SLAP: Improving Physical Adversarial Examples with Short-Lived Adversarial Perturbations

Giulio Lovisotto  
*University of Oxford*

Henry Turner  
*University of Oxford*

Ivo Sluganovic  
*University of Oxford*

Martin Strohmeier  
*armasuisse*

Ivan Martinovic  
*University of Oxford*

Abstract

Whilst significant research effort into adversarial examples (AE) has emerged in recent years, the main vector to realize these attacks in the real-world currently relies on static adversarial patches, which are limited in their conspicuousness and can not be modified once deployed.

In this paper, we propose Short-Lived Adversarial Perturbations (SLAP), a novel technique that allows adversaries to realize robust, dynamic real-world AE from a distance. As we show in this paper, such attacks can be achieved using a light projector to shine a specifically crafted adversarial image in order to perturb real-world objects and transform them into AE. This allows the adversary greater control over the attack compared to adversarial patches: (i) projections can be dynamically turned on and off or modified at will, (ii) projections do not suffer from the locality constraint imposed by patches, making them harder to detect.

We study the feasibility of SLAP in the self-driving scenario, targeting both object detector and traffic sign recognition tasks. We demonstrate that the proposed method generates AE that are robust to different environmental conditions for several networks and lighting conditions: we successfully cause misclassifications of state-of-the-art networks such as Yolov3 and Mask-RCNN with us to 98% success rate for a variety of angles and distances. Additionally, we demonstrate that AE generated with SLAP can bypass SentiNet, a recent AE detection method which relies on the fact that adversarial patches generate highly salient and localized areas in the input images.

1 Introduction

Recent advances in computational capabilities and machine learning algorithms have lead to deep neural networks (DNN) rapidly becoming the dominant choice for a wide range of computer vision tasks. Due to their performance, DNNs are increasingly being used in security-critical contexts, such as biometric authentication or object recognition for autonomous driving. However, if a malicious actor controls the input of the network, it is a well known issue that DNNs are susceptible to carefully crafted adversarial examples (AE) [36], which leverage specific directions in input space to create examples which may resemble legitimate images, but cause misclassification at test time.

Significant body of earlier research focused on analyzing AE in the digital domain, where it is assumed that the adversary has the capability of making pixel-specific manipulations in the input. This concept has further been brought into practice with the realization of physically robust AE [10, 13, 14, 34, 38], which are examples that survive a set of real-world environmental conditions, such as different viewing distances or angles. In order to realize the AE, these works either print patches (e.g. as stickers or in the form of glasses in case of face recognition), or replace an entire object by printing it with subtle differences (e.g., replace a stop sign with an adversarial poster of a stop sign). However, the use of patches to realize AE has multiple limitations. Firstly, such patches usually generate highly salient areas in the network inputs, which makes them detectable by recent countermeasures [11]. Secondly, in the autonomous driving scenario, sticking patches on a traffic sign leads to continuous misdetection of such signs, which is equivalent to removing the sign from the road or covering it. Lastly, once placed, these patches can not be modified and are easily spotted by every passerby.
In this paper, we focus on the autonomous driving scenario and propose using a light projector to achieve Short-Lived Adversarial Perturbations (SLAPs), a novel AE approach that allows adversaries to realize robust, dynamic real-world AE from a distance. This class of AE techniques provides the attacker with multiple benefits over existing patch-based methods and evade detection by existing countermeasures. Firstly, an adversary is free to control when to turn on and off a certain projection or can even modify the projected image, allowing a specific perturbation to be arbitrarily short-lived. This means that adversaries can (i) target specific vehicles by using the projector when the target vehicle is approaching a sign and (ii) able to leave no traces of a malicious attack. Secondly, a projection is more dynamic compared to a patch, meaning that instead of crafting a patch that works in many different conditions (e.g., different ambient light, different angles) the adversary can dynamically change the projection based on the current situation. Additionally, in black-box scenarios, when adversaries do not know the detection system operating on a car, they can use a set of pre-computed projections or make different adjustments until they find a successful projection that leads to misclassifications.

As part of designing the SLAP attack, we propose a method to quickly and accurately model the effect of projections under certain environmental conditions, by analyzing the absolute changes in pixel colors captured by an RGB camera as different projections are being shown. The method consists of fitting a differentiable model that allows us to propagate the derivatives of the projection through such model during the AE crafting phase. We also improve the robustness of AE in the physical world by systematically identifying and accounting for several varying environmental conditions, which were limited to viewing angles and distance in previous works [13, 15, 38]. We empirically analyze the feasibility of SLAP on two different use-cases: (i) object detection and (ii) traffic sign recognition. The goal of object detection is to recognize instances of semantic objects from an image (including traffic signs), while traffic sign recognition consists in distinguishing different traffic signs (given a cutout of the sign).

We carry out our attack on four different models: Yolov3, Mask-RCNN, Lisa-CNN, and Gtsrb-CNN, demonstrating that depending on the ambient light conditions, the attack successfully makes a stop sign go undetected in over 90% of camera frames. We also evaluate the transferability of such attack showing that depending on the model used during the AE crafting phase, SLAP could be used to conduct black-box attacks. In particular we test a black-box attack against the Google Vision API showing that Mask-RCNN and Yolov3 provide adversarial patterns that lead to stop sign being not detected in over 95% of frames. Finally, we show how SLAP can bypass SentiNet [11], a recent AE detection method which leverages the high salient areas generated by adversarial patches to detect malicious examples. Since SLAP does not present a locality constraint as adversarial patches, we can bypass detection by SentiNet, leading to AE being incorrectly classified as benign samples in over 99% of cases.

**Contributions.**

- We propose SLAP, a novel attack vector for the realizability of AE in the physical world by using an RGB projector. This technique gives a set of advantages compared to printing patches and sticking them to the target object. These include that the attack is (i) opportunistically applicable and short-lived, which extends its inconspicuousness and targeting capabilities and (ii) dynamic, which leads to more flexibility to varying conditions at the time of attack.

