2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) | 978-1-6654-0191-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCVW54120.2021.00016

Optical Adversarial Attack

Abhiram Gnanasambandam, Alex M. Sherman, and Stanley H. Chan
Purdue University, West Lafayette, Indiana, USA
{agnanasa,sherma10,stanchan}@purdue.edu

Abstract

*We introduce **OP**tical **AD**versarial attack (OPAD). OPAD is an adversarial attack in the physical space aiming to fool image classifiers without physically touching the objects (e.g., moving or painting the objects). The principle of OPAD is to use structured illumination to alter the appearance of the target objects. The system consists of a low-cost projector, a camera, and a computer. The challenge of the problem is the non-linearity of the radiometric response of the projector and the spatially varying spectral response of the scene. Attacks generated in a conventional approach do not work in this setting unless they are calibrated to compensate for such a projector-camera model. The proposed solution incorporates the projector-camera model into the adversarial attack optimization, where a new attack formulation is derived. Experimental results prove the validity of the solution. It is demonstrated that OPAD can optically attack a real 3D object in the presence of background lighting for white-box, black-box, targeted, and untargeted attacks. Theoretical analysis is presented to quantify the fundamental performance limit of the system.*

1. Introduction

1.1. What is OPAD?

Adversarial attacks and defenses today are predominantly driven by studies in the *digital* space [2, 9, 10, 12, 13, 17, 18, 20, 21, 23, 26, 26, 27, 30] where the attacker manipulates a digital image on a computer. The other form of attacks, which are the *physical* attacks, have been reported in the literature [3, 7, 8, 16, 22, 31–33], but most of the existing ones are invasive in the sense that they need to touch the objects, for example, painting a stop sign [8], wearing a colored shirt [33], or 3D-printing a turtle [1]. In this paper, we present a non-invasive attack using structured illumination. The new attack, called the **OP**tical **AD**versarial attack (OPAD), is based on a low-cost projector-camera system where we project calculated patterns to alter the appearance of the 3D objects.

The difficulty of launching an optical attack is making sure that the perturbations are imperceptible while compensating for the environmental attenuations and the instrument's nonlinearity. An optimal attack pattern in the digital space can become a completely different pattern when illuminated in the real 3D space because of the background lighting, object reflectance, and nonlinear response of the light sources. OPAD overcomes these difficulties by taking into consideration the environment and the algorithm. OPAD is a meta-attack framework that can be applied to any existing digital attack. The uniqueness and novelty of OPAD are summarized in three aspects:

- OPAD is the first method in the literature that explicitly models the instrumentation and the environment. Thus, the adversarial loss function in the OPAD optimization is interpretable and is transparent to the users.
- Most of the illumination-based attacks in the literature require iteratively capturing and optimizing for the attack patterns. OPAD is non-iterative. It attacks real 3D objects in a single shot. Furthermore, it can launch targeted, untargeted, white-box, and black-box attacks.
- OPAD has a theoretical guarantee. With OPAD, we know exactly what objects can be attacked and what cannot. We know the smallest perturbation that is required to compensate for the environmental and instrumental attenuations.

To provide a preview of the proposed OPAD framework, in Figure 1 we show a schematic diagram and four scenarios. The OPAD system consists of a projector and a camera. When the projector is off, the camera sees the unperturbed object. This is the vanilla baseline used by a conventional digital attack. When the projector is turned on, it will project a uniform pattern onto the object. This is the new baseline. Note that this new baseline is required for all-optical perturbations. As long as an active light source exists, a constant offset will be introduced through the uniform illumination. Most classifiers are not affected by such an offset. The interesting phenomenon happens when we project a digital attack pattern (e.g., FGSM [10]). Since we have not compensated for the projector's nonlinearity, the

978-1-6654-0191-3/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCVW54120.2021.00016

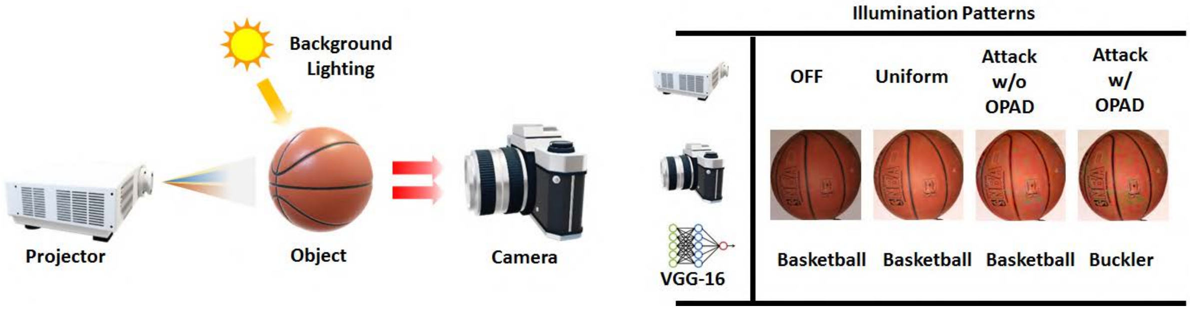

Figure 1. OPAD is a projector-camera system that uses structured illumination to perturb the appearance of objects. The four configurations shown on the right-hand side are: (1) When the projector is off, the classification remains correct. (2) Whenever we turn on the projector, it will generate a uniform illumination. Most classifiers today are robust to this kind of brightness offset. (3) If we illuminate the object with an attack pattern but do not compensate for the environment loss, the classification remains correct. (4) When we use OPAD to compensate for the loss, we successfully attack the classifier to the targeted class.

