GhostImage: Remote Perception Attacks against Camera-based Image Classification Systems

Yanmao Man[1], Ming Li[1], and Ryan Gerdes[2]

[1]University of Arizona
[2]Virginia Tech

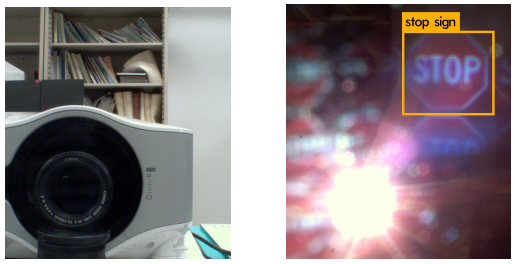
Abstract

In vision-based object classification 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 an alarm to indicate the presence of an intruder in surveillance settings. In this work we demonstrate how the perception domain can be remotely and unobtrusively exploited to enable an attacker to create spurious objects or alter an existing object. An automated system relying on a detection/classification framework subject to our attack could be made to undertake actions with catastrophic results due to attacker-induced misperception.

We focus on camera-based systems and show that it is possible to remotely project adversarial patterns into camera systems by exploiting two common effects in optical imaging systems, viz., lens flare/ghost effects and auto-exposure control. To improve the robustness of the attack to channel effects, 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 image datasets, in indoor and outdoor environments, and with three different cameras. Experimental results show that, depending on the projector-camera distance, attack success rates can reach as high as 100% and under targeted conditions.

1 Introduction

Object detection and classification have been widely adopted in autonomous systems, such as automated vehicles [1, 2] and unmanned aerial vehicles [3], as well as surveillance systems, e.g., smart home monitoring systems [4, 5]. 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).



(a) Projector off (b) Projector on

Figure 1: A STOP sign image was injected into a camera by a projector, which was detected by YOLOv3 [17].

While the cyber (digital) attack surface of such systems have been widely studied [6, 7], 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 demonstrated false data injection against sensors in a remote manner via either electromagnetic (radio frequency) interference [8], laser pulses (against microphones [9], or LiDARs [10–12]), and acoustic waves [13, 14]. 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 [15, 16]. These attacks can also be remote in that the attacker needn't physically contact/access/modify devices or objects.

Among the aforementioned sensors, at least for automated systems in the transportation and surveillance domains, cameras are more common/crucial. Existing *remote* attacks against cameras are limited to, essentially, denial-of-service attacks [11, 18, 19], which are easily detectable (e.g., by tampering detection [20]) and for which effective mitigation strategies exist (e.g., by sensor fusion [21]). In this work, we consider attacks that cause camera-based image classification

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-determined object (e.g., changing a STOP sign to a YIELD sign or changing an intruder into a bicycle).

As it is not possible, due to optical principles, to directly project an image into a camera, 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's 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 detected and classified as a STOP sign by YOLOv3 [17], a state-of-the-art object detector.

Theoretically arbitrary patterns can be injected via ghosts. However, it is challenging to practically and precisely control the ghosts, in terms of their resolutions and positions in images, making arbitrary injection impracticable in some scenarios. Hence, we propose an empirical projector-camera channel model that predicts the resolution and color of injected ghost patterns, as well as the location of ghosts, for a given projector-camera arrangement. Experimental results show that at short distances attack success rates are as high as 100%, but at longer distances the rates decrease sharply; this is because at long distances ghost resolutions are low, resulting in patterns that cannot be recognized by the classifier.

To improve the efficacy of our attack, which we dub GhostImage, especially at lower resolutions, we assume that the attacker possesses knowledge about the image classification/detection algorithm. Based on this knowledge the attacker is able to formulate and solve an optimization problem to find optimal attack patterns, of varying resolutions, to project that will be recognized by the image classifier as the intended target class [23, 24]; i.e., the pattern projected will yield a classification result of the attacker's choice. As the channel may distort the injected image (in terms of color, brightness, and noise), we extend our projector-camera model to include auto exposure control and color calibration and integrate the channel model into our optimization formulation. This results in a pattern generation approach that is resistant to channel effects and thus able to defeat a classifier 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, and environmental conditions. Results show that our attacks are able to achieve attack success rates as high as 100%, depending on the projector-camera distance. Our contributions are summarized as follows.

- We are the first to study remote perception attacks against camera-based classification systems, whereby the attacker induces misclassification of objects by injecting light, conveying adversarially generated patterns, into the camera.
- Our attack leverages optical effects/techniques, namely, lens flare and auto-exposure control, that are widespread and common, making the attack likely to be effective against most cameras. Furthermore, we incorporate these effects in an end-to-end manner into an adversarial machine learning-based optimization framework to find the optimal patterns an attacker should inject to cause misperception.
- We demonstrate the efficacy of the attacks through experiments with varying image datasets, cameras, distances, and indoor to outdoor environments. Results show that GhostImage attacks are able to achieve attack success rates as high as 100%, depending on the projector-camera distance.

2 System and Threat Model

System and attack models are described, including two attack objectives and the attacker's capabilities.

2.1 System Model

We assume an end-to-end camera-based object classification system (Fig. 2) in which a camera captures an image of a scene with objects of interest. The image is then fed to an object detector to crop out the areas of objects, and finally these areas are given to a neural network to classify the objects. Autonomous systems increasingly rely on such classification systems to make decisions and actions. If the classification 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.

2.2 Threat Model

We consider two different attack objectives. In **creation attacks** the goal is to inject a spurious (i.e., non-existent) object

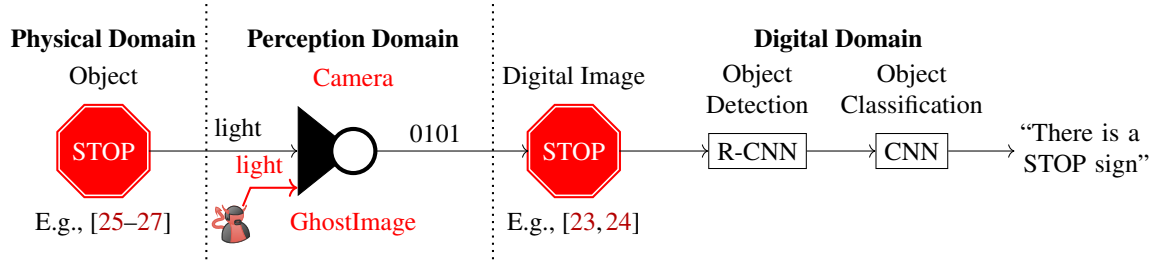


Figure 2: Camera-based object classification systems. GhostImage attacks target the perception domain, i.e., the camera.

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** who 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 can 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 the camera-aware attackers, but also know about the image classifier including its architecture and parameters, i.e., black-box attacks on the camera but white-box attacks on the classifier. With such capabilities, it is able to achieve creation attacks and alteration attacks as well, but with higher attack success rates.

Both types of attackers are remote (unlike the lens sticker attack [28]), i.e., they do not have 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 and aim victim cameras [12, 18, 29].

3 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.