- We propose a method to craft robust AE designed for use with a projector. The method models the effect of projections as perceived by a camera. We also enhance the robustness of such AE by systematically identifying and accounting for several varying environmental conditions during the optimization process.

- We evaluate the SLAP attack on two different computer vision tasks related to autonomous driving: (i) object detection and (ii) traffic sign recognition. We empirically show that our AE can achieve robust misclassification results under a set of different environmental conditions while also bypassing locality-based detection measures [11].

## 2 Background and Related Work

We start by introducing the necessary background on LCD projectors and object detection. We then cover the related work in physically-realizable adversarial examples.

### 2.1 Projector technology

A common LCD (liquid crystal display) projector works by sending light through a series of dichroic filters in order to form the red, green and blue components of the projected images. As the light passes through, individual pixels may be opened or closed to allow the light to pass [16], creating a wide range of colors. The total amount of light that projectors emit (measured in lumens), as well as the amount of light per area (measured in lux) is an important factor for determining the image quality, with stronger output leading to more accurate images in a range of conditions. Common office projectors are in the range of 2,000-3,000 lumens of emitted light, while the higher-end projectors can achieve up to tens of thousands of lumens (e.g., the projectors used during the London 2012 Olympics).\(^1\) As lumens only measure the total quantity of light emitted, a projector’s performance can be assessed by looking at its brightness in terms of lumens. However, it’s important to note that while lumens measure the total quantity of light, they do not account for the color quality or the contrast ratio of the projected image. The contrast ratio is determined by the difference between the black and white levels of the projected image, and is often expressed as a ratio (e.g., 1000:1).

\(^1\)https://www.digitaltrends.com/mobile/panasonic-tokyo-2020-technology-interview/
visible light emitted from the projector, the current ambient light perceived on the projection surface has an important role in determining the formed image contrast and color quality. The brighter the ambient light, the less visible will the image formed by a projector be due to weaker contrast and narrower range of colors.

As an example, a 2,000 ANSI lumens projector can emit enough light to obtain a light intensity of 2,000 lux on a square meter area (measured for white light [35]). Such a projector would reproduce an image in an office quite well (ambient ~500 lux), but could hardly make the image visible if it was placed outside in a sunny day (~18,000 lux). Additionally, projectors are generally used and tested while projecting on a (white) projection screen, which are designed to optimize the resulting image quality. When projecting on different materials and non-white surfaces, the resulting image will vary greatly given that light propagation significantly changes depending on the material in use and the background color. In particular, traffic signs are made of retroreflective materials [30] which allows most of the light to travel back to its source, i.e., the car headlights. In Section 4.1 we explain how we model such changes in an empirical way that accounts for many variability factors.

2.2 Object Detection

Object detection refers to the task of segmenting instances of semantic objects in an image. The output of object detectors is generally a set coordinates of bounding boxes in the input image that contain specific objects. In the following we detail two object detectors, Yolov3 [32] and Faster-RCNN [33] which are used throughout this paper.

Yolov3 is a single-shot detector which runs inputs through a single convolutional neural network (CNN). The CNN uses a back-bone network to compute feature maps for each cell in a square grid of the input image. Three grid sizes are used in Yolov3 to increase accuracy of detecting smaller objects (13x13, 26x26, 52x52). Yolov3 is used in many real-time processing systems [4, 6, 37].

Faster-RCNN is the result of a series of improvements on the initial R-CNN object detector network [17]. Faster-RCNN uses a two-stage detection method, where an initial network generates region proposals and a second network predicts labels for proposals. More recently, Mask-RCNN [19] extended Faster-RCNN in order to add object segmentation to object detection. Both Yolov3 and Mask-RCNN use non-maximum suppression in post-processing to remove redundant boxes with high overlap.

Traffic Sign Recognition. The task of traffic sign recognition consists in distinguishing between different traffic signs. Differently from object detection, in traffic sign recognition the networks typically require a cutout of the sign as input, rather than the full scene. Several datasets of videos from car dash cameras are available online, such as LISA [31] or

Figure 2: Example of an adversarial patch attack [18]. The network has been compromised and reacts to the sunflower being placed in the input by misclassifying the stop sign. SentiNet [11] leverages the locality of the patch to detect regions with high saliency, and can therefore detect the attack. The figure is taken from Figure 5 in [11].

GTSRB [20], in which a region of interest that identifies the ground-truth position of the traffic sign in each video frame is generally manually annotated. In this paper, for continuity, we consider two different models for traffic sign recognition, Lisa-CNN and Gtsrb-CNN, both introduced in [14], one of the earliest works in real-world robust AE.

2.3 Physical Adversarial Examples

Kurakin et al. [23] showed that perturbations computed with the fast gradient descent [36] method can survive printing and re-capture with a camera. However, these perturbations would not be realizable on a real (3D) input, therefore other works on physical attacks against neural networks have focused on adversarial patches [8, 22]. Evtimov et al. [13, 15] showed how to craft robust physical perturbations for stop signs, that survive changes when reproduced in the physical world (e.g., distance and viewing angle). The perturbation is in the form of a poster overlaid on the stop sign itself or a sticker patch that the authors apply to the sign. Sharif et al. [34] showed that physical AE for face recognition can be realized by using colored eye-glass frames, further strengthening the realizability of the perturbation in the presence of input noise (e.g., different user poses, limited color gamut of printers). Although most of these attacks are focused on evasion attacks, localized perturbations have also been used in poisoning attacks [18, 26] both by altering the training process or the network parameters post-training.

More recent works have focused on AE for object detection [10, 13, 38]. All these works use either printed posters or patches to apply on top of the traffic signs as an attack vector. As discussed in the previous section, patches suffer from several disadvantages that can be overcome with a projector, in particular projections are (i) short-lived and (ii) dynamic. This allows adversaries to turn the projection on/off as they please, which can be used to target specific vehicles and allows them
We focus on an autonomous driving scenario, where cameras will also cause misclassifications in other classes, which is an
will cause the stop sign to be misclassified as a warning sign.
Chou et al. [11] exploited the locality of ad-
AE Detection.