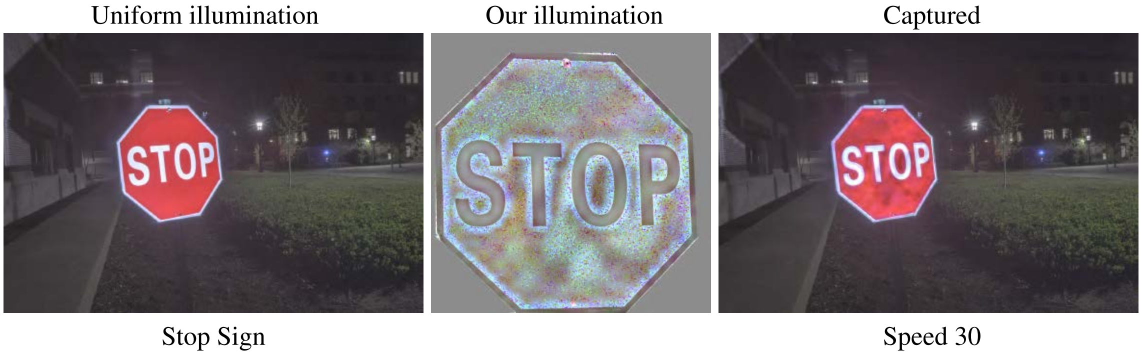

Figure 2. An actual optical setup for OPAD. In this experiment, we attack a real STOP sign. The baseline image is obtained by illuminating the object with a uniform illumination of an intensity 140/255. To attack the object, we generate a projector-compensated illumination with Madry et al. [19] (ℓ_∞ projected gradient descent attack) as the backbone. When projecting this structured illumination onto the metallic stop sign, the prediction becomes Speed 30.

attack will be attenuated. Thus, the object will still be classified correctly. With OPAD, we compensate for the environmental and instrumental distortions. The new perturbation can mislead the classifier. Figure 2 shows our proof-of-concept OPAD system in a real outdoor scene. The system consists of a ViewSonic 3600 Lumens SVGA projector, a Canon T6i camera, and a laptop computer. We showed how OPAD could make the metallic stop sign be classified as Speed 30.

1.2. Related work

The scope of the paper belongs to optics-based attacks. The reported results in the literature are few [6, 24, 25, 32] and there are many limitations.

• **Iterative approaches** [25,32]: These methods do not consider the forward model of the instrument and environment. Thus they need to capture images and calculate the attack iteratively. OPAD is a single-shot attack.

• **Attack a displayed image** [32]: Attacks of this type cannot attack real 3D objects. OPAD can attack 3D objects.

• **Un-targeted attack with unbounded magnitude** [6]: One can use a strong laser to create a beam in the scene. However, such an attack is un-targeted, and the magnitude of the perturbation is practically unbounded. Perturbations of OPAD are targeted, and they are minimally perceptible.

• **Global color correction methods** [24]: Methods based on this principle have limited generality because the optics are spatially varying and spectrally nonlinear. OPAD explicitly takes into account these considerations.

A key component OPAD builds upon is the prior work of Grossberg et al. [11] (and follow-ups [15,28]). Although in a different context, Grossberg et al. demonstrated a principled way to compensate for the loss induced by the projector. OPAD integrates the model with the adversarial attack loss maximization. This new combination of optics and algorithms is new in the literature.

2. Projector-camera model

OPAD is an integration of a projector-camera model and a loss maximization algorithm. In this section, we discuss the projector-camera model.

2.1. Notation

We use $x \in \mathbb{R}^2$ to denote the 2D coordinate of a digital image. The x-th pixel of the source illumination pattern being sent to the projector is denoted as $\boldsymbol{f}(x) = [f_R(x),\ f_G(x),\ f_B(x)]^T \in \mathbb{R}^3$, and the overall source pattern is $\mathbf{f} = [\boldsymbol{f}(x_1), \boldsymbol{f}(x_2), \ldots, \boldsymbol{f}(x_N)]^T \in \mathbb{R}^{3N}$, where N is the number of pixels.

As the source pattern goes through the projector and is reflected by the scene, the actual image captured by the camera is

$$\mathbf{g} = \mathcal{T}(\mathbf{f}), \tag{1}$$

where $\mathbf{g} \in \mathbb{R}^{3N}$ is the observed image, and $\mathcal{T} : \mathbb{R}^{3N} \to \mathbb{R}^{3N}$ is the overall mapping of the forward model. To specify the mapping at pixel x, we denote $\mathcal{T}^{(x)} : \mathbb{R}^3 \to \mathbb{R}^3$ with $\boldsymbol{g}(x) = \mathcal{T}^{(x)}(\boldsymbol{f}(x))$, or simply $\boldsymbol{g}(x) = \mathcal{T}(\boldsymbol{f}(x))$ if the context is clear.

2.2. Radiometric response

As the source pixel $\boldsymbol{f}(x) \in \mathbb{R}^3$ is sent to the projector, the nonlinearity of the projector will alter the intensity per color channel. This is done by a radiometric response function $\mathcal{M} = [\mathcal{M}_R, \mathcal{M}_G, \mathcal{M}_B]^T$ which transforms the desired signal $\boldsymbol{f}(x)$ to a projector brightness signal $\boldsymbol{z}(x) \in R^3$:

$$\boldsymbol{z}(x) \overset{\text{def}}{=} \begin{bmatrix} z_R(x) \\ z_G(x) \\ z_B(x) \end{bmatrix} = \begin{bmatrix} \mathcal{M}_R(f_R(x)) \\ \mathcal{M}_G(f_G(x)) \\ \mathcal{M}_B(f_B(x)) \end{bmatrix} = \mathcal{M}(\boldsymbol{f}(x)).$$

This is illustrated in Figure 3.

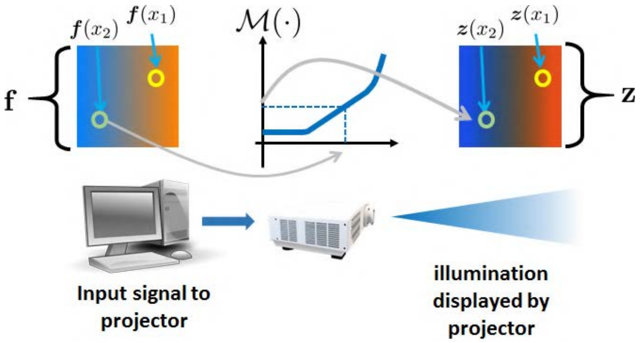

Figure 3. Radiometric response of a projector. As we send an input illumination from the computer to the projector, the input signal $\mathbf{f}$ is altered by the radiometric response function $\mathcal{M}(\cdot)$. $\mathcal{M}(\cdot)$ is performed per color channel per pixel. The actual signal displayed by the projector is $\mathbf{z}$.

Every projector has its own radiometric response. This is intrinsic to the projector, but it is independent of the scene.