3.1 Optical Imaging Principles

Due to the optical principles of camera-based imaging systems, it is not feasible to directly point a projector at a camera, hoping that the projected patterns can appear at the same location with the image of the targeted object, because 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, 30] refer to a phenomenon where one or more undesirable artifacts appear on an image because bright light get scattered or flared in a non-ideal lens system (Fig. 3). Ideally, all light beams should pass directly through the lens and reach the CMOS sensor. However, due to the quality of the lens elements, a small portion of light gets 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 [31]), the ghost effects become visible. Fig. 3 shows only one reflection path, but there are many other paths and that is why there are usually multiple ghosts in an image.

Existing literature [22] about ghosts focused on the simulation of ghosts given the detailed lens configurations, in which the algorithms simulate every possible reflection path. Such white-box models are computationally expensive, and also requires white-box knowledge of internal lens configurations, thus are not suitable for our purposes. In Sections 4 and 5, we study flare effects in a black-box manner (more general than Vitoria et al. [30]), where we train a lightweight end-to-end model that is able to predict the locations of ghosts, estimate the resolutions within ghost areas, and also calibrate colors.

Exposure control mechanisms [32] 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

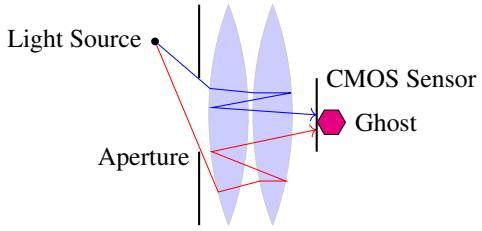


Figure 3: Ghost effect principle

balance between the targeted object and the injected attack patterns in ghosts.

3.2 Neural Nets and Adversarial Examples

We abstract a neural network as a function $Y = f_\theta(x)$ and we omit the details of it due to the page limit. 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 to 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 $C(x) = \text{argmax}_i Y_i$. Also, the inputs to the softmax layer are called *logits* and denoted as $Z(x)$.

An adversarial example [23] is denoted as y, where $y = x + \Delta$. Here, Δ is additive noise that has the same dimensionality with x. Given a benign image x and a target label t, an adversary wants to find a Δ such that $C(x+\Delta) = t$, i.e., *targeted attacks*. Note that, in this paper, the magnitude of Δ is not constrained below a small threshold, since the perceived images are usually not directly observed by human users. But we still try to minimize it because it represents the attack power and cost.

4 Camera-aware GhostImage Attacks

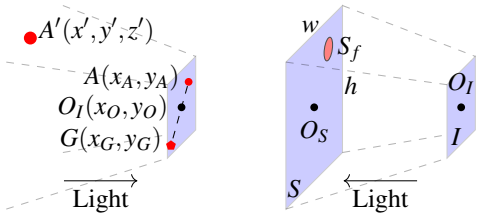
In this section, we will discuss how a camera-aware attacker is able to inject arbitrary patterns in the perceived image of the victim camera using projectors.

4.1 Technical Challenges

Since we assume that the attacker do not have access to the images that the targeted camera captures, he/she will have to be able to predict how ghosts might appear in the image. First, 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. Second, since a projector can inject shapes in ghost areas, the attacker needs to find out the maximum resolution of shapes that it can inject. Lastly, it is also challenging to realize the attacks derived from the position and resolution models above with a limited budget.

4.2 Ghost Pixel Coordinates

Given the pixel coordinates of the target object G (Fig. 4a), we need to derive the real-world coordinates A' of the projector so that we know where to place the projector in order to let one of the ghosts overlap with the image of the object. To do this, we derive the relationship between G and A' in two steps: We first calculate the pixel coordinates of the light source A given A', and then we calculate G based on A.



(a) Capture (b) Projection

Figure 4: Capture and projection are reverses of each other.

Based on homogeneous coordinates [33], assuming the camera is at the origin of the coordinate system, we have

$$(u,v,w)^\top = M_c \cdot (x',y',z',1)^\top, \tag{1}$$

where M_c is the camera's geometric model [33], a 3×4 matrix. M_c can be trained from another (similar) camera, and then be applied to the victim camera. The coordinates of A is then $A = (x_A,y_A)^\top = (u/w, v/w)^\top$, by the homogeneous transformation. Note that, A does not have to appear in the view of the camera, which makes the attack more stealthy [31].

In order to find the relationship of the pixel coordinates between light sources A and their ghosts G, we did a simple experiment where we moved around a flashlight in front of the camera [34], and recorded the pixel coordinates of the flashlight and the ghosts. Similar to Vitoria et al.'s results [30], we observe that, for each G, we have $\overline{AO_I}/\overline{O_IG} = r_G$ (being constant), wherever A is, and $r \in (-\infty,\infty)$. This means the feasible region for the placement of the projector is large; to attack an autonomous vehicle, for example, it can be located on an overbridge, on a traffic island, or even in the preceding vehicle or on a drone, etc. Finally, given $A = (x_A,y_A)$, $O_I = (x_O,y_O)$ and r, we can derive the coordinates of ghosts as

$$G = \begin{pmatrix} x_O - (x_A - x_O)/r \\ y_O - (y_A - y_O)/r \end{pmatrix}. \tag{2}$$

With G's coordinates, the attacker is able to predict the pixel location of ghosts and try adjusting the position and orientation (which implies the angle) of the light source in the real world so as to align one or more ghosts with the image of the object, whose pixel coordinates can be derived using (1), too.

4.3 Ghost Resolution

In our daily life, ghosts normally appear as pieces of single-color polygon-shaped artifacts; this is because the light sources that cause these regular ghosts are single-point sources of light that have just one single color, such as light bulbs, flashlights, etc. In this work, however, we find out that one is able to bring patterns into these ghost areas by simply using a low-cost projector, a special source of light that shines variant patterns in variant colors. For example, in Fig. 1, an

image of a STOP sign that is projected by a projector, appears in one of the ghost areas in the image; this is because the pixel resolution of the projector is high enough that multiple light beams in different colors (got reflected among lenses and then) go into the same ghost. In this subsection, we study the resolution of the patterns in ghost areas [1].

Let us first define the *throwing ratio* of a projector. In Fig. 4b, let plane S be the projected screen (e.g. on a wall), whose height and width are denoted as h and w, respectively. The distance $d = \overline{O_S O_I}$ is called the throwing distance. The throwing ratio of this projection is $r_{\text{throw}} = d/w$. The (physical) size of the projected screen at the victim camera's location is denoted S_O, a part of which is captured by the CMOS sensor of the camera in the ghost area, and we denote the (physical) size of that area as S_f. Let us also define the resolution of the entire projected screen as P_O in terms of pixels (e.g., 1024×768), and the resolution of the ghost as P_f. Clearly, there is a linear relationship among them: $P_f/P_O = S_f/S_O$, where $S_O = wh$. Finally, we can calculate the resolution of the ghost given d and r_{throw}:

$$P_f = \frac{P_O S_f}{\frac{h}{w}\left(\frac{d}{r_{\text{throw}}}\right)^2}. \tag{3}$$

Here, S_f is a constant because the size of the lens is fixed; e.g., the camera [34] has $S_f = 0.0156\ \text{cm}^2$.

