH 2011)* 30, 4 (2011), 108:1–108:9.
[23] X. Ji, Y. Cheng, Y. Zhang, K. Wang, C. Yan, W. Xu, and K. Fu. 2021. Poltergeist: Acoustic Adversarial Machine Learning against Cameras and Computer Vision. In *2021 2021 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 1573–1588. https://doi.org/10.1109/SP40001.2021.00091
[24] Gunawan Kartapranata. 2010. *Lens Flare at Borobudur Stairs Kala Arches*.
[25] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
[26] Sebastian Köhler, Giulio Lovisotto, Simon Birnbach, Richard Baker, and Ivan Martinovic. 2021. They See Me Rollin': Inherent Vulnerability of the Rolling Shutter in CMOS Image Sensors. *arXiv preprint arXiv:2101.10011* (2021).
[27] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.
[28] Denis Foo Kune, John Backes, Shane S Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. 2013. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *2013 IEEE Symposium on Security and Privacy*. IEEE, 145–159.
[29] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial Examples in the Physical World. In *Artificial Intelligence Safety and Security*. Chapman and Hall/CRC, 99–112.
[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* (2015).
[31] Hsien-Che Lee. 2005. *Introduction to color imaging science*. Cambridge University Press.
[32] Juncheng B Li, Frank R Schmidt, and J Zico Kolter. 2019. Adversarial camera stickers: A physical camera attack on deep learning classifier. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.
[33] Shasha Li, Shitong Zhu, Sudipta Paul, Amit Roy-Chowdhury, Chengyu Song, Srikanth Krishnamurthy, Ananthram Swami, and Kevin S Chan. 2020. Connecting the dots: Detecting adversarial perturbations using context inconsistency. In *European Conference on Computer Vision*. Springer.
[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer.
[35] Jinshan Liu and Jerry Park. 2021. "Seeing is not Always Believing": Detecting Perception Error Attacks Against Autonomous Vehicles. *IEEE Transactions on Dependable and Secure Computing* (2021).
[36] Stephen Lombardi and Ko Nishino. 2015. Reflectance and illumination recovery in the wild. *IEEE transactions on pattern analysis and machine intelligence* 38, 1 (2015), 129–141.
[37] Giulio Lovisotto, Henry Turner, Ivo Sluganovic, Martin Strohmeier, and Ivan Martinovic. 2021. SLAP: Improving Physical Adversarial Examples with Short-Lived Adversarial Perturbations. In *USENIX Security 2021*.
[38] Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan Lee, and Xiangyu Zhang. 2019. NIC: Detecting Adversarial Samples with Neural Network Invariant Checking.. In *Network and Distributed System Security Symposium*.
[39] Yuzhe Ma, Jon Sharp, Ruizhe Wang, Earlence Fernandes, and Xiaojin Zhu. 2021. Sequential Attacks on Kalman Filter-based Forward Collision Warning Systems. In *The AAAI Conference on Artificial Intelligence (AAAI)*.
[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJzIBfZAb

[41] Saurav Maji, Utsav Banerjee, and Anantha P Chandrakasan. 2021. Leaky nets: Recovering embedded neural network models and inputs through simple power and timing side-channels—Attacks and defenses. *IEEE Internet of Things Journal* 8, 15 (2021), 12079–12092.
[42] Yanmao Man, Ming Li, and Ryan Gerdes. 2020. GhostImage: Remote Perception Attacks against Camera-based Image Classification Systems. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. USENIX Association, San Sebastian, 317–332. https://www.usenix.org/conference/raid2020/presentation/man
[43] Yanmao Man, Ming Li, and Ryan Gerdes. 2020. GhostImage: Remote Perception Attacks against Camera-based Image Classification Systems. *arXiv preprint arXiv:2001.07792* (2020).
[44] Yanmao Man, Raymond Muller, Ming Li, Z. Berkay Celik, and Ryan Gerdes. 2023. That Person Moves Like A Car: Misclassification Attack Detection for Autonomous Systems Using Spatiotemporal Consistency. In *USENIX Security*.
[45] Dongyu Meng and Hao Chen. 2017. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 135–147.
[46] KYLE MIZOKAMI. 2019. China Could Blind U.S. Satellites With Lasers. https://www.popularmechanics.com/military/weapons/a29307535/china-satellite-laser-blinding/.
[47] Andreas Mogelmose, Mohan Manubhai Trivedi, and Thomas B Moeslund. 2012. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems* 13, 4 (2012), 1484–1497.
[48] Shoei Nashimoto, Daisuke Suzuki, Takeshi Sugawara, and Kazuo Sakiyama. 2018. Sensor CON-Fusion: Defeating Kalman filter in signal injection attack. In *Asia Conference on Computer and Communications Security*.
[49] Ben Nassi, Yisroel Mirsky, Dudi Nassi, Raz Ben-Netanel, Oleg Drokin, and Yuval Elovici. 2020. Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 293–308.
[50] NEC 2007. *NP Installation Series User's Manual*. NEC.
[51] Luan Nguyen, Sunpreet S. Arora, Yuhang Wu, and Hao Yang. 2020. Adversarial Light Projection Attacks on Face Recognition Systems: A Feasibility Study. arXiv:2003.11145 [cs.CV]
[52] Nvidia 2016. *NVIDIA TESLA P100 GPU ACCELERATOR*. Nvidia.
[53] ON semiconductor 2017. *MT9M034 1/3-Inch CMOS Digital Image Sensor*. ON semiconductor.
[54] ON semiconductor 2017. *MT9V034 1/3-Inch Wide-VGA CMOS Digital Image Sensor*. ON semiconductor.
[55] Opteka. 2019. Opteka 650-1300mm Telephoto Zoom Lens.
[56] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).
[57] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy*.
[58] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*.