2.3. Spectral response

The second component of the camera-projector model is the spectral response due to the irradiance and reflectance of the scene. This converts $\boldsymbol{z}(x)$ to the observed pixel $\boldsymbol{g}(x)$ using a color transform. Since the spectral response encodes the scene, and the color transform is spatially varying,

$$\boldsymbol{g}(x) = \boldsymbol{V}^{(x)}\boldsymbol{z}(x) + \boldsymbol{b}^{(x)}, \tag{2}$$

where $\boldsymbol{V}^{(x)}$ is a 3×3 color mixing matrix defined as

$$\boldsymbol{V}^{(x)} = \begin{bmatrix} V_{RR}^{(x)} & V_{RG}^{(x)} & V_{RB}^{(x)} \\ V_{GR}^{(x)} & V_{GG}^{(x)} & V_{GB}^{(x)} \\ V_{BR}^{(x)} & V_{BG}^{(x)} & V_{BB}^{(x)} \end{bmatrix}. \tag{3}$$

Here, the superscript $(\cdot)^{(x)}$ emphasizes the spatially varying nature of the matrix $\boldsymbol{V}^{(x)}$, whereas the subscript clarifies the color mixing process from one input color to another output color. The vector $\boldsymbol{b}^{(x)}$ is an offset accounting for background illumination. It is defined as $\boldsymbol{b}^{(x)} = [b_R^{(x)}; b_G^{(x)}; b_B^{(x)}] \in \mathbb{R}^3$. A schematic diagram illustrating the spectral response is shown in Figure 4.

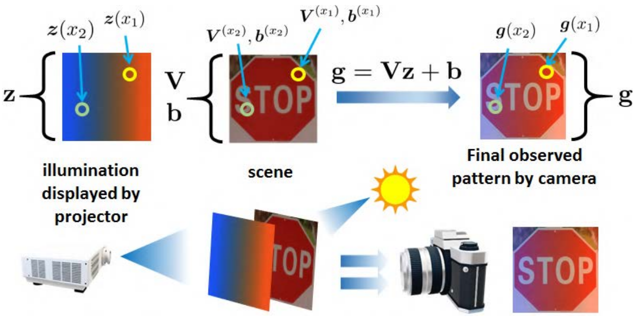
$\boldsymbol{z}(x_2)$
$\boldsymbol{z}(x_1)$
$\boldsymbol{V}^{(x_2)}, \boldsymbol{b}^{(x_2)}$
$\boldsymbol{V}^{(x_1)}, \boldsymbol{b}^{(x_1)}$
$\mathbf{z}$
$\mathbf{V}$
$\mathbf{b}$
$\mathbf{g} = \mathbf{Vz} + \mathbf{b}$
$\boldsymbol{g}(x_2)$
$\boldsymbol{g}(x_1)$
$\mathbf{g}$
STOP
illumination displayed by projector
scene
Final observed pattern by camera

Figure 4. Spectral response of a projector. Given the illumination pattern displayed by the projector, the scene and background lighting will be applied to the illumination via a color transform matrix $\boldsymbol{V}^{(x)}$ and an offset vector $\boldsymbol{b}^{(x)}$. The final image captured by the camera follows (2).

If the input illumination is $\mathbf{f}$, the final output observed by the camera is

$$\mathbf{g} = \underbrace{\mathbf{V}\mathcal{M}(\mathbf{f}) + \mathbf{b}}_{\mathcal{T}(\mathbf{f})}, \tag{4}$$

where $\mathbf{V} = \text{diag}\{\boldsymbol{V}^{(x_1)}, \boldsymbol{V}^{(x_2)}, \ldots, \boldsymbol{V}^{(x_N)}\} \in \mathbb{R}^{3N \times 3N}$ is a block diagonal matrix where each block $\boldsymbol{V}^{(x_n)}$ is a 3-by-3 matrix. The mapping $\mathcal{M}$ is an elementwise transform representing the radiometric response. The vector $\mathbf{b} = [\boldsymbol{b}^{(x_1)}, \ldots, \boldsymbol{b}^{(x_N)}]^T \in \mathbb{R}^{3N}$ is the overall offset.

The estimation of both the radiometric and the spectral response is discussed in the supplementary material.

3. OPAD Algorithm

OPAD is a meta procedure that can be applied to any existing adversarial loss maximization. Because the radiometric and spectral response of the projector-camera system treats the illumination and the scene in two different ways, the loss maximization in OPAD is also different from a conventional attack as illustrated in Figure 5.

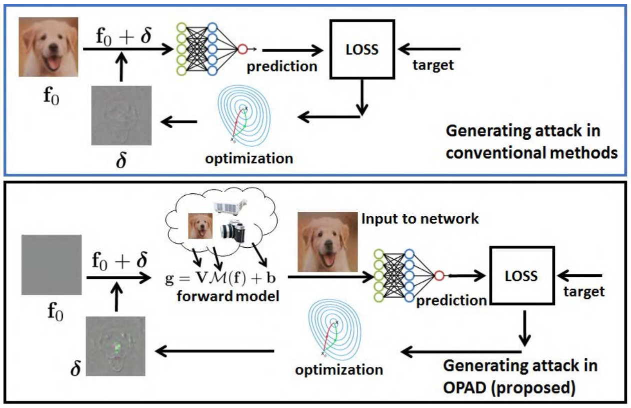
$\mathrm{f}_0 + \boldsymbol{\delta}$
prediction
LOSS
target
f_0
$\boldsymbol{\delta}$
optimization
Generating attack in conventional methods
$\mathrm{g} = \mathbf{V}\mathcal{M}(\mathrm{f}) + \mathrm{b}$
forward model
Input to network
$\mathrm{f}_0 + \boldsymbol{\delta}$
f_0
prediction
LOSS
target
$\boldsymbol{\delta}$
optimization
Generating attack in OPAD (proposed)

Figure 5. In conventional digital attack, the perturbation is directly added to the input image by computing the gradient of the network. In OPAD, the base input $\mathbf{f}_0$ is the uniform illumination. Perturbation $\boldsymbol{\delta}$ is added to $\mathbf{f}_0$. The projector and the scene kick in through the radiometric response and the spectral response, respectively.