4.4 Attack Realization and Experiment Setup

According to Eq. 3, if the attacker wants to carry out long-distance and high-resolution GhostImage attacks, he/she needs a projector with a large throwing ratio r_{throw}. However, the factory longest-throw lenses (NEC NP05ZL Zoom Lens [37]) of our projector can achieve a throwing ratio of maximum 7.2 (which means 9×9 at one meter), and expensive (about $1600). Instead, we use a cheap ($80) zoom lens (Fig. 5, Right) [36] that was originally designed for Canon cameras. In our experiments, such a configuration is interestingly feasible[2] (Fig. 5), achieving the maximum throwing ratio of 20 when the focal length is 250 mm, which means that at a distance of one meter, 32×32-resolution attacks can be achieved. See Sec. 7.1 for more discussion on lens and projector selection.

Fig. 5 (left) shows a general diagram of GhostImage attacks, where the light source (i.e., a projector) is pointing at the camera from the side, so that the camera can still capture the object (e.g., a STOP sign) for alteration attacks. The light source injects light interference (marked in blue) into the camera, which gets reflected among the lenses of the camera, resulting in ghosts that overlap with the object in the image.

[1] We are interested in the resolution of the projector pixels, not camera pixels; a projector pixel is usually captured by multiple camera pixels.

[2] Because projectors and cameras are dual devices (Fig. 4), their lenses are interchangeable.

Accordingly, a photo of our in-lab experiment setup is given in Fig. 5. The Canon lens was loaded in the NEC projector, though it cannot be seen in the photo. We will evaluate our attack on three different cameras (Sec. 6.2.3).

To mount a creation attack, the attacker computes the maximum resolution P_f for the ghost with a distance d based on (3), and then *downsamples* the target image to the resolution P_f in order to fit in the ghost area. The attacker chooses downsampling as a heuristic approach because he/she is not aware of the classification algorithm.

To mount alteration attacks, in addition to (3) for downsampling, the attacker also needs to consider the pixel coordinates (Eq. 2) of the ghost because the attacker needs to align the ghost with the image of the object of interest so that the resulting, combined image deceives the classifier.

4.5 Camera-aware Attack Evaluation

We substantiate camera-aware attacks on an image classification system that we envision would be used for automated vehicles. Specifically, images, taken by an Aptina MT9M034 camera [34], are fed to a traffic sign image classifier trained on the LISA dataset [38]. In Sec. 6, we will evaluate classification systems for other applications, with different cameras and different datasets.

4.5.1 Dataset and neural network architecture

In order to train an unbiased classifier, we selected eight traffic signs (with 80 instances) from the LISA dataset [38] (Fig. 12a). The network architecture is identical to [25]. We used 80% of samples from the balanced dataset to train the network and the rest 20% to test the network; it achieved an accuracy of 96%.

4.5.2 Evaluation methodology

We iterated five distances, m source classes, m target classes. For each target class, we sampled k images randomly from the dataset. For every combination, we first downsampled the target image based on (3), and projected the image at the camera using the NEC projector. We then took the captured image, cropped out the ghost area, and used the classifier to classify it. If the classification result is the target class, we count it as a successful attack. The procedure for creation attacks is slightly different: Rather than printed traffic signs, we placed a blackboard as the background as it helped us locate the ghosts. Given a throwing radio of 20 (thanks to the Canon lens) we evaluated five different distances from one meter to five meters. Based on (3), they resulted in 32×32, 16×16, 8×8, 4×4, and 2×2 resolutions, respectively.

Figure 5: (Left) Attack setup diagram. (Middle) In-lab experiment setup. (Right) Attack equipments: We replaced the original lens of the NEC NP3150 Projector [35] with a Canon EFS 55-250 mm zoom lens [36].

4.5.3 Results

The results about attack success rates of camera-aware attacks at varying distances are shown in Table 1 (Fig. 6 illustrates two successful camera-aware attacks). For the digital domain, we simply added attack images Δ on benign images x as $y = (x+\Delta)/\|x+\Delta\|_\infty$. Based on these experiments, we observe: First, as the distance increases, the success rate decreases. This is because lower-resolution images are less well recognized by the classifier. Second, digital domain results are better than perception domain one, because images are distorted by the projector-camera channel effects. Third, creation attacks result in higher success rates than alteration attacks do because in alteration attacks there are benign images in the background, encouraging the classifier to make correct classifications. We will address these issues in the next section, so as to increase the overall attack success rate.

5 System-aware GhostImage Attacks

There are some limitations of the camera-aware attack introduced in the previous section. 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, as well as an optimization formulation based on which the attacker can solve for optimal attack patterns that cause misclassification by the target classifier with high confidence.

Table 1: Camera-aware attack success rates

Distances	Creation Attacks		Alteration Attacks	
(meter)	Digital	Perception	Digital	Perception
1	98%	41%	95%	33%
2	98%	36%	88%	33%
3	80%	34%	67%	34%
4	36%	15%	28%	10%
5	14%	10%	13%	0%

5.1 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 not able 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 distances and projector lens (Eq. 3), thus the ghost patterns must be carefully designed to fit the resolution with few degrees of freedom.

5.2 System-aware Attack Overview

The system-aware attacker aims to find optimal patterns that can cause misclassification by the target classifier with high confidence by taking advantage of the non-robustness of the classifier [23]. We adopt an adversarial example-based optimization formulation into GhostImage attacks, in which the attacker tries to solve

$$\Delta^* = \arg\min_{\Delta} \ \|\Delta\|_p + c \cdot \mathcal{L}_{\text{adv}}(y, t, \theta), \tag{4}$$

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

Figure 6: Camera-aware attack examples at one meter in perception domain. Left: Creating a Merge sign. Right: Altering a STOP sign (in the background) into a Merge sign.

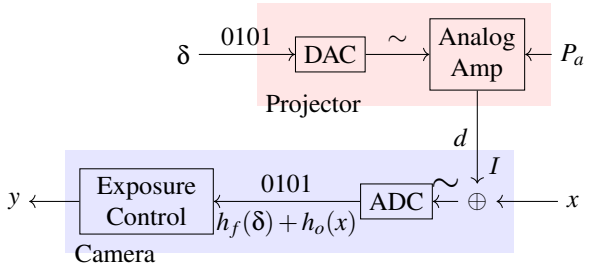


Figure 7: Projector-camera channel model

is the target class, 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. Here, we aim to minimize the power of the projector required for a successful attack, meanwhile maximizing the successful chance of attacks. The relative importance of these two objectives is balanced by a constant c. Sec. 5.4 details (4) in terms of how we handle Δ being a non-negative tensor that is also able to depict grid-style patterns in different resolutions.

More importantly, in (4) y is the final perceived image used as input to the classifier, which is estimated by our channel model in an end-to-end style (Fig. 7), 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

$$y = g\left(h_f(\Delta) + h_o(x)\right). \tag{5}$$

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 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. 5.3 introduces the derivation of (5).