The same flower patch can be applied to different images and will cause the stop sign to be misclassified as a warning sign. This way SentiNet can adapt for unseen attacks and therefore claims to generalize to different attack methods. Further details can be found in the paper [11]. In this paper we show how AE generate with SLAP can bypass such detection.

3 Threat Model

We focus on an autonomous driving scenario, where cameras are placed in vehicles and the vehicle makes decisions based on the cameras’ inputs. The vehicle uses camera(s) to detect and track the objects in the scene, including traffic signs.

Goal. The adversaries’ goal is to cause a stop sign to be undetected by the neural networks processing the camera feeds within the car, which will cause vehicles approaching the stop sign to ignore them, potentially leading to accidents and dangerous situations. The adversary may want to target specific vehicles, therefore using patches to stick on the stop sign is not a suitable attack vector. In fact, patches would lead to the stop sign always being undetected by each passing vehicle and would cause suspicion among the victim drivers (e.g., each driver realizes that the car did not stop because of an altered sign).

Capabilities and Knowledge. The adversary has access to the general proximity of the target stop sign and can control a projector so that it points to the sign, see Figure 3. We note that the adversary does not necessarily need to have direct physical access to the sign itself – rather to a position from which a visual line of sight exists. In the paper we analyze both adversaries with white-box knowledge and a black-box scenario based on the transferability of adversarial examples.

4 Method

In this section we explain our method to carry out the attack.

4.1 Modeling projectable perturbations

When shining light with the projector, the resulting output color as captured by a camera depends on a multitude of factors, including: (i) projector strength, (ii) projector distance, (iii) ambient light, (iv) camera exposure, (v) color and material properties (diffusion, reflections) of the surface the projection is being shone on (hereafter, projection surface). The achievable color spectrum is significantly smaller than the spectrum available to printed stickers as a result of these factors (e.g., a patch can be black or white, while most projections on a stop sign will result in red-ish images). In order to understand the feasibility of certain input perturbations, we model these phenomena as follows.

Formalizing the problem. We wish to create a model which, given a certain projection and a projection surface, predicts the resulting colors in output (as captured by a camera). We can describe this model \( \mathcal{P} \) as follows:

\[
\mathcal{P}(\theta_1, S, P) = O, \tag{1}
\]

where \( S \) is the projection surface, \( P \) is the projected image, \( O \) is the image formed by of projecting \( P \) on \( S \) and \( \theta_1 \) are the model parameters, respectively.

Finding a perfect model would require taking all of the factors listed above into account, some of which may not be available to an adversary and is also likely to be time consuming due to the volume of possible combinations. Therefore, we opt for a sampling approach, in which we iteratively shine a set of colors on the target surface (the traffic sign) and collect the outputs captured by the camera. We then fit a model to the collected data, which approximates the resulting output color for given projected images and projection surfaces.

Collecting projectable colors. We define *projectable colors* for a given pixel in \( S \) as the set of color which are achievable in output for that pixel given all possible projection images. To collect the projectable colors, we do as follows:

1. collect an image of the projection surface (\( S \) in Eq. 1).
2. This is an image of the actual stop sign.
2. select a color \( c_p = [r, g, b] \), shine an image of that color \( P_{c_p} \) over the projection surface and collect the output \( O_{c_p} \).

3. repeat the previous step with different colors until enough data is collected.

In practice, with \( r, g \) and \( b \in [0, 255] \) we choose a certain quantization per-color channel and project all possible colors consecutively, while recording a video of the projection surface. This allows us to collect enough information about the full color space. With this method, we found that a quantization of 127 is enough to obtain sufficient accuracy for our method, so that we only need to project \( 3^3 = 27 \) colors to obtain enough data for our model.

Camera noise. In order to collect accurate data, our modelling technique has to account for noise that is being introduced by the camera. At first, we remove noise originating from the sensitivity of the light sensor (ISO [29]), shown in Figure 4. In fact, in non-bright lighting conditions, the camera increases the light-sensitivity of its sensor, which generates subtle pixel changes across consecutive (static) frames [5]. To overcome this factor, instead of collecting individual frames for \( S, P_{c_p}, O_{c_p} \), we collect 10 consecutive frames and compute and use the median of each pixel as our final image, the sensor does not update immediately when a certain color is produced by the camera. At first, we remove noise originating from the camera light sensor [20] between the two

Fitting a projection model. Once we have collected a set of \( S, P_{c_p}, O_{c_p} \) for the chosen set of colors, we construct a training dataset as follows. First we group together pixels of the same color by creating a mask for each unique color in the projection surface. In other words, we find the set of unique colors present in \( S \), i.e., \( c_s \in S_{uniq} \) and then create a mask for each color \( M(c_s) = \{i_1, \ldots, i_k\} \) such that:

\[
i \in M(c_s) \text{ i.f.f. } i^{th} \text{ pixel in } S == c_s.
\]

Then, for each unique source color \( c_s \), we extract all the mask-matching pixels from the output \( O_{c_p} \), average their colors to get an output color \( c_o^{(s,p)} \), and save the following triple for our training data \( \{c_s, c_p, c_o^{(s,p)}\} \). Such triple indicates that by projecting \( c_p \) on pixels of color \( c_s \), we obtained (on average) the color \( c_o^{(s,p)} \).

We then use the triples to fit a neural network composed of two hidden layers with ReLU activation, we re-write Equation 1 as an optimization problem as follows:

\[
\text{Loss}_{p} = \arg\min_{\theta_1} \sum_{c_s, c_p} \left\| P(c_s, c_p) - c_o^{(s,p)} \right\|_1,
\]

where \( P \) is the model. We optimize the network using gradient descent and Adam optimizer. Using \( P \) we have a differ-
entiable model which can used to propagate the derivatives through it during the AE generation 4.2.