3.1. OPAD loss maximization

For simplicity we formulate the targeted white-box attack. Other forms of attacks (black-box, and/or untargeted) can be derived similarly. Consider a uniform illumination pattern $\mathbf{f}_0$ that gives a clean image $\mathbf{g}_0 = \mathbf{V}\mathcal{M}(\mathbf{f}_0) + \mathbf{b}$. Our goal is to make the classifier think that the label is ℓ_{target}. The white-box attack is given by

$$\begin{aligned} \boldsymbol{\delta} &= \underset{\boldsymbol{\delta}}{\text{argmax}} \quad \mathcal{L}(\mathcal{T}(\mathbf{f}_0 + \boldsymbol{\delta}), \ell_{\text{target}}) \\ &= \underset{\boldsymbol{\delta}}{\text{argmax}} \quad \mathcal{L}(\mathbf{V}\mathcal{M}(\mathbf{f}_0 + \boldsymbol{\delta}) + \mathbf{b}, \ell_{\text{target}}). \end{aligned} \tag{5}$$

In most of the attack methods, the attack $\boldsymbol{\delta}$ is constrained in the *input* space through an ϵ-ball such as $\|\boldsymbol{\delta}\| < \epsilon$. This only ensures that the input is similar before and after the attack. In our problem, we are interested in two constraints:

- The attack in the *output* space should have a small magnitude so that the displayed images before and after attack are visually similar. That is, we want

$$\Bigg\| \underbrace{(\mathbf{V}\mathcal{M}(\mathbf{f}_0) + \mathbf{b})}_{\overset{\text{def}}{=}\mathbf{g}_0} - \underbrace{(\mathbf{V}\mathcal{M}(\mathbf{f}_0 + \boldsymbol{\delta}) + \mathbf{b})}_{\overset{\text{def}}{=}\mathbf{g}} \Bigg\| < \alpha, \tag{6}$$

  for some upper bound constant α.

- The perturbed illumination has to be physically achievable, meaning that

$$0 \le \mathbf{f}_0 + \boldsymbol{\delta} \le 1. \tag{7}$$

Putting these constraints into the formulation, the attack is obtained by solving the optimization

$$\begin{aligned} \boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\text{argmax}} \quad & \mathcal{L}(\mathbf{V}\mathcal{M}(\mathbf{f}_0 + \boldsymbol{\delta}) + \mathbf{b}, \ell_{\text{target}}) \\ \text{subject to} \quad & \\ & \| (\mathbf{V}\mathcal{M}(\mathbf{f}_0) + \mathbf{b}) - (\mathbf{V}\mathcal{M}(\mathbf{f}_0 + \boldsymbol{\delta}) + \mathbf{b}) \| < \alpha. \\ & 0 \le \mathbf{f}_0 + \boldsymbol{\delta} \le 1. \end{aligned} \tag{P1}$$

3.2. Simplifying the formulation

Solving (P1) is challenging because it involves computing $\mathbf{V}$ and $\mathcal{M}$ in a nonlinear way. However, it is possible to simplify the problem. Using $\mathbf{g}$ and $\mathbf{g}_0$ defined in the first constraint, we can define a perturbation $\boldsymbol{\eta}$ in the *output* space as

$$\boldsymbol{\eta} \overset{\text{def}}{=} \mathbf{g} - \mathbf{g}_0. \tag{8}$$

Substituting this into (P1), we can rewrite the problem as

$$\begin{aligned} \boldsymbol{\eta}^* = \underset{\boldsymbol{\eta}}{\text{argmax}} \quad & \mathcal{L}(\mathbf{g}_0 + \boldsymbol{\eta}, \ell_{\text{target}}) \\ \text{subject to} \quad & \|\boldsymbol{\eta}\| < \alpha, \\ & 0 \le \mathbf{f}_0 + \boldsymbol{\delta} \le 1. \end{aligned} \tag{9}$$

Thus, it remains to rewrite the second constraint. To this end, we notice that if we apply $\mathcal{M}$ and $\mathbf{V}$ to $\mathbf{f}_0 + \boldsymbol{\delta}$, we will alter the box constraint $0 \le \mathbf{f}_0 + \boldsymbol{\delta} \le 1$ (which is equivalent to $\boldsymbol{c}_\ell \le \boldsymbol{\delta} \le \boldsymbol{c}_u$ where $\boldsymbol{c}_\ell = -\mathbf{f}_0$ and $\boldsymbol{c}_u = 1 - \mathbf{f}_0$) to a new constraint set:

$$\Omega = \left\{ \boldsymbol{\eta} \;\middle|\; \boldsymbol{\eta} = \mathbf{V}\mathcal{M}(\mathbf{f}_0 + \boldsymbol{\delta}) - \mathbf{V}\mathcal{M}(\mathbf{f}_0), \;\; \boldsymbol{c}_\ell \le \boldsymbol{\delta} \le \boldsymbol{c}_u \right\}.$$

As we will derive below, this constraint is met by projecting the current estimate onto Ω. Putting everything together, we arrive at the final attack formulation:

$$\begin{aligned} \boldsymbol{\eta}^* = \underset{\boldsymbol{\eta} \in \Omega}{\text{argmax}} \quad & \mathcal{L}(\mathbf{g}_0 + \boldsymbol{\eta}, \ell_{\text{target}}) \\ \text{subject to} \quad & \|\boldsymbol{\eta}\| < \alpha. \end{aligned} \tag{P2}$$

3.3. OPAD procedure

If we ignore the constraint set Ω for a moment, (P2) is a standard attack optimization that can be solved in various ways, e.g., fast gradient sign method (FGSM) [10], projected gradient descent (PGD) [19], and many others [2, 4, 14, 16]. The per-iteration update of these algorithms can be written in a generic form as

$$\boldsymbol{\eta}^{t+1} = \text{my attack}(\mathbf{f_0}, \boldsymbol{\eta}^t, \ell), \tag{10}$$

where 'my attack$(\cdot)$' can be chosen as any of the attacks listed above. For example if we use PGD with ℓ_∞ constraint, then $\boldsymbol{\eta}^{t+1} = \alpha \cdot \text{sign}\{\nabla \mathcal{L}(\boldsymbol{g}_0 + \boldsymbol{\eta}^t, \ell_{\text{target}})\}$.

In the presence of the constraint set Ω, the per-iteration update will involve a projection:

$$\boldsymbol{\eta}^{t+1} = \text{Project}_{\Omega}\left\{\underbrace{\text{my attack}(\mathbf{f_0}, \boldsymbol{\eta}^{t}, \ell_{\text{target}})}_{=\boldsymbol{\eta}^{t+\frac{1}{2}}}\right\}. \tag{11}$$

Specific to our problem, the projection operation inverts the current estimate from the output space to the input space and do the clipping in the input space. Then, we re-map the signal back to the output space. Mathematically, the projection is defined as

$$\text{Project}_{\Omega}(\boldsymbol{\eta}^{t+\frac{1}{2}}) = \mathcal{T}\left(\left[\mathcal{T}^{-1}\left(\mathbf{g}_0 + \boldsymbol{\eta}^{t+\frac{1}{2}}\right)\right]_{[0,1]}\right) - \mathbf{g_0}, \tag{12}$$

where $\mathcal{T}$ is the forward mapping defined in (4) and $[\,\cdot\,]_{[0,1]}$ means clipping the signal to $[0, 1]$.