Next, we will first present the channel model, and then formulate the optimization problem for finding the optimal adversarial ghost patterns.

5.3 Projector-Camera Channel Model

We consider the projector to camera channel model (Fig. 7) 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 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

[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.

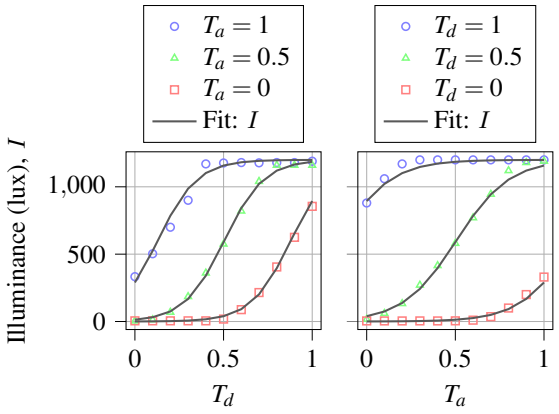


Figure 8: Illuminance depends on the RGB amplitude T_d, and the light bulb intensity T_a.

channel would yield $y = x + \delta$ but due to channel effects, we need to find a way to adjust the projected RGB value such that the perceived RGB value is as intended, i.e., to find the appropriate x given y.

5.3.1 Exposure control

As we discussed in Section 3.1, cameras are usually equipped with auto-exposure control, where according to the overall brightness of the image, the camera adjusts its exposure by changing the exposure time, or the size of its aperture, or both. We observed from our experiments that, as we increase the luminescence of the projector (I), in the image the brightness of the object (x) decreases but the ghost (δ) does not decrease as much. Modeling such phenomena helps the attacker to precisely predict the perceived image. For the following, we will first find out how the illuminance I depends on δ and P_a (the normalized power of light bulb ranging from 0% to 100%), and then analyze how y depends on I.

How does I depend on δ and P_a? We conducted a series of experiments, where $T_d = \|\delta\|_\infty = \max_i \delta_i$ and P_a were varied. We recorded the illuminance directly in front of the camera using an illuminance meter, with the projector one meter away. The results are plotted in Fig. 8, which shows that

$$I(T_d, P_a, d) = \frac{c_d}{d^2} \cdot \frac{I_{\max}}{1 + e^{-t}}, \tag{6}$$

where $t = a \times T_d + b \times P_a + c_t$, and a, b, c_d and c_t are constants derived from the data. $I_{\max}$ is the maximum illuminance of the projector at a distance of one meter. Such a sigmoid-like function captures the luminescence saturation property of the projector hardware.

How does the perceived x depend on I? In the same experiments we also recorded the RGB value of the ghost (δ)

with a blackboard as background (in order to reduce ambient impacts), and a piece of white paper (x) that was also on the blackboard but did not overlay with the ghost. Their data are shown in Fig. 9, from which we can derive the *dimming ratio* that measures the change of exposure/brightness:

$$\gamma(I) = \frac{I_{\text{env}}}{I + I_{\text{env}}}, \tag{7}$$

where I_{env} is the ambient lighting condition in illuminance which differs from indoors to outdoors for instances. From this equation, we see that in an environment with static lighting condition, as the luminescence of the projector increases, the dimming ratio decreases, hence the objects become darker. With (7), the adversary is able to conduct real-time attacks by simply plugging in the momentary I_{env}.

How does the perceived δ depend on I? When $x = 0$, $\|y_f\| = \|y_f\|_\infty$ (the lower subplot of Fig. 9) depends on I in two ways:

$$\|y_f\|(I) = \gamma(I) \cdot \rho \cdot I.$$

On one hand, the last term I increases the intensity of ghosts, but on the other hand the dimming ratio $\gamma(I)$ dims down ghost, whereby ρ is a trainable constant. With this, we can rewrite the perceived flare as

$$y_f = \|y_f\| H_c \frac{\delta}{\|\delta\|},$$

where H_c is the color calibration matrix to deal with color distortion, which will be discussed in Section 5.3.2. The term $1/\|\delta\|$ normalizes δ. In the end, we have the channel model

$$y = \gamma(I) \left(\rho I H_c \frac{\delta}{\|\delta\|} + x \right). \tag{8}$$

Compared to (5),

$$h_f(\delta) = \rho I H_c \frac{\delta}{\|\delta\|}, \quad g(t) = \gamma(I)t, \quad h_o(x) = x.$$

With (8), the attacker is able to predict how bright and what colors/pixel values the ghost and the object will be, given the projected pixels, the power of the projector, and the distance.

5.3.2 Color calibration

Considering a dark background (i.e., $x = 0$), (8) can be simplified as $y = \gamma(I)\rho I H_c \delta / \|\delta\|$, where H_c is a 3×3 matrix (as three color channels) that calibrates colors. Both y and δ are 3×1 column vectors. H_c should be an identity matrix for an ideal channel, but due to the color-imperfection of both the projector and the camera, H_c needs to be learned from data. To simplify notations, we define corrected x and y as

$$\hat{x} = \frac{\delta}{\|\delta\|}, \quad \hat{y} = \frac{y}{\rho I \gamma(I)},$$

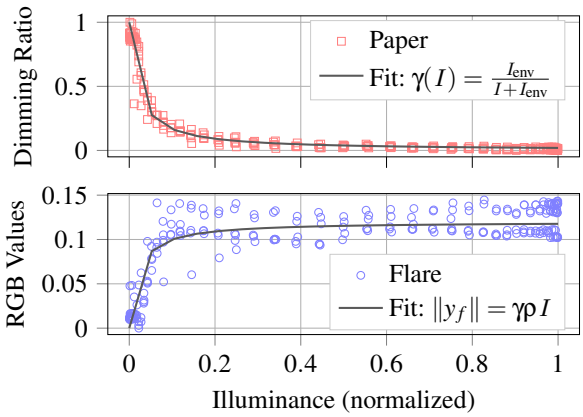


Figure 9: Perceived RGB values v.s. illuminance.

so that we can write

$$\hat{y} = H_c \hat{x}.$$

We did another set of experiments where we collected $n = 100$ pairs of $(\hat{x}, \hat{y})$ with dark background (to make $x = 0$), with δ being assigned randomly, and $P_a = 30\%$. We grouped them into X and Y:

$$X = \left[\hat{x}_1^\top, \hat{x}_2^\top, \cdots, \hat{x}_n^\top\right]^\top, \quad Y = \left[\hat{y}_1^\top, \hat{y}_2^\top, \cdots, \hat{y}_n^\top\right]^\top,$$

where both X and Y are $n \times 3$ matrices. We solve

$$\min_{H_c} \|Y - XH_c\|_2^2.$$

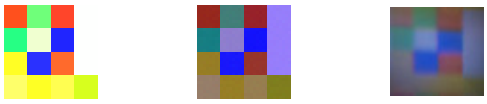
to obtain H_c, which is known as a non-homogeneous least square problem [33], and has a closed-form solution:

$$H_c = \left((X^\top X)^{-1} X^\top Y\right)^\top.$$

Plugging H_c back to (8) completes our channel model.