Visualizing the Learned Model. Since the projection surface \( S \) is a stop sign, pixels in \( S \) generally can be separated into two clusters based on their color, corresponding to the “red” and “white” part of the sign. The presence of these two clusters is reflected in the outputs of the projection model, as different colors will be achievable in output for the red and white parts of the stop sign. We visualize the outputs of the projection model in Figure 5, where we use a learned projection model \( P \), the captured source image \( S \) and we compute a set of output colors for random projection colors \( c_p \). Each data point in Figure 5 corresponds to the color of an output pixel and is marked by a different marker (either triangle or circle) based on whether the corresponding source pixel was into the red or white cluster. Figure 5 shows that the model learns a different function for red or white source pixels, obtaining in output more blue tones for white pixels while different shades of red for the remaining red pixels.

4.2 AE Generation

In this section we describe our method for generating the adversarial projection. As a starting point, we combine the projection model described in Section 4.1 with the target network and use gradient descent along both to optimize the projected image. In its basic form, we optimize the following loss function:

\[
\arg\min_{\delta_x} J(f(t + P(x, \delta_x))) \quad \text{s.t. } 0 \leq \delta_x \leq 1
\]

where \( \delta_x \) is the projected image, \( f \) the detection network, \( P \) the projection model, \( x \) the input image background, \( x \) a stop sign image, and \( J \) the detection loss, described later. In the following we describe how we augmented the loss function in order to facilitate the physical feasibility of the adversarial perturbation and the convergence of the optimization.

Physical Constraints. In order to maintain the physical realizability of the projection, we do the following. At first, we restrict the granularity of the projection in a fixed grid of \( n \times n \) cells, so that each cell contains pixels of the same color. This allows us to use the same projection for different distances of viewing the stop sign. Secondly, we include the total variation of the projection in the loss function in order to reduce the effect of camera smoothing and/or blurring [28].

Variable Substitution. Since the optimization problem for the projection is bounded in [0,1] (space of RGB images) to ease the flowing of gradients when backpropagating we remove this box constraint. Given the image to project \( \delta_x \), we substitute \( \delta_x \) with a new variable \( w \) such that

\[
w = \frac{\tanh \delta_x}{2} + 0.5
\]

and instead optimize for \( w \). Since \( \tanh \delta_x \) is bounded in \([-1, 1] \) we find that this substitution leads to faster convergence in the optimization.

Loss Function. We also limit the amount of perturbation in our loss so that our final optimization looks as follows:

\[
\arg\min_w J(f(t + P(x, w))) + \lambda \|P(x, w) - x\|_p + TV(w),
\]

where \( \lambda \) is a parameter used to control the importance of the \( p \)-norm \( \| \cdot \|_p \) and TV is the total variation described above. Since we operate on both object detectors and traffic sign recognizers, we use two different losses \( J \) depending on the target network. For object detectors, we consider that the network returns a finite set of boxes \( b \in B \) where for each box there is an associated probability output of the box containing a semantic object of class \( j \), i.e., \( p^{(b)}_j \). For traffic sign recognizers, the network returns a probability vector containing the probability of the input image being traffic sign of class \( j \), i.e., \( p_j \). We then use the following loss functions in the two cases:

- **Object Detectors**: the loss is the sum of the detection probabilities for stop signs, i.e., \( \sum_{b \in B} p^{(b)}_j \);
- **Traffic Sign Classification**: the loss is the probability for the stop sign class \( p_j \).

4.3 Training Data Augmentation

Generating adversarial examples that work effectively in the physical world requires taking into account different environmental conditions. Adversarial examples computed with straightforward approaches such as in [36] do not survive different viewing angles or viewing distances [13]. In order to enhance the physical realizability of these samples, different input transformations need to be accounted for during the optimization. We use the Expectation over Transformation (EOT) method [14], which consists in reducing the loss over a set of training images computed synthetically. These training images are generated using linear transformations of the

Figure 7: Overview of the adversarial samples generation pipeline. We optimize the projected image which passes through the projection model in order to minimize the target detection score on a given DNN for a set of randomly generated permutations of the input.
desired input, i.e., an image containing stop signs, so that different environmental conditions can be accounted for during the optimization. Using EOT, our final loss becomes:

$$\text{Loss}_f = \arg \min_w E_{b \sim T, m_j \sim M} \mathcal{J}(f(t_i + m_j \cdot \mathcal{P}(x, w))) + \lambda \|\mathcal{P}(x, w) - x\|_p + \text{TV}(w),$$

(3)

where $T$ is a distribution over several background images and $M$ is an alignment function that applies linear transformations to the perturbed sign. In this work, we augment the set of the transformations to account for additional environmental conditions that are disregarded in previous work.

**Background and Traffic Sign Post.** Similarly to [38] we select a set of road backgrounds and carefully place the stop sign on a post at the edge of the road. In [38] it is shown that the post provides useful information to the detector and should therefore be included when crafting the adversarial perturbation.

**Perspective.** We vary the angle at which the camera is looking at the stop sign. Since we do not want to account for all perspective transforms, we use the following observations. Firstly, a traffic sign is mostly placed on one side of the lane (to the right in right-driving countries), meaning that rarely a camera mounted on a car would see a sign on the left-part of the frame. Secondly, traffic signs are mounted at specific heights (e.g., 5 or 7 feet in the US [3]), which normally exceed the height of cars for better visibility. Given these two observations, we prioritize perspective transforms that match these conditions.

**Distance.** As the car is approaching the stop sign, the sign will appear with different sizes in the camera frame. Our goal is for the car to misclassify the stop sign in every frame, therefore we place stop signs with different sizes during the optimization. We test the detection of the stop sign in non-adversarial settings with decreasing stop sign sizes and we set the minimum size of the sign to be the smallest size at which the sign is detected with high confidence. In other words, we only optimize for signs sizes that are large enough to be detected by the classifier.

**Rotation.** As shown in [12], simple rotations may lead to misclassifications when those transformations are not captured in the training dataset. We therefore add rotation to the stop sign when crafting the adversarial perturbation.