For implementation, the overall attack is estimated by first running an off-the-shelf adversarial attack for one iteration. Then we use the pre-computed projector-camera model $\mathcal{T}$ (and its inverse $\mathcal{T}^{-1}$) to handle the constraint set Ω. Since $\mathbf{V}$ is just a matrix vector multiplication, and $\mathcal{M}$ is a pixel-wise mapping (which can be stored as a look-up table), the overall computational cost is comparable to the original attack.

4. Understanding the geometry of OPAD

We analyze the fundamental limit of OPAD by considering linear classifiers. Consider a binary classification problem with a true label $\ell_{\text{true}} \in \{+1, -1\}$. We assume that the classifier $h : \mathbb{R}^{3N} \to \{+1, -1\}$ is linear, so that the prediction is given by

$$\widehat{\ell}_{\text{predict}} = h(\mathbf{g}_0) = \text{sign}(\boldsymbol{\theta}^T \mathbf{g}_0), \tag{13}$$

where $\boldsymbol{\theta}$ is the classifier's parameter, and $\mathbf{g}_0 = \mathbf{V}\mathcal{M}(\mathbf{f}_0) + \mathbf{b}$ is the clean image generated by the lower pipeline of Figure 5. The loss function $\mathcal{L}_{\boldsymbol{\theta}}(\cdot)$ for this sample $\mathbf{g}_0$ is

$$\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{g}_0, \ell_{\text{true}}) = -\ell_{\text{true}} \cdot \boldsymbol{\theta}^T \mathbf{g}_0 \tag{14}$$

Suppose that we attack the classifier by defining $\mathbf{g} = \mathbf{g}_0 + \boldsymbol{\eta}$. Then, the loss function becomes

$$\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{g}, \ell_{\text{target}}) = -\ell_{\text{target}} \cdot \boldsymbol{\theta}^T (\mathbf{g}_0 + \boldsymbol{\eta}). \tag{15}$$

Substituting (15) into (P2), we show that

$$\begin{aligned} \boldsymbol{\eta}^* = \underset{\boldsymbol{\eta}}{\text{argmax}} \quad & -\ell_{\text{target}} \cdot \boldsymbol{\theta}^T \boldsymbol{\eta}, \\ \text{subject to} \quad & \boldsymbol{\eta} \in \Omega, \quad \underbrace{\|\boldsymbol{\eta}\| \le \alpha}_{\Psi \overset{\text{def}}{=} \{\boldsymbol{\eta} \mid \|\boldsymbol{\eta}\| < \alpha\}} . \end{aligned} \tag{16}$$

Therefore, to analyze OPAD, we just need to understand the sets Ω, Ψ, and the parameter $\boldsymbol{\theta}$.

4.1. Geometry of the constraints

There are two constraints in (16). The first constraint $\boldsymbol{\eta} \in \Psi$ is a simple α-ball surrounding the input. It says that the perturbation in the camera space should be bounded.

The more interesting constraint is $\boldsymbol{\eta} \in \Omega$. To understand how Ω contributes to the feasibility of OPAD, we consider one pixel location x_1 of the object. This pixel has three colors $\boldsymbol{f}(x_1) = [f_R(x_1), f_G(x_1), f_B(x_1)]^T$. Since $\boldsymbol{f}(x_1)$ is the signal we send to the projector, it holds that $0 \le \boldsymbol{f}(x_1) \le 1$ as illustrated in Figure 6.

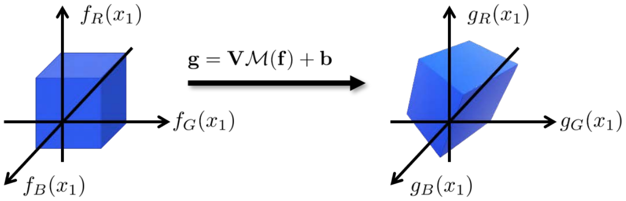
$f_R(x_1)$
$f_G(x_1)$
$f_B(x_1)$
$\mathbf{g} = \mathbf{V}\mathcal{M}(\mathbf{f}) + \mathbf{b}$
$g_R(x_1)$
$g_G(x_1)$
$g_B(x_1)$

Figure 6. The constraint Ω is constructed by converting a box constraint $0 \le \boldsymbol{f}(x_1) \le 1$ to a rectangle via a per-pixel per-color transformation $\mathcal{M}$, followed by a color mixing process.

After passing through the projector-camera model, the observed pixel by the camera is $\boldsymbol{g}(x_1)$. The relationship between $\boldsymbol{g}(x_1)$ and $\boldsymbol{f}(x_1)$ is given by (4) (for pixel x_1). The conversion from $\boldsymbol{f}(x_1)$ to $\boldsymbol{g}(x_1)$ involves $\mathcal{M}$ which is a pixel-wise and color-wise mapping. Since $\boldsymbol{f}(x_1)$ lives in a unit cube, $\mathcal{M}(\boldsymbol{f}(x_1))$ will live in a rectangular box. The second conversion from $\mathcal{M}(\boldsymbol{f}(x_1))$ to $\boldsymbol{g}(x_1)$ involves an affine transformation. Therefore, the resulting set that contains $\boldsymbol{g}(x_1)$ will be a 3D polygon. This 3D polygon is Ω.

4.2. When will OPAD fail?

The feasibility of OPAD is determined by $\Omega \cap \Psi$ and the decision boundary $\boldsymbol{\theta}$, as illustrated in Figure 7. Given a clean classifier $\boldsymbol{\theta}$, we partition the space into two half spaces (Class 1 and Class 0). The correct class is Class 1. To move the classification from Class 1 to Class 0, we must search along the feasible direction where Ψ and Ω intersects.