5.3.3 Model validation

Fig. 10 demonstrates the accuracy of our channel model. In it the left image is the original input to the projector, the middle image is the estimated output from the camera based on our channel model (Eq. 8), and the image on the right is the actual image in a ghost captured by the camera. As can be seen, the difference between the actual and predicted is much less than the actual and original. While blurring effect is apparent in the actual y, we do not model it but the success rates are still high despite it. As we will see in Section 6, our channel model is general enough that once trained on one camera in one environment, it can be transferred to different environments and different cameras without retraining.



(a) Input x_f (b) Estimated y (c) Actual y

Figure 10: An example of channel model prediction

5.4 Optimal Adversarial Projection Patterns

In long-distance, low-resolution GhostImage attacks there are only a few pixels in the ghost area. A camera-aware attacker's strategy is to simply downsample attack images into low resolutions, but that does not result in high success rates. While (4) is abstract, for the rest of this subsection, we will progressively detail it and show how it can be solved in light of the channel model to improve attack success rates. We will start with the simplest case where adversarial perturbations are random noise (Sec. 5.1). Then, single-color ghosts will be introduced. Later, we will consider how to find semi-positive additive noise due to the fact that superposition can only increase perceived light intensity but not decrease it. Finally, we examine the optimization problem to find optimal ghost patterns in grids at different resolutions.

5.4.1 Ghosts in random noise

Let us consider the simplest case first where the random noise Δ is drawn from one single Gaussian distribution for all three channels, i.e., $\Delta \sim \mathcal{N}(\mu, \sigma^2)$, where the size of Δ is $w \times h \times 3$ with w and h representing the width and height of the benign image x. This is because the values of each pixel that appear in the ghost area follow Gaussian distributions according to statistics obtained from our experiments.

The adversary needs to find μ and σ such that when Δ is added to the benign image x, the resulting image y will be classified as the target class t. That said, the logits value (Section 3.2) of the target class should be as high as possible compared with the logits values of other classes [24]. Such a difference is measured by the loss function $\mathcal{L}_{\text{adv}}(y,t)$

$$\mathcal{L}_{\text{adv}}(y,t) = \max\left\{-\kappa, \max_{i:i\neq t}\{\mathbb{E}[Z_i(y)]\} - \mathbb{E}[Z_t(y)]\right\}, \quad (9)$$

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. The attacker needs $\mathcal{L}_{\text{adv}}$ as low as possible so that the neural network would classify y as Class t. Most importantly, y is computed based on our channel model (Eq. 8), so that the optimizer finds the optimal ghost patterns that are resistant to the channel effects. Unfortunately, due to the complexity of neural networks, the expectations of logits values $\mathbb{E}[Z_i(y)]$ are hard to be expressed analytically; we instead use Monte Carlo methods to approximate it:

$$\hat{\mathbb{E}}[Z_i(y)] = \frac{1}{T}\sum_{j=1}^{T} Z_i(y_j),$$

where T is the number of trials, and y_j is of the j-th trial.

Meanwhile, the adversary also needs to minimize the magnitude of Δ to reduce the attack power and noticeability, as well as its peak energy consumption, quantified by σ. The expectation of the magnitude of Δ is

$$\mathbb{E}[\|\Delta\|_p] = \mu n^{1/p}, \quad \text{with } n = 3wh. \quad (10)$$

Putting (9) and (10) together with a tunable constant c, we have our optimization problem for the simplest case

$$\begin{aligned}\mu^*, \sigma^* = \ & \arg\min_{\mu,\sigma} && \mathbb{E}[\|\Delta\|_p] + \sigma + c \cdot \mathcal{L}_{\text{adv}}(y,t), \\ & \text{subject to} && \sigma > \sigma_l,\end{aligned}$$

Here, σ_l is the lower bound of the standard deviation σ, meaning that the interference generator and the channel environment can provide random noise with the standard deviation of at least σ_l. When $\sigma_l = 0$, the adversary is able to manipulate pixels deterministically. Therefore, when we fix σ as σ_l in the optimization problem, the attack success rate when deploying μ^* would be the lower bound of the attack success rate. In other words, the adversary equipped with an attack setup that can produce noise with a lower variance (than σ_l^2) can carry out attacks with higher success rates. Therefore, we can simplify our formulation by removing the constraint about σ, so the optimization problem becomes

$$\mu^* = \arg\min_{\mu} \ \mathbb{E}[\|\Delta\|_p] + c \cdot \mathcal{L}_{\text{adv}}(y,t). \quad (11)$$

For the rest of the paper we will simply use σ to denote σ_l.

5.4.2 Ghosts in single-color

Since in (11) there is only one variable that the adversary is able to control, it is infeasible to launch a targeted attack with such few degrees of freedom. As a result, the adversary needs to manipulate each channel individually. That is, for each channel, there will be an independent distribution from which noise will be drawn. This is feasible because noise can appear in different colors in the ghost areas in which three channels are perturbed differently when using projectors. Let us decompose Δ as $\Delta = [\Delta_R, \Delta_G, \Delta_B]$, where the dimension of $\Delta_{\{R,G,B\}}$ is $w \times h$, and they follow three independent Gaussian distributions

$$\Delta_R \sim \mathcal{N}(\mu_R, \sigma_R^2), \ \Delta_G \sim \mathcal{N}(\mu_G, \sigma_G^2), \ \Delta_B \sim \mathcal{N}(\mu_B, \sigma_B^2).$$

Here, $\mu_{\{R,G,B\}}$ and $\sigma_{\{R,G,B\}}$ are the means and the standard deviations (σ) of the three Gaussian distributions, respectively.

The expectation of such Δ is then

$$\mathbb{E}[\|\Delta\|_p] = \left[\frac{n}{3}\left(\mu_R^p + \mu_G^p + \mu_B^p\right)\right]^{\frac{1}{p}}. \tag{12}$$

(10) is a special case of (12) when $\mu = \mu_R = \mu_G = \mu_B$. We denote $\boldsymbol{\mu} = [\mu_R, \mu_G, \mu_B]^\top$. Hence, similar to (11), we have the optimization problem for single-color perturbation [39]

$$\boldsymbol{\mu}^* = \arg\min_{\boldsymbol{\mu}} \ \mathbb{E}[\|\Delta\|_p] + c \cdot \mathcal{L}_{\text{adv}}(y,t), \tag{13}$$

by which the adversary finds $\boldsymbol{\mu}^*$ from which Δ is drawn.

5.4.3 Ghost grids

Since projector's pixels are arranged in grids, the attack patterns are in grids as well, especially in lower resolutions. We enable Δ with patterns in different resolutions. Such a grid pattern $\boldsymbol{\Delta}$ can be composed of several blocks $\Delta_{i,j,k}$, i.e., $\Delta_{i,j,k} : \{1 \le i \le N_{\text{row}}, 1 \le j \le N_{\text{col}}, 1 \le k \le N_{\text{chn}}\}$ where N_{row}, N_{col} and N_{chn} is the number of rows, columns, and channels of a grid pattern, respectively, in terms of blocks. In a word, $\Delta_{i,j,k}$ is the perturbation block at i-th row, j-th column and k-th channel. A block $\Delta_{i,j,k}$ is a random matrix and its size is $\frac{w}{N_{\text{col}}} \times \frac{h}{N_{\text{row}}}$, so that the size of $\boldsymbol{\Delta}$ is still $w \times h \times 3$. Besides, the elements in the random matrix $\Delta_{i,j,k}$ is i.i.d. drawn from a Gaussian distribution, i.e., $\Delta_{i,j,k} \sim \mathcal{N}(\mu_{i,j,k}, \sigma^2)$.