**Brightness.** The color of the stop sign changes based on a combination of ambient light and camera settings, e.g., in sunny days the colors appear brighter to the camera. To account for this, we apply different brightness transformations to the stop sign, so that we include a wider range of color tones. Since different colors contribute differently to an image brightness, we transform the stop sign image from RGB to YCrCb format [1], increase the luma component (Y) by a specified delta and then bring the image back into RGB.

| Parameter | Yolov3 | Mask-RCNN | Lisa-CNN | Gtsrb-CNN |
|-----------|--------|-----------|----------|-----------|
| learning rate | 0.005 | 0.005 | 0.05 | 0.05 |
| brightness | $[-13, +13]$ | $[0, 255]$ |
| perspective | $x$-axis $[-30^\circ, +30^\circ]$, $y$-axis $[-30^\circ, +30^\circ]$ |
| rotation | $[-5^\circ, +5^\circ]$ |
| aspect ratio | from 4:3 to 16:9 |
| sign size | [25, 90] pixels |
| grid size | $25 \times 25$ |

Table 1: Parameters used for the AE generation and the training data augmentation. The values for brightness, perspective, rotation, aspect ratio indicate the ranges for the applied transformations. All parameters are picked uniformly at random (with the exception of perspective) during the AE generation for each sample in the generated training data.

**Camera Aspect Ratio.** We observe that popular object detectors resize the input images to be squared before being processed by the network (e.g., Yolov3 resizes images to 416x416 pixels), to speed up the processing. However, the typical native aspect ratio of cameras, i.e., the size of the sensor, is 4:3 (e.g., the Aptina AR0132 chip used in the front-viewing cameras by Tesla, has a resolution of 1280x960 [2]). This leads to objects in the frames to be being distorted when the frames are resized to squared. To account for this distortion, we choose the dimension of the stop sign so that its height is greater than its width, reflecting a 4:3 to 1:1 resizing.

### 4.4 Remarks

We use AdamOptimizer to run the AE generation. We optimize a single variable that is the image to project with the projector (its substitute, see Section 4.2). We use batches of size 20. All the training images are created synthetically by placing a stop sign on a road background and applying the transformations described in the previous section. We do not use a fixed pre-computed dataset, a new batch with new images is created after every backpass on the network. The parameters for the transformations are chosen uniformly at random in the ranges shown in Table 1. For all operations that require resizing, we use cubic interpolation, finding that it provides more robust results compared to alternatives in this use-case. We run the optimization for 50 epochs, for each epoch we optimize the 20% worst-performing batches by backpropagating twice, convergence is usually reached before the last epoch. Compared to similar works [38], our method runs significantly faster requiring only 50 modifications of the perturbation (compared to 500), which takes less than 10 minutes on an NVIDIA Titan V GPU for Yolov3.
Table 2: Preliminary results for the two light settings considered in the experiment. The camera exposure is the exposure of the camera used for profiling (set automatically by the camera itself). The table shows the losses resulting from the optimization: $Loss_P$ refers to the loss in Equation 2, while $Loss_f$ refers to the loss in Equation 3, for each network.

| lux   | camera exposure (ms) | $Loss_P$ Yolov3 | Mask-RCNN | Gtsrb-CNN | Lisa-CNN |
|-------|----------------------|-----------------|-----------|-----------|----------|
| 180   | 120                  | 0.022           | 0.11      | 0.16      | 0.07     | 0.01     |
| 440   | 250                  | 0.016           | 1.44      | 4.24      | 0.27     | 0.12     |

5 Evaluation

In this section, we test the feasibility of the attack in real-world settings.

5.1 Experimental Setup

Projector Setup. To test our projection, we buy a real stop sign of size 600x600mm. For all of our experiments, we use a Sanyo PLC-XU4000 projector, which is a mid-range office projector (roughly $1,500) with 4,000 maximum lumens. We measure the projector light intensity with a Lux Meter Neoteck, following the 9-points measuring procedure used to measure ANSI lumens [35], which reports that in default settings the projector emits around 2,200 lumens. For the experiments, we place the projector 2 meters away from the stop sign, which, at maximum zoom, allows us to obtain roughly 800 lux of light on the stop sign surface. A similar amount of projected light can be obtained from greater distances by using long throw projectors, which are available for few thousand dollars (e.g., $3,200 for Panasonic PT AE8000\(^3\), see Section 6 for a more elaborate discussion). We align the projection to match the stop sign outline by transforming the perspective of the image.

Ambient Light. As mentioned in Section 4.1 the amount of ambient light limits the control on the input space for the adversary. In fact, as the ambient light increases, fewer colors are achievable as the projector-emitted light becomes less in the resulting appearance of the sign. To account for different ambient light levels, even though we conduct our experiments indoor we reproduce two separate lighting conditions: 180 lux and 440 lux (measured on the sign). For reference, on a clear day at sunrise/sunset the ambient light is roughly 400 lux, while on an overcast day at the same hours there are roughly 40 lux [7].

Networks and Detection Thresholds. We consider four different networks in our experiments: two object detectors, (1) Yolov3 and (2) Mask-RCNN, two traffic sign recognizers, (3) Lisa-CNN and (4) Gtsrb-CNN. For Yolov3, we use the Darknet-53 backbone of the original paper [32]. For Mask-RCNN, we use Resnet-101 as a backbone and feature pyramid network [24] for the region proposals. We download the weights for Lisa-CNN and Gtsrb-CNN from the GitHub of the paper authors [14]. As Mask-RCNN and Yolov3 return a list of boxes with a confidence score threshold for the output class, we set the threshold for detection at 0.6 and 0.4 respectively (i.e., we count detection as “there is a box labeled stop sign with score higher than x”). These are the thresholds that bring the highest mean Average Precision (mAP) in the coco object detection benchmark [25]. For Lisa-CNN and Gtsrb-CNN we set the detection threshold as 0.5. The input images are resized to 416x416 for Yolov3 and Mask-RCNN and to 32x32 for Lisa-CNN and Gtsrb-CNN.