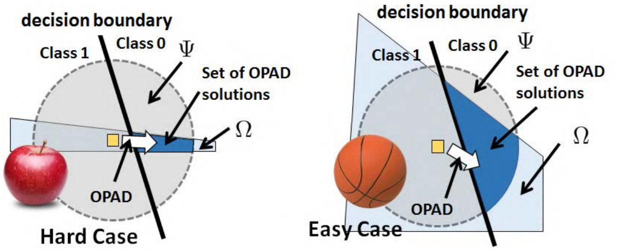

Figure 7. (a) OPAD is hard when the color transformation $\mathbf{V}$ is nearly singular. This happens when the object has saturated pixels or is reflective. (b) OPAD is easy when Ω covers a large portion of the target class. This happens when $\mathbf{V}$ is invertible, e.g., for fabric, textile, rough surfaces etc.

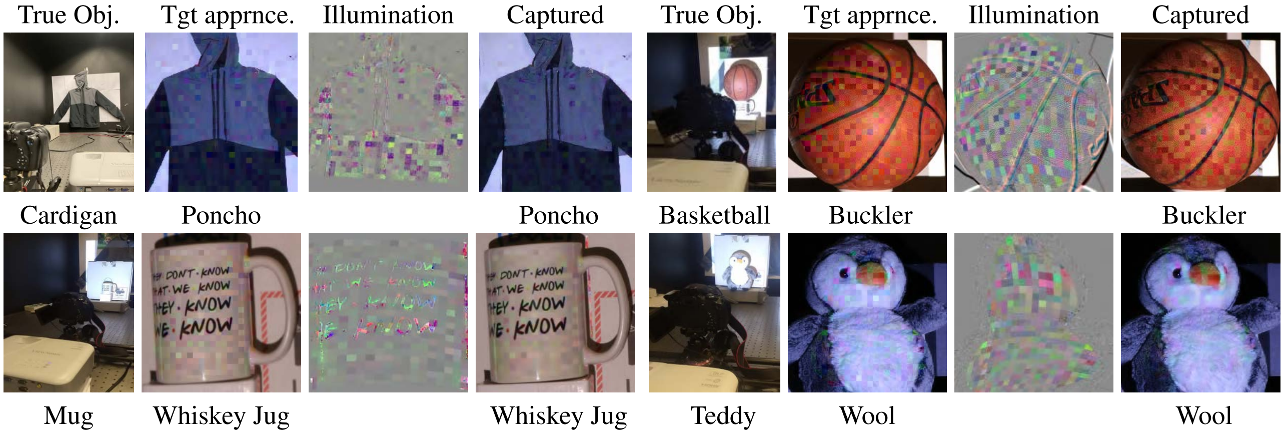

Image	Algorithm	Generated on	Tested on	Block Size	Step-size	Norm. ℓ_2 Dist.
Cardigan	PGD (ℓ_∞)	VGG-16	VGG-16	8×8	0.05	5.8/255
Basketball	PGD (ℓ_∞)	Resnet-50	Resnet-50	8×8	0.05	3.2/255
Coffee Mug	PGD (ℓ_2)	VGG-16	VGG-16	8×8	0.5	4.1/255
Teddy	PGD (ℓ_2)	Resnet-50	Resnet-50	8×8	0.5	4.3/255

Figure 8. OPAD on real 3D objects. In each example, we show the targeted appearance (tgt apprnce) which is how the un-compensated attack should look like on a digital computer. The illumination is the OPAD illumination, and the capture is what the camera sees. Normalized ℓ_2 distance measures the average ℓ_2 difference between the original image and the captured image.

The geometry above shows that the success/failure of OPAD is object dependent. Some objects are easier to attack, because the $\mathbf{V}$ matrix and the $\mathbf{b}$ vector create a "bigger" Ω. This happens when the object surface is responsive to the illumination, e.g. the basketball shown in Figure 7. The hard cases happen $\mathbf{V}$ and $\mathbf{b}$ create a very "narrow" Ω. For example, a bright red color shirt is difficult because its red pixel is too strong. An apple is difficult because it reflects the light. Note that this is an intrinsic problem of the optics, and not the problem of the algorithm.

4.3. Can we make OPAD unnoticeable?

The short answer is no. Unlike digital attacks where the average ℓ_2 (or ℓ_∞) distance is driven by the decision boundary, the minimum amount of perturbation in OPAD is driven by the decision boundary *and* the optics. The perturbation has to go through the radiometric response of the projector and the spectral response of the scene, not to mention other optical limits such as diffraction and out-of-focus. For tough surfaces, if the feasible set Ω is small, we have no choice but to increase the perturbation strength.

The conclusion of OPAD may appear pessimistic, because there are many objects we cannot attack. However, on the positive side OPAD suggests ways to *defend* optical attacks. For example, one can configure the environment such that certain key features of the object are close to being saturated. Constantly illuminating the object with pre-defined patterns will also help defending against attacks. These are interesting topics for future research.

5. Experiments

We report experimental results on real 3D objects. For the results in the main paper, we mainly use white-box projected gradient descents (PGD) [19], and fast gradient sign method (FGSM) [10] to attack the objects. Additional results using black-box and other attack algorithms (e.g. colorization [2]) are reported in the supplementary material.

5.1. Quantitative evaluation

We first conduct a quantitative experiment on four real 3D objects (teddy, cardigan, basketball, and mug) as illustrated in Figure 8. For each object, we generate 16 different targeted attacks: 4 different target classes (poncho, buckler, whiskey jug and wool), 2 different constraints (ℓ_2 and ℓ_∞), and 2 different classifiers (VGG-16, and Resnet-50). The PGD [19] is used for all the 64 attacks. The parameter α was set to be 0.05 for ℓ_∞ constraints and 0.5 for ℓ_2 constraints. We use the same gradient for each 8×8 pixels. 20 iterations were used for generating each attack.

The result in Table 1 indicates that OPAD worked for 31 times out of the total 64 attacks (48%). While this may not appear as a high success rate, the result is valid. The reason is that the success rate depends the specific types of objects being attacked. For example, the cardigan and the ball are easier to attack, but the teddy and the mug are difficult to attack. This is consistent with our theoretical analysis in Figure 7, where certain objects have a small feasible set Ω due to the poor spectral response. We emphasize that this is the limitation of the optics, and not the attack optimization.

Tar./Obj.	Cardigan	Teddy	Ball	Mug
Poncho	4/4, 0.85	0/4, 0.13	2/4, 0.46	0/4, 0.10
Wool	4/4, 0.74	3/4, 0.22	1/4, 0.27	0/4, 0.14
Buckler	4/4, 0.48	0/4, 0.08	3/4, 0.72	1/4, 0.18
Jug	4/4, 0.65	0/4, 0.05	2/4, 0.51	3/4, 0.34

Table 1. Success rate and confidence for the targeted attack experiments. Notice that the success of the attack depends on the object being attacked. While Cardigan and ball are easier to attack, teddy and the mug are not.