The adversary finds the optimal grid pattern $\boldsymbol{\Delta}$ by solving

$$M^* = \arg\min_{M} \ \mathbb{E}[R(\|\boldsymbol{\Delta}\|_p)] + c \cdot \mathcal{L}_{\text{adv}}(y,t), \tag{14}$$

where $R(\boldsymbol{\Delta})$ is the softplus function to guarantee that the perturbation is always positive. $M = \{\mu_{i,j,k}\}$ is a tensor in shape $N_{\text{row}} \times N_{\text{col}} \times N_{\text{chn}}$. See Fig. 12a for some examples of adversarial grids in different resolutions.

6 System-aware Attack Evaluation

In this section, we consider camera-based image classification 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 (Section 4.5), and *attack robustness*, namely how well system-aware attacks are when being evaluated in different setups.

We will again use attack success rates as our metric. We used the Adam Optimizer [40] to solve our optimization problems. There are two sets of results: *Emulation results* refer to the classification results on emulated, combined images of benign images and attack patterns using our channel model (Equation 8). Emulation helps us conduct scalable and fast evaluations of GhostImage attacks before conducting real-world experiments[4]. *Experimental results* refer to the classification results on the images that are actually captured by the victim cameras when the projector is on.

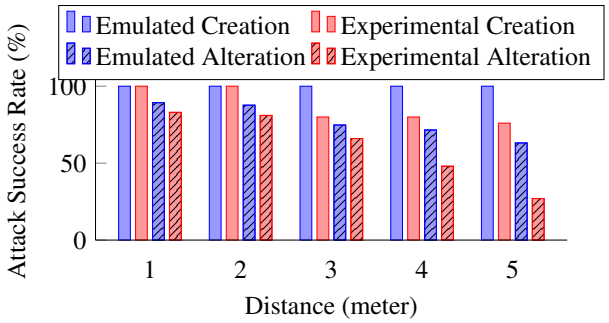


Figure 11: System-aware creation and alteration

6.1 Attack Effectiveness

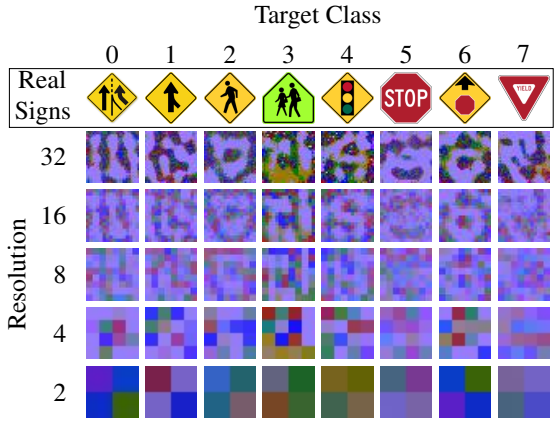
To compare with camera-aware attacks, system-aware attacks are evaluated in a similar procedure, targeting a camera-based object classification system with the LISA dataset and its classifier. The system uses an Aptina MT9M034 camera [34] in an in-lab environment.

6.1.1 Creation attacks

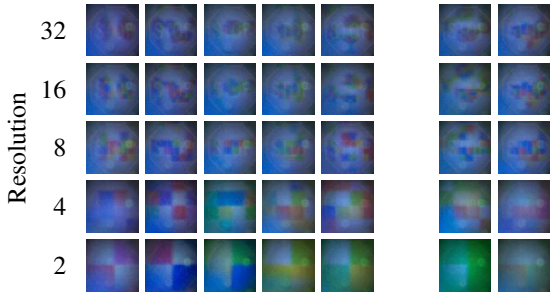
For emulated creation attacks, all distances (or all resolutions) yield attack success rates of 100% (Fig. 11), which means that our optimization problem is easy to solve. In terms of computational overhead, we need roughly 30 s per image at 2×2-resolution, and 10 s at 4×4 or above (because of more degrees of freedom) using an NVIDIA Tesla P100 [41]. Fig. 12a 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 [42] 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. 12a. However, CIFAR-10 [43] and ImageNet [44] 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

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



(a) Emulated creation attacks



(b) Experimental alteration attacks

Figure 12: System-aware attack pattern examples.

different, such as two different cats, but in LISA, two images of STOP signs do not look as different as two cats.

For the experimental results of creation attacks, we see that as distances increase, success rates decrease a little (Fig. 11), but much better than the camera-aware attacks (Table 1), because the optimization formulation helped find those optimal attack patterns with high confidence.

6.1.2 Alteration attacks

The emulated and experimental results of alteration attacks are shown in Fig. 11. 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. 11 with Table 1 for camera-aware alteration attacks, we can see large improvements. Fig. 12b 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 STOP.

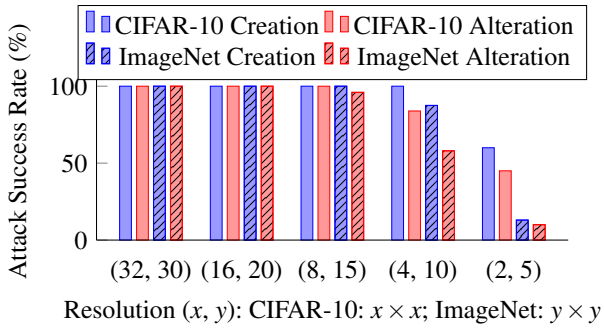


Figure 13: System-aware attacks on CIFAR-10 and ImageNet

6.2 Attack Robustness

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

6.2.1 Different image datasets

Here we evaluate our system-aware attacks on two other datasets, CIFAR-10 [43] and ImageNet [44], 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 [24]. The network was trained with the distillation defense [45] 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. 4.5.2. Results are shown in Fig. 13. 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 6.1.1), 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.

ImageNet We used a pre-trained Inception V3 neural network [46] 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 [24]. 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. 13.

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

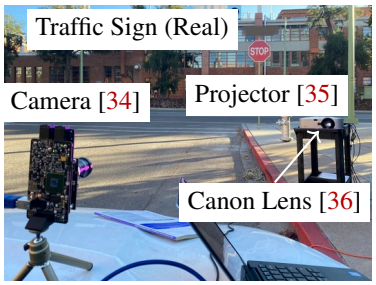


Figure 14: Outdoor experiment setup

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 were more successful than on LISA, because of the high variation within one class.

6.2.2 Outdoor experiments

In order to evaluate system-aware attacks in a real-world environment, we also conducted experiments outdoor (Fig. 14), 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 [47] 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. 7.1 for more discussion on ambient lighting conditions.