Metrics and Measurements. Throughout our experiments we firstly monitor how often the stop sign goes undetected. For object detectors (Yolov3 and Mask-RCNN), we feed each frame into the network and we count how many times a stop sign is detected in input. For traffic sign recognizer (Gtsrb-CNN and Lisa-CNN), the network expects a cutout of a traffic sign rather than the full frame. In order to obtain the cutout, we manually label the bounding box surrounding the stop sign and use a CSRT tracker [27] to track the stop sign over the frames. We then count how often the predicted label is a stop sign. In order to monitor viewing angle and distance from the sign, we reconstruct the angle of view and distance based on the distortion on the octagonal outline of the sign and our recording camera field-of-view. We use the default camera app on an iPhone X to record a set of videos of the stop sign.

---

3https://www.projectorcentral.com/Sanyo-PLC-XU4000.htm
4https://github.com/evtimovi/robust_physical_perturbations/
at different distances and angles, with the projection being shone. As mentioned in Section 4.3, to match the 4:3 aspect ratio, we crop the 1080p video from the iPhone X (which has a resolution of 1920x1080) to 1440x1080 by removing the sides. The videos are recorded in a large lecture theater in our institution.

**Experimental Procedure.** Experiments follow this pipeline:

- **Step 1:** We setup the stop sign and measure the amount of lux on the stop sign surface;
- **Step 2:** We carry out the profiling procedure to construct a projection model (Section 4.1);
- **Step 3:** We use the projection model to run the AE generation (Section 4.2) and compute the optimal image to project;
- **Step 4:** We shine the image on the sign and we take a set of videos at different distances and angles.

Unless otherwise stated, the parameters used for the optimization (Step 3) are those of Table 1. Recording the profiling video of Step 2 requires less than 2 minutes, so does fitting the projection model.

**Preliminary Results.** Table 2 shows parameters and resulting value of the loss functions at the end of the optimizations for the two light settings. The table shows that our projection model fits the collected color triples: a loss $< 0.03$ shows that the error in the predicted colors, which are in $[0, 1]$ color space, is less than 1% per channel. The loss for the AE generation is also quite low for the 180 lux setting, where more colors are achievable, and a bit higher for the 440 lux setting. It should be noted that for object detectors this loss includes all predicted boxes scores before the non-maximum suppression step (used to remove redundant boxes), leading at times to scores higher than 1. We report in Figure 8 the projected images output of the optimization process, for each network and light setting. Figure 8 shows that the 440 lux brighter setting leads to the network using brighter colors in the projections, because these colors lead to more visible changes in the output.

### 5.2 Detection Results

In this section we present the results of the detection in white-box settings. Differently from related work [10, 13, 14, 38], we also report in Figure 9 the baseline results of using the networks to detect/classify the stop sign, by recording a video of the stop sign unaltered. This is fundamental to understand the effect of our projection compared to simple input noise causing misclassifications. Figure 9 shows that all networks work quite well in non-adversarial conditions, with the exception of Lisa-CNN which shows some misdetections for some combinations of angles and distances.

We report in Figure 10 the results of the detection in the 180 lux setting for the different networks, as the percentage of frames where the stop sign was not detected by the network. The reported signs are computed with the respective projection model (one for the 180 and one for the 440 setting), so they resemble how the stop sign looks in practice. The figures shows that the attack is extremely successful for a variety of angles and distances. For Yolov3 and Lisa-CNN, we obtain more than 97% success on average between 3-12m of distance and $-30\degree$ to $30\degree$ viewing angles. For Mask-RCNN and Gtsrb-CNN there is a fading effect for angles greater than $20\degree$ degrees and higher distances, while high success is obtained for closer ranges 3-6m and smaller angles. We found that at greater distances the changes induced by the projection are not as clear, as less of the projected light bounces from the sign onto the recording camera, making the perturbation less evident.

Figure 10 also shows the same figures for the 440 lux setting. Given the higher ambient light, the amount of change achievable on the sign is greatly reduced, leading to better detection rates for the sign. The figure shows the same trend for increasing distances, as now the changes induced by the projector are not perceived well from far away. We found that Mask-RCNN is more resilient than the other networks in the detection, leading to success rates below 10% on average. In
Figure 10: Attack success rate at different angles, distances and for different networks, as the percentage of frames where a stop sign is not detected. Darker shades represent higher success rates. Percentages for 0-3m are omitted for clarity, but the corresponding cone section is colored accordingly. The first row of figures (Figure 10a, 10b, 10c and 10d) refer to the 180 lux setting. The second row of figures (Figure 10e, 10f, 10g and 10h) refer to the 440 lux setting. The images of the stop signs in the figure are computed using the projection models for the two light settings, so they resemble what the adversarial stop sign looks like in practice. (*) we set slightly different parameters for Mask-RCNN as it lead to more robust AE: the stop sign size is [30, 104] while the grid size is 35 ¥ 35, see Table 1 for reference.

particular we found that Mask-RCNN sometimes recognizes stop signs just based on the octagonal silhouette of the sign or even just with faded reflections of the sign on windows. This is probably a result of the training time augmentations used by the training process of Mask-RCNN (different scales, horizontal flips, see [19]).

Straight Approach. We also record videos of straight approaches towards the stop sign, Figure 11 reports the result for the 180 lux setting. The attack works well for most networks with detection rates below 5% for all networks except for Mask-RCNN, which shows greater resilience at higher distances (as the projection fades away) and at shorter ones.

5.3 Attack Transferability

In this section we test the transferability of our attack across networks, testing all pairwise combinations of our models. We also use the Google Vision API5 to test our projections against their proprietary models. The API returns a list of labeled objects in the image with associated confidence scores and bounding boxes, "stop sign" is one of the labels. We set the detection threshold for Google Vision API as 0.01, i.e., we count that a stop sign is detected in a frame if the API replies with a stop sign object with confidence greater than 0.01.