In Table 2, we compare the influence due to the classification network and perturbation ℓ_p-norm constraints. The results indicate that VGG-16 is easier to attack than ResNet-50, and ℓ_2 is slightly easier to attack than ℓ_∞. However, the difference is not significant. We believe that the optics plays a bigger role here.

Network		Constraint	
VGG-16	Resnet-50	ℓ_2	ℓ_∞
17/32, 0.43	14/32, 0.31	16/32, 0.40	15/32, 0.34

Table 2. Success rate and confidence for targeted attack using different networks and different constraints.

5.2. Need for projector-camera compensation

This experiment aims to verify the necessity of the projector-camera compensation step in OPAD. To reach a conclusion, we consider another projector-based attack method proposed by Nguyen et al. [24]. It is a *single-capture* optical attack like our proposed method. Its projector-camera compensation consists of a simple global color correction without taking the spatially varying color mixing matrix $\mathbf{V}$ into consideration.

We attack a VGG-16 network using PGD, with $\alpha = 1$. Both the algorithms were run for 20 iterations. We attacked the object 'book' by targeting 15 random classes from the ImageNet dataset [5]. Table 3 summarizes the quantitative evaluation results (with details in the supplementary materials). OPAD worked 10/15 times, while [24] worked only 3/15 times.

From "book" to one of the 15 target classes							
1	2	3	4	5	6	7	8
✓/×	✓/×	×/×	✓/✓	✓/×	✓/×	✓/×	×/×
9	10	11	12	13	14	15	
✓/✓	×/×	✓/×	×/×	×/×	✓/✓	✓/×	

Table 3. Comparing (with OPAD / with [24]) when attacking a real 3D book to 15 target classes. Out of the 15 attacks, an attack with OPAD succeeds in 10/15 time whereas an attack without OPAD succeeds in only 3/15 times.

5.3. How strong should OPAD be?

A natural question following Table 3 is the minimum perturbation strength that is needed for OPAD. As discussed in Section 4.3, OPAD is fundamentally limited by the optics and the decision boundary. Unlike digital attacks where the ℓ_p ball can be made very small, OPAD attack has to be reasonably strong to compensate for the optical loss. In Figure 9, we conduct an experiment to understand how imperceptible OPAD can be. We want to turn a "book" into a "comic-book" or a "pretzel". For the both targets, we launch 4 attacks using $\alpha \in \{0.1, 0.5, 1.0, 1.5\}$. We see that a smaller α is sufficient for "comic-book" and a larger α is needed for "pretzel". In both cases, the perturbation is not too strong but still visible.

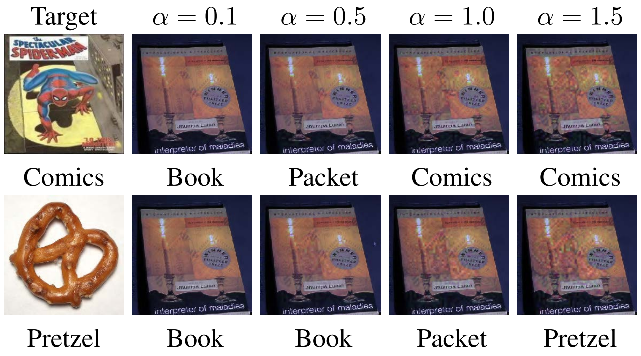

Figure 9. Unlike digital attacks where the perturbation is determined by the ℓ_p-ball, OPAD needs to overcome the optics. The experiment here shows the amount of perturbation required to turn a "book" to a "comic-book" and a "pretzel".

5.4. Significance of the constraint Ω

In this experiment we shift our attention to the constraint Ω because it is this constraint Ω that makes our problem special. We ask: *what happens if we ignore the constraint $\boldsymbol{\eta} \in \Omega$ in the optimization?* The short answer is that we will generate illumination patterns that are not feasible. To justify this claim, we conduct an experiment by launching a white-box FGSM attack on VGG-16 for a real 3D cardigan shown in Figure 10. The result shows that if we ignore the constraint, FGSM will generate a pattern that contains colors that are not achievable. In contrast, when Ω is included, the optimization solution will be optically feasible.

5.5. Robustness against perspective and ISO

Our final experiment concerns about the robustness of OPAD against translation, zoom, and varying ISO settings. This is important because OPAD can potentially be used to fool a neighboring camera and not just the OPAD camera. We conduct two experiments using a real metallic STOP sign. The classifier is trained based on [8], using German Traffic Sign Recognition Benchmark dataset [29].

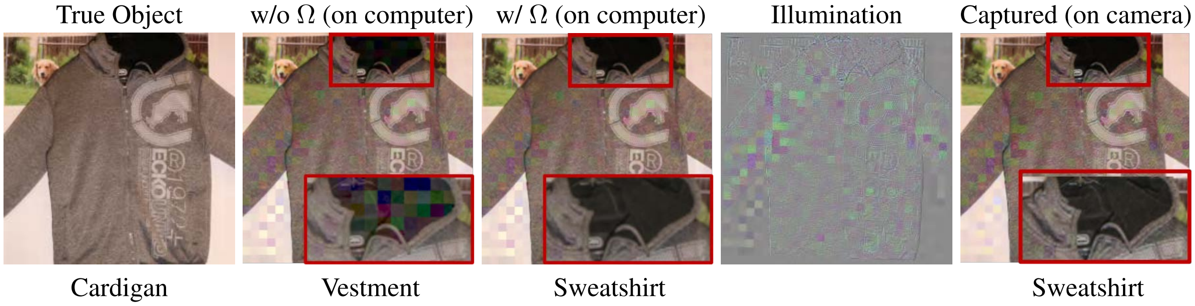

Figure 10. Significance of the constraint $\boldsymbol{\eta} \in \Omega$. Notice that running the optimization without Ω, we will generate an image that may not be optically achievable. The inset images are displayed with MATLAB's tonemap function.