Results (Tab. 2) of the other cases show that the success rates are 30% lower than our in-lab experiments (the four-meter case from Fig. 11), 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.

6.2.3 Different cameras

Previously, we conducted GhostImage attacks on Aptina MT9M034 camera [34] designed for autonomous driving. Here, we evaluate two other cameras, an Aptina MT9V034 [48] with a simpler lens design, and a Ring indoor security camera [49] for surveillance applications.

Table 2: Outdoor alteration attack 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 6.1 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 [48] 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 [49] 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 [50] is detected by the Inception V3 neural network [46]. 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

Table 3: 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

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 3).

7 Discussion

In this section, we discuss practical challenges to GhostImage attacks, speculate as to effective countermeasures.

7.1 Practicality of GhostImage Attacks

Moving targets and alignment: 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. Note that, although aiming (or tracking) moving targets is generally challenging in remote sensor attacks (e.g., the AdvLiDAR attack [12] assumes the attacker can achieve this via camera-based object detection and tracking), existing works [18, 29] 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.

Conspicuousness: The light bursts around the light source in Figures 1 and 6 may raise stealthiness concerns about our attacks. However, according to our analysis in Sec. 4.2, such bursts can actually be eliminated because the light source can be outside of view [31]. 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 [1], 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. 14). Finally, the light source only needs to be on for a short amount of time, as a few tampered frames can cause incorrect actions [51].

Projectors, lenses, and attack distances: Based on our model (Eq. 7) and experiments (Tab. 4), 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., [52] with Luminance 9×10^3 lm) should work with a telephoto lens [53] (with a throwing radio 100) at a distance of one meter. For longer distances or brighter backgrounds, one can either acquire a more powerful projector (e.g., [47] with 75×10^3 lm), or combine multiple lenses to achieve much larger throwing ratios (e.g., two Optela lenses [53] yield 200, etc.), or both.

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 to 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, e.g., the AdvLiDAR attack [12]. Also, we assume white-box attacks on the neural network, though this assumption can be eliminated by leveraging the transferability of adversarial examples [54, 55].

Object detection: We have assumed that the object detector can crop out the region of the image which contains the projected ghost pattern(s). Though it cannot be guaranteed that an object detector will automatically include the ghost patterns, we note that a GhostImage attacker could design ghost patterns that cause an object detector to include them [27, 56] and, at the same time, the cropped image would fool the subsequent object classifier.

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. Please see our full version [57] for more details.

7.2 Countermeasures

The most straightforward countermeasure to GhostImage attacks is flare elimination, either by using a lens hood or through flare detection. Lens hoods are generally not favored as they reduce the angle of view of the camera, which is unacceptable for many autonomous vehicle and surveillance applications. Adversarial flare detection is challenging as they are typically transparent [58], and hard to be distinguished from natural ghosts.

A complementary line of defense would be to make neural networks themselves robust to GhostImage attacks. Existing approaches against adversarial examples (e.g., [45, 59–61], etc.) are ill-suited for this task, however, as GhostImage attacks do not necessarily follow the constraints placed on traditional adversarial examples in that perturbations do not have to be bounded within a small norm, meanwhile these defenses

were not designed for arbitrarily large perturbations. As this work mainly focuses on sensor attacks, we leave the validation of defenses as future work [12, 25–27].

Another complementary approach of defense is to exploit prior knowledge, such as GPS locations of signs, to make decisions, instead of only depending on real-time sensor perception (though this approach would not work for spontaneous appearance of objects, e.g., in the context of collision avoidance). Sensor redundancy/fusion could also be helpful: autonomous vehicles could be equipped with multiple cameras and/or other types of sensors, such as LiDARs and radars, which would at least increase the cost of the attack by requiring the attacker to target multiple sensors. However, a powerful attacker may be able to attack LiDARs [12], radars [19] and cameras simultaneously to defeat sensor fusion. Finally, temporal consistency via object tracking (e.g., "the object should not have appeared from nowhere of a sudden.") may also be used to detect the attack, or at least complicate it.

8 Related Work

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 [15, 16]. For example, anti-lock braking system (ABS) sensors have been manipulated via magnetic fields by Shoukry et al. [62], microphones have been subject to inaudible voice and light-based attacks [9, 63], and light sensors can be influenced via electromagnetic interference to report lighter or darker conditions [8]. The reader is referred to [15, 16] for a review of analog sensor attacks.

Existing remote attacks against cameras [11, 18, 19] are denial-of-service attacks and do not seek to compromise the object classifier as our GhostImage attacks do. Those attacks that do target object classification [25, 27, 64] 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 [15, 16]. Li et al. [28]'s attacks on cameras require attackers to place stickers on lenses, to which is generally hard to get access. Similarly, several light-based attacks [51, 65, 66] fall within the domain of physical attacks, as opposed to our perception domain attack, because these approaches illuminate the object of interest with visible or infrared light. We did not consider infrared noise in our attacks as it can be easily eliminated from visible light systems using infrared filters. Attacks on LiDAR systems [10–12, 67] 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.

Adversarial examples State-of-the-art adversarial examples can be categorized as digital (e.g., [23, 24]), or physical domain attacks (e.g., [25, 26, 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 [69], though they are limited in the digital domain.

9 Conclusion

In this work we presented GhostImage attacks against camera-based object classifiers. 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 that predicts the location of ghosts, the resolution of the patterns in ghosts, given the projector-camera arrangement, and accounts for exposure control and color calibration. GhostImage attacks also leverage adversarial examples generation techniques to find optimal attack patterns. We evaluated GhostImage attacks using three image datasets and in both indoor and outdoor environments on three cameras. Experimental results show that GhostImage attacks were able to achieve attack success rates as high as 100%, and also have potential impact on autonomous systems, such as self-driving cars and surveillance systems.

Acknowledgments

The work was partly supported by NSF grants CNS-1801402, CNS-1410000, and CNS-1801611. We would like to thank the anonymous reviewers, and Xiaolan Gu, Mingshun Sun for their helps on this work.

A Illustrative Channel Model Parameters

Table 4 lists all parameters of the projector-camera channel model. The color calibration matrix is

$$H_c = \begin{bmatrix} 0.5 & 0 & 0.1 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}.$$

References

[1] Waymo. Waymo. https://waymo.com, 2020.

[2] Tesla. Autopilot. https://www.tesla.com/autopilot, 2020.

[3] Amazon. Prime air delivery.

[4] Google. Nest and google home. now under one roof. nest.com, 2020.

[5] Amazon. Ring. ring.com, 2020.

Table 4: Channel model parameter examples

Description	Symbol	Value
Throwing ratio	r_{throw}	20
Physical size of ghosts	S_f	0.0156 cm^2
Projection resolution	P_O	1024×768
Flare booster	ρ	30
Bulb intensity	T_a	$[0,1]$
Ambient illuminance	I_{env}	300 lx (indoor)
Projector ill.	I	400 lx (at 1 m)
Projector max ill.	$I_{\max}$	1200 lx (at 1 m)
Camera matrix	M	See below
Color calibration matrix	H_c	See below
In Equation 6	a	8.9
In Equation 6	b	6.7
In Equation 6	c_t	-7.8
In Equation 6	c_d	0.25

[6] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, et al. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*, volume 4, pages 447–462. San Francisco, 2011.