Figure 11: Ratio of detection of stop sign during the straight approach towards it, for the 180 ambient light settings. The data are grouped into 1m bins, the ratio of detection shows the ratio of frames where a stop sign is detected by the network, vertical bars show the standard deviation for the bin. The data points are slightly shifted on the x-axis to avoid overlaps.

5https://cloud.google.com/vision
We report the results in Table 3. The table shows the source (white-box) model on the left, which identifies the projection shown in the tested videos (we also report the number of frames tested). On the right, there are the success rates of the attack as a percentage of the frames where the stop sign was undetected. Table 3 shows that our attack does not transfer well across many pairs. This is due to the limited control over the input space that a projector allows to adversaries. In comparison, patches allow for changes in input close to X since the patches are overlayed on top of the stop sign. The fact that in brighter settings (440 lux) the results worsen is also a result of this limited control. We find that Mask-RCNN transfers better to Yolov3 compared to the opposite direction. AE computed on Mask-RCNN transfer better to black-box models. We find that the our adversarial projections are particularly effective against the Google Vision API, this is particularly true for Yolov3 and Mask-RCNN.

| lux | Source Model | no. frames | Yolov3 | Mask-RCNN | Gtsrb-CNN | Lisa-CNN | Google Vision* |
|-----|--------------|------------|--------|------------|-----------|---------|---------------|
| 180 | Yolov3       | 7390       | -      | 4.0%       | 14.6%     | 17.5%   | 95.8%         |
|     | Mask-RCNN    | 6668       | 22.4%  | -          | 25.0%     | 30.5%   | 95.5%         |
|     | Gtsrb-CNN    | 7058       | 12.3%  | 2.0%       | -         | 13.4%   | 76.4%         |
|     | Lisa-CNN     | 5839       | 9.0%   | 0.6%       | 35.7%     | -       | 58.4%         |
| 440 | Yolov3       | 6340       | -      | 0.8%       | 40.3%     | 40.9%   | 71.2%         |
|     | Mask-RCNN    | 5589       | 5.4%   | -          | 41.1%     | 35.6%   | 67.4%         |
|     | Gtsrb-CNN    | 6149       | 0.7%   | 0.7%       | -         | 35.6%   | 72.5%         |
|     | Lisa-CNN     | 5830       | 1.0%   | 2.4%       | 26.4%     | -       | 66.9%         |

Table 3: Transferability results. The table reports the percentage of frames where the stop sign was undetected by the target network. We test all the frames from the collected videos with a certain projection being shone against a different target model. Results shown for both light settings. (*) we only test one frame every 30 frames, i.e., one per second.

5.4 Bypassing attack detection

We evaluate the AE generated by SLAP against SentiNet, which is a system designed to detect adversarial examples leveraging the intuition of locality of patches. In short, if an adversarial sample contains a patch which causes a misclassification, then the saliency of the area containing the patch will be high. Therefore, the salient area will cause misclassifications on other legitimate samples when overlayed on them (the full details of the SentiNet implementation can be found in the paper [11]).

We attempt to detect adversarial examples with SentiNet in Gtsrb-CNN and Lisa-CNN. We select 500 random frames from the videos targeting each of these networks (we use videos captured in the 180 lux setting) to be tested for AE. We use 50 test images taken from the GTSRB and the LISA datasets within the SentiNet algorithm. The suspected adversarial regions are overlaid on these test images to detect whether they would cause misclassifications. We use 500 benign images for each network to calculate the threshold function, which defines whether an image will be classified as an AE or not.

To compute the salient areas in input SentiNet uses GradCam++ [9], which backpropagates the outputs to the last convolutional layer of the network and checks which region of the input lead to greater activations. Since the resolution of this layer is only 4x4 for both Gtsrb-CNN and Lisa-CNN, we instead use XRAI [21], a newer and more accurate method to compute salient areas. Using GradCam for this task would make the output masks unusable as a resolution of 4x4 leads to coarse block like regions where salient areas cannot be accurately identified (resolution also is pointed out as a problem in the original paper [11]). XRAI on the other hand produces saliency regions at the input resolution, leading to more granular salient areas, using an algorithm that incrementally grows salient regions. As a consequence of this improved technique XRAI has been shown to outperform older saliency algorithms, producing higher quality, tightly bound saliency regions [21].

SentiNet relies on being able to compute a threshold function which separates the behavior of AE from benign images. In the original SentiNet paper this threshold function is computed using the Average Confidence, i.e., the average confidence of the network prediction made on benign test images where salient masks are replaced with inert patterns added to them and the Fooled Percentage, i.e., the percentage of benign test images where overlaying the salient mask leads the network to predict the suspected adversarial class. These two scores characterize benign behaviour and can almost perfectly separate benign from adversarial inputs in SentiNet. We follow the same technique as in the original paper for fitting the threshold function that separates the malicious and bening data.

Results. Figure 12 shows the results of the detection. We find that for both Lisa-CNN and Gtsrb-CNN the thresholded curve includes all bening and malicious examples together. We find that the true positive rate is <2% in all cases, due
6 Discussion

In this section we discuss the attack feasibility based on our experimental results and point out the advantages of SLAP compared to adversarial patches.

Attack Feasibility. Our experiments demonstrate that the ambient light can easily limit the feasibility of the attack in bright conditions. In practice, the attack could be carried during non-bright days, e.g., overcast days, after sunset or before sunrise, as the level of ambient light during these times is low (<400 lux). The amount of light that reaches the sign depends on three factors: (i) the distance between projector and sign, (ii) the throw ratio (or projector lens field-of-view) of the projector and (iii) the amount of lumens the projector can emit. We report in Figure 13 a representation of how the distance between the projector and the stop sign relates to the attack success rate. We consider two projectors with long throw distance as example, the Panasonic PT-RZ570BU and the NEC PH1202HL1, which are available for $3,200 and $44,379 respectively. We use the projector’s throw ratios (2.93 and 3.02) and their emitted lumens (5,000 and 12,000 lumens) to calculate how many lux of light the projector can shine on the sign surface for increasing distances. We consider the 180 lux ambient light setting, as shown in our experiments, obtaining 800 lux of light on the sign with the projector is sufficient to achieve consistent attack success in this case. Figure 13 shows that the attack could be carried out from 7.5m away with the weaker projector and up to 13m away with the more expensive one. Adversaries could also use different lenses to increase the throw ratio of cheaper projectors, or stack multiple projectors in order to increase the amount of lux achieved on the sign.