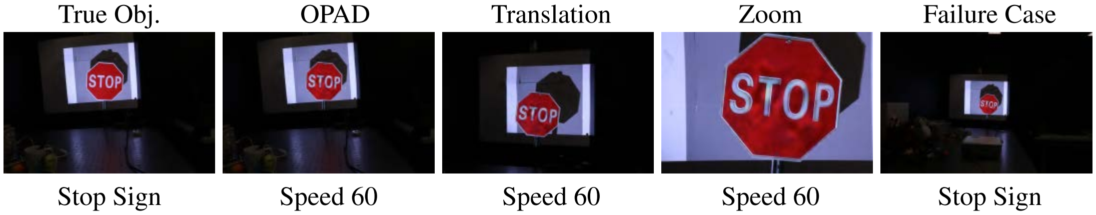

Figure 11. Robustness of OPAD against perspective change (using a real metallic STOP sign). As we translate the camera, the same OPAD illumination is still capable of attacking. The failure happens when the camera zooms out too much.

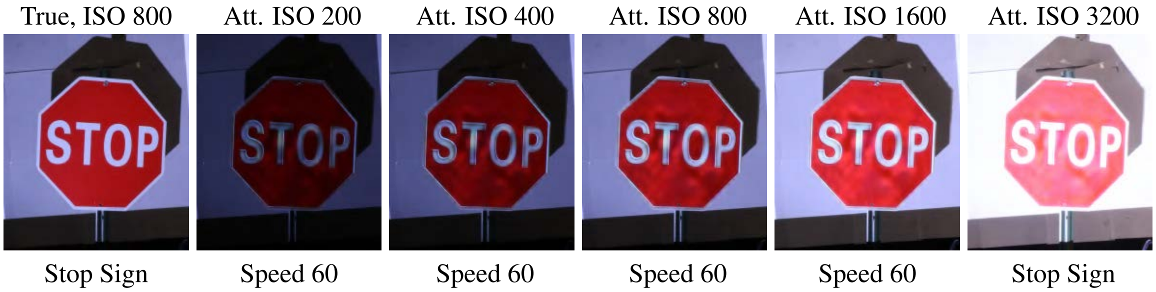

Figure 12. Robustness of OPAD against different ISO of the camera (using a real metallic STOP sign). The OPAD attack is computed for ISO 800, but the images are captured at other ISO levels. The failure happens when the ISO is too high so that many pixels are saturated.

In Figure 11, we capture the scene with different camera location and zooms. We first generate a successful attack on the STOP sign, which is classified as 'Speed limit 60'. The camera is then translated by 30° w.r.t the position of the STOP sign. We also capture the scene with zoom in and zoom out. The result shows that the OPAD still works until the object is zoomed out for a long distance.

In Figure 12, we adjust the camera with different ISO settings. The attack is generated using an ISO setting of 800. As the ISO changes in the range $[200, 400, 800, 1600, 3200]$, OPAD remains robust except for 3200 ISO where a lot of pixels are saturated.

6. Discussions and conclusion

While this paper focuses exclusively on demonstrating how to *attack*, OPAD has the potential to address the critical need in robust machine learning today where we do not have a way to model the environment. OPAD provides a parametric model where the parameters are controlled through the hardware. If we want to mimic an environment, we can adjust the OPAD parameters until the scene is reproduced. Consequently, we can analyze the robustness of the classifiers and defense techniques. We note that none of the existing optical attacks has this potential.

OPAD is a non-invasive adversarial attack based on structured illumination. For a variety of existing attack methodologies (targeted, untargeted, white-box, black-box, FGSM, PGD, and colorization), OPAD can transform the known digital results into real 3D objects. The feasibility of OPAD is constrained by the surface material of the object and the saturation of color. The success of OPAD demonstrates the possibility of using an optical system to alter faces or for long-range surveillance tasks. It would be interesting to see how these can be realized in the future.

7. Acknowledgement

The work is funded partially by the Army Research Office under the contract W911NF-20-1-0179 and the National Science Foundation under grant ECCS-2030570.

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*, 2018. 1

[2] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. Unrestricted adversarial examples via semantic manipulation. *ICLR*, 2020. 1, 4, 6

[3] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*, 2019. 1

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symp. Security and Privacy*, pages 39–57, 2017. 4

[5] 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. 7

[6] Ranjie Duan, Xiaofeng Mao, AK Qin, Yun Yang, Yuefeng Chen, Shaokai Ye, and Yuan He. Adversarial laser beam: Effective physical-world attack to DNNs in a blink. *arXiv preprint arXiv:2103.06504*, 2021. 2

[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. *arXiv preprint arXiv:1807.07769*, 2018. 1

[8] 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 *CVPR*, 2018. 1, 7

[9] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017. 1

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015. 1, 4, 6

[11] Michael D Grossberg, Harish Peri, Shree K Nayar, and Peter N Belhumeur. Making one object look like another: Controlling appearance using a projector-camera system. In *CVPR*, 2004. 2

[12] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017. 1

[13] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[15] Bingyao Huang, Tao Sun, and Haibin Ling. Compennet++: End-to-end full projector compensation. *ICCV*, 2019. 2

[16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1, 4

[17] Bo Li and Yevgeniy Vorobeychik. Feature cross-substitution in adversarial classification. In *NeurIPS*, 2014. 1

[18] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *ICCV*, 2017. 1

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018. 2, 4, 6

[20] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *ICLR*, 2017. 1

[21] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017. 1

[22] Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, and Yuval Weisglass. Fooling a real car with adversarial traffic signs. *arXiv preprint arXiv:1907.00374*, 2019. 1

[23] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015. 1

[24] Dinh-Luan Nguyen, Sunpreet S Arora, Yuhang Wu, and Hao Yang. Adversarial light projection attacks on face recognition systems: A feasibility study. In *CVPR Workshops*, pages 814–815, 2020. 2, 7

[25] Nicole Nichols and Robert Jasper. Projecting trouble: Light based adversarial attacks on deep learning classifiers. 2018. 2

[26] 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. 1

[27] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. *ICLR*, 2016. 1

[28] Christian Siegl, Matteo Colaianni, Marc Stamminger, and Frank Bauer. Adaptive stray-light compensation in dynamic multi-projection mapping. *Comp. Visual Media*, 3(3):263–271, 2017. 2

[29] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012. 7

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICLR*, 2014. 1

[31] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *CVPR*, 2020. 1

[32] Nils Worzyk, Hendrik Kahlen, and Oliver Kramer. Physical adversarial attacks by projecting perturbations. In *Intl. Conf. Artificial Neural Networks*. Springer, 2019. 1, 2

[33] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! Evading person detectors in a physical world. *ECCV*, 2020. 1