[7] Andrei Costin, Jonas Zaddach, Aurélien Francillon, and Davide Balzarotti. A large-scale analysis of the security of embedded firmwares. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 95–110, 2014.

[8] Jayaprakash Selvaraj, Gökçen Y Dayanıklı, Neelam Prabhu Gaunkar, David Ware, Ryan M Gerdes, Mani Mina, et al. Electromagnetic induction attacks against embedded systems. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 499–510. ACM, 2018.

[9] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. Light commands: Laser-based audio injection attacks on voice-controllable systems.

[10] Hocheol Shin, Dohyun Kim, Yujin Kwon, and Yongdae Kim. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 445–467. Springer, 2017.

[11] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe*, 11:2015, 2015.

[12] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2267–2281. ACM, 2019.

[13] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. Rocking drones with intentional sound noise on gyroscopic sensors. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 881–896, 2015.

[14] Qiben Yan, Kehai Liu, Qin Zhou, Hanqing Guo, and Ning Zhang. Surfingattack: Interactive hidden attack on voice assistants using ultrasonic guided wave. In *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[15] C. Yan, H. Shin, C. Bolton, W. Xu, Y. Kim, and K. Fu. Sok: A minimalist approach to formalizing analog sensor security. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 480–495, Los Alamitos, CA, USA, may 2020. IEEE Computer Society.

[16] Ilias Giechaskiel and Kasper Bonne Rasmussen. Taxonomy and challenges of out-of-band signal injection attacks and defenses. *IEEE Communication Surveys & Tutorials*, 2020.

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

[18] Khai N Truong, Shwetak N Patel, Jay W Summet, and Gregory D Abowd. Preventing camera recording by designing a capture-resistant environment. In *International conference on ubiquitous computing*, pages 73–86. Springer, 2005.

[19] Chen Yan, Wenyuan Xu, and Jianhao Liu. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEF CON*, 24, 2016.

[20] Evan Ribnick, Stefan Atev, Osama Masoud, Nikolaos Papanikolopoulos, and Richard Voyles. Real-time detection of camera tampering. In *2006 IEEE International Conference on Video and Signal Based Surveillance*, pages 10–10. IEEE, 2006.

[21] Qingquan Li, Long Chen, Ming Li, Shih-Lung Shaw, and Andreas Nüchter. A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios. *IEEE Transactions on Vehicular Technology*, 63(2):540–555, 2013.

[22] Matthias B. Hullin, Elmar Eisemann, Hans-Peter Seidel, and Sungkil Lee. Physically-based real-time lens flare rendering. *ACM Trans. Graph. (Proc. SIGGRAPH 2011)*, 30(4):108:1–108:9, 2011.

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[24] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017.

[25] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

[26] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM, 2016.

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

[28] Juncheng B Li, Frank R Schmidt, and J Zico Kolter. Adversarial camera stickers: A physical camera attack on deep learning classifier. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2019.

[29] KYLE MIZOKAMI. China could blind u.s. satellites with lasers. https://www.popularmechanics.com/military/weapons/a29307535/china-satellite-laser-blinding/, 2019.

[30] Patricia Vitoria and Coloma Ballester. Automatic flare spot artifact detection and removal in photographs. *Journal of Mathematical Imaging and Vision*, 61(4):515–533, 2019.

[31] Gunawan Kartapranata. Lens flare at borobudur stairs kala arches, 2010.

[32] Hsien-Che Lee. *Introduction to color imaging science*. Cambridge University Press, 2005.

[33] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[34] ON semiconductor. *MT9M034 1/3-Inch CMOS Digital Image Sensor*.

[35] NEC. *NP Installation Series User's Manual*, 10 2007.

[36] Canon. Telephoto zoom ef-s 55-250mm.

[37] NEC. Np05zl, 4.62–7.02:1 zoom lens.

[38] Andreas Mogelmose, Mohan Manubhai Trivedi, and Thomas B Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012.

[39] Yanmao Man, Ming Li, and Ryan Gerdes. Poster: Perceived adversarial examples. In *IEEE Symposium on Security and Privacy*, 2019.

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[41] Nvidia. *NVIDIA TESLA P100 GPU ACCELERATOR*, 2016.

[42] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[43] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[45] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 582–597. IEEE, 2016.

[46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[47] Barco. Xdl-4k75.

[48] ON semiconductor. *MT9V034 1/3-Inch Wide-VGA CMOS Digital Image Sensor*, 2017.

[49] Ring. Indoor security cameras. https://shop.ring.com/collections/security-cams#indoor, 2019.

[50] Ring. Standard and advanced motion detection systems used in ring devices. https://support.ring.com/hc/en-us/articles/115005914666-Standard-and-Advanced-Motion-Detection-Systems-Used-in-Ring-Devices, 2020.

[51] Ben Nassi, Dudi Nassi, Raz Ben-Netanel, Yisroel Mirsky, Oleg Drokin, and Yuval Elovici. Phantom of the adas: Phantom attacks on driver-assistance systems.

[52] Epson. Pro l1490u wuxga 3lcd laser projector.

[53] Opteka. Opteka 650-1300mm telephoto zoom lens.

[54] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[55] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *29th USENIX Security Symposium (USENIX Security 20)*, 2020.

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

[57] Yanmao Man, Ming Li, and Ryan Gerdes. Ghostimage: Perception domain attacks against vision-based object classification systems. *arXiv preprint arXiv:2001.07792*, 2020.

[58] Yichao Xu, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi. Transcut: Transparent object segmentation from a light-field image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3442–3450, 2015.

[59] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[60] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy (S&P)*, 2019.

[61] Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. Defending against physically realizable attacks on image classification. In *8th International Conference on Learning Representations (ICLR)*, 2020.

[62] Yasser Shoukry, Paul Martin, Paulo Tabuada, and Mani Srivastava. Non-invasive spoofing attacks for anti-lock braking systems. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 55–72. Springer, 2013.

[63] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 103–117, New York, NY, USA, 2017. Association for Computing Machinery.

[64] Alesia Chernikova, Alina Oprea, Cristina Nita-Rotaru, and BaekGyu Kim. Are self-driving cars secure? evasion attacks against deep neural networks for steering angle prediction. In *IEEE Security and Privacy Workshop on IoT*. IEEE, 2019.

[65] Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, and Kehuan Zhang. Invisible mask: Practical attacks on face recognition with infrared. *arXiv preprint arXiv:1803.04683*, 2018.

[66] Luan Nguyen, Sunpreet S. Arora, Yuhang Wu, and Hao Yang. Adversarial light projection attacks on face recognition systems: A feasibility study, 2020.

[67] James Tu, Mengye Ren, Siva Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection, 2020.

[68] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*, pages 99–112. Chapman and Hall/CRC, 2018.

[69] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, pages 8312–8323, 2018.