[59] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. 2015. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe* 11 (2015), 2015.
[60] Raul Quinonez, Jairo Giraldo, Luis Salazar, Erick Bauman, Alvaro Cardenas, and Zhiqiang Lin. 2020. SAVIOR: Securing autonomous vehicles with robust physical invariants. In *Usenix Security*.
[61] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.
[62] Evan Ribnick, Stefan Atev, Osama Masoud, Nikolaos Papanikolopoulos, and Richard Voyles. 2006. Real-time detection of camera tampering. In *2006 IEEE International Conference on Video and Signal Based Surveillance*. IEEE, 10–10.
[63] Ring. 2019. Indoor Security Cameras. https://shop.ring.com/collections/security-cams#indoor.
[64] Adrian Rosebrock. 2016. Detecting multiple bright spots in an image with Python and OpenCV. https://www.pyimagesearch.com/2016/10/31/detecting-multiple-bright-spots-in-an-image-with-python-and-opencv/.
[65] Aniruddha Saha, Akshayvarun Subramanya, Koninika Patil, and Hamed Pirsiavash. 2020. Role of Spatial Context in Adversarial Robustness for Object Detection. In *CVPR Workshop on Adversarial Machine Learning in Computer Vision*.
[66] Athena Sayles, Ashish Hooda, Mohit Gupta, Rahul Chatterjee, and Earlence Fernandes. 2021. Invisible Perturbations: Physical Adversarial Examples Exploiting the Rolling Shutter Effect. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14666–14675.
[67] Jayaprakash Selvaraj, Gökçen Y Dayanıklı, Neelam Prabhu Gaunkar, David Ware, Ryan M Gerdes, Mani Mina, et al. 2018. Electromagnetic induction attacks against embedded systems. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM, 499–510.
[68] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. 2016. 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*. ACM,

1528–1540.
[69] Junjie Shen, Jun Yeon Won, Zeyuan Chen, and Qi Alfred Chen. 2020. Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing. In *29th USENIX Security Symposium (USENIX Security 20)*. 931–948.
[70] Hocheol Shin, Dohyun Kim, Yujin Kwon, and Yongdae Kim. 2017. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 445–467.
[71] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *24th USENIX Security Symposium (USENIX Security 15)*. 881–896.
[72] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. 2018. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*. 8312–8323.
[73] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. 2020. Light Commands: Laser-Based Audio Injection Attacks on Voice-Controllable Systems. In *29th USENIX Security Symposium (USENIX Security 20)*.
[74] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. 2020. Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*. 877–894.
[75] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
[76] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.
[77] Tesla. 2020. Autopilot. https://www.tesla.com/autopilot.
[78] Khai N Truong, Shwetak N Patel, Jay W Summet, and Gregory D Abowd. 2005. Preventing camera recording by designing a capture-resistant environment. In *International conference on ubiquitous computing*. Springer, 73–86.
[79] Patricia Vitoria and Coloma Ballester. 2019. Automatic Flare Spot Artifact Detection and Removal in Photographs. *Journal of Mathematical Imaging and Vision* 61, 4 (2019), 515–533.
[80] Waymo. 2020. Waymo. https://waymo.com.
[81] Wikipedia. 2021. Pan–tilt–zoom camera. https://en.wikipedia.org/wiki/Pan-tilt-zoom-camera.
[82] Chaowei Xiao, Ruizhi Deng, Bo Li, Taesung Lee, Benjamin Edwards, Jinfeng Yi, Dawn Song, Mingyan Liu, and Ian Molloy. 2019. AdvIT: Adversarial Frames Identifier Based on Temporal Consistency in Videos. In *IEEE ICCV*.
[83] C. Yan, H. Shin, C. Bolton, W. Xu, Y. Kim, and K. Fu. 2020. SoK: A Minimalist Approach to Formalizing Analog Sensor Security. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 480–495. https://doi.org/10.1109/SP.2020.00026
[84] Chen Yan, Wenyuan Xu, and Jianhao Liu. 2016. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEF CON* 24 (2016).
[85] Mingjun Yin, Shasha Li, Zikui Cai, Chengyu Song, M Salman Asif, Amit K Roy-Chowdhury, and Srikanth V Krishnamurthy. 2021. Exploiting Multi-Object Relationships for Detecting Adversarial Attacks in Complex Scenes. In *International Conference on Computer Vision*.
[86] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
[87] Honggang Yu, Haocheng Ma, Kaichen Yang, Yiqiang Zhao, and Yier Jin. 2020. Deepem: Deep neural networks model recovery through em side-channel information leakage. In *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 209–218.
[88] YunYang1994. 2020. tensorflow-yolov3. https://github.com/YunYang1994/tensorflow-yolov3.
[89] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. DolphinAttack: Inaudible Voice Commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 103–117. https://doi.org/10.1145/3133956.3134052
[90] Jindi Zhang, Yifan Zhang, Kejie Lu, Jianping Wang, Kui Wu, Xiaohua Jia, and Bin Liu. 2020. Detecting and Identifying Optical Signal Attacks on Autonomous Driving Systems. *IEEE Internet of Things Journal* (2020).
[91] Youqian Zhang and KB Rasmussen. 2020. Detection of electromagnetic interference attacks on sensor systems. In *IEEE Symposium on Security and Privacy (S&P)*.
[92] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. 2019. 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*. ACM, 1989–2004.
[93] Ce Zhou, Qiben Yan, Yan Shi, and Lichao Sun. 2022. DoubleStar: Long-Range Attack Towards Depth Estimation based Obstacle Avoidance in Autonomous Systems. In *USENIX Security Symposium*.
[94] Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, and Kehuan Zhang. 2018. Invisible mask: Practical attacks on face recognition with infrared. *arXiv preprint arXiv:1803.04683* (2018).