Attack Flexibility. Compared to adversarial patches, the SLAP attack offers several advantages. Firstly, the projection can be turned on and off, which leads to the ability of carrying out opportunistic attacks which target specific vehicles. Secondly, using a projector makes for a much stealthier attack compared to patches/stickers as: (i) the stop sign does not change its appearance until the attack starts being carried

Figure 12: SentiNet detection results for Lisa-CNN and Gtsrb-CNN. The figure shows that the benign behavior (in terms of Average Confidence and Fooled Percentage) modeled by the fitted SentiNet line does not separate bening examples from AE, and thus SentiNet fails to detect our attack. Results are visualized for the random noise intert pattern.

Figure 13: Amount of lux achievable on the stop sign surface for increasing projection distances and two different projectors. The horizontal line shows the threshold for success measured in our experiments (800 lux).
Table 4: Results of AE detection using SentiNet. We test both random noise and checkerboard inert patterns as suggested in the original paper. The prediction is positive when the examples is an AE, while negative when it is a bening example. Results show that the larger perturbation obtained with the SLAP attack bypasses the detection.

| Network  | Inert Noise     | TPR | TNR | FPR | FNR |
|----------|-----------------|-----|-----|-----|-----|
| Gtsrb-CNN| Checkerboard    | 1%  | 0%  | 100%| 98% |
|          | Random          | 2%  | 0%  | 100%| 97% |
| Lisa-CNN | Checkerboard    | 1%  | 0%  | 100%| 98% |
|          | Random          | 1%  | 0%  | 100%| 98% |

out, (ii) the attack generates larger areas of perturbation in the input and can therefore bypass detection countermeasures such as SentiNet. Additionally, using SLAP is a more flexible way to obtain robust AE which allows adversaries to improve their adversarial perturbation much more dynamically as the conditions change, as there is no need to print a different patch and apply it on the sign.

7 Conclusions

In this paper we presented SLAP, a new attack vector to realize dynamic and inconspicuous physical adversarial examples by leveraging the use of a light projector. We investigate a scenario in autonomous driving, where the attacker’s goal is to change the appearance of a stop sign by shining a specifically crafted projection onto it so that the sign is undetected by object detectors or traffic sign recognition networks.

Given the non-trivial physical constraints of projecting specific light patterns on various materials in a range of conditions, we proposed a method to generate projections that is based on fitting a predictive color model and using an AE generation pipeline that enhances the robustness of the resulting AEs. We evaluated the proposed attack against state-of-the-art object detectors Yolov3 and Mask-RCNN and against traffic sign recognition networks Lisa-CNN and Gtsrb-CNN. The experimental results show that SLAP can generate AEs that are robust in the real-world, for varying angles ranging from $-30^\circ$ to $30^\circ$ and distances in 3-12m. We additionally showed that our AEs can perfectly bypass previously proposed detection measures that rely on the locality of adversarial patches.

The novel capability of modifying how an individual object is detected by common DNN models, combined with the capability of carrying out opportunistic attacks, makes SLAP an important and powerful new attack vector that requires further investigation. By making the full analysis and the source code available to the wider research community, we believe this paper makes an important first step towards increasing the awareness and research of countermeasures against the light-projection based adversarial attacks against DNNs.

Acknowledgments

(blinked for review)

Source Code Availability

We make all of the code used in our experiments available at the following GitHub repository: (blinded for review). Additionally, we include detailed instructions for repeating the experiments, as well as Docker containers to ease the setup of such experiments within the repository.

References

[1] Jpeg file interchange format. https://www.w3.org/Graphics/JPEG/jfif3.pdf, 1992. Accessed: 2020-01-15.

[2] 1/3-inch cmos digital image sensor ar0132at data sheet. https://datasheetspdf.com/pdf/829321/AptinaImagingCorporation/AR0132AT, 2012. Accessed: 2020-01-08.

[3] Manual of uniform traffic control devices for street and highways. https://mutcd.fhwa.dot.gov/pdfs/2009r1r2/mutcd2009r1r2edition.pdf, 2012. Accessed: 2020-01-08.

[4] Apolloon - an open autonomous driving platform. https://github.com/apolloon, 2017. Accessed: 2020-01-14.

[5] The exposure triangle: aperture, shutter speed and iso explained. https://www.techradar.com/uk/how-to/the-exposure-triangle, 2018. Accessed: 2019-11-26.

[6] BMW innovation lab. https://github.com/BMW-InnovationLab, 2019. Accessed: 2020-01-14.

[7] Daylight. https://en.wikipedia.org/wiki/Daylight, 2019. Accessed: 2020-02-20.

[8] Tom Brown, Dandelion Mane, Aurko Roy, Martin Abadi, and Justin Gilmer. Adversarial Patch. Number ARXiv, 2018.

[9] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 839–847. IEEE, 2018.

[10] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 52–68. Springer, 2018.

[11] Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. SentiNet: Detecting Physical Attacks Against Deep Learning Systems. 2018.

[12] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the Landscape of Spatial Robustness. Proceedings of the 36th International Conference on Machine Learning, 97:1802–1811, 2019.
Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick.
Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and
Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan
Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He,
Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversar-
Danny Karmon, Daniel Zoran, and Yoav Goldberg. LaV AN :
Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc
Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Bad-
Mark Fischetti. Two technologies shine. Scientific American,
Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Ma-
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Bad-
Mark Fischetti. Two technologies shine. Scientific American,
Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Ma-
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir
Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir