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

Giulio Lovisotto
University of Oxford, UK

Henry Turner
University of Oxford, UK

Ivo Sluganovic
University of Oxford, UK

Martin Strohmeier
armasuisse

Ivan Martinovic
University of Oxford, UK

Abstract

Research into adversarial examples (AE) has developed rapidly, yet static adversarial patches are still the main technique for conducting attacks in the real world, despite being obvious, semi-permanent and unmodifiable once deployed.

In this paper, we propose Short-Lived Adversarial Perturbations (SLAP), a novel technique that allows adversaries to realize physically robust real-world AE by using a projector. Attackers can project specifically crafted adversarial perturbations onto real-world objects, transforming them into AE. This grants adversaries greater control over the attack compared to adversarial patches, as projections can be turned on and off as needed and leave no obvious trace of an attack.

We study the feasibility of SLAP in the self-driving scenario, targeting both object detector and traffic sign recognition tasks, focusing on the detection of stop signs. We conduct experiments in a variety of ambient light conditions, including outdoors, showing how in non-bright settings the proposed method generates AE that are extremely robust, causing misclassifications on state-of-the-art neural networks with up to 99% success rate. Our experiments show that SLAP-generated AE do not present detectable behaviours seen in adversarial patches and therefore bypass SentiNet, a physical AE detection method. We evaluate other defences including an adaptive defender using adversarial learning which is able to thwart the attack effectiveness up to 80% even in favourable attacker conditions.

1 Introduction

Recent advances in computational capabilities and machine learning algorithms have led 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 to the network, DNNs are susceptible to carefully crafted adversarial examples (AE) [40], which leverage specific directions in input space to create examples which whilst resembling legitimate images, will be misclassified at test time.

(a) Non adversarial scenario. (b) Adversarial projection.

Figure 1: The attack visualized. A projector shines a specific pattern on the stop sign causing an object detector (Yolov3 in this picture) to misdetect the object.

A significant body of earlier research focused on analyzing AE in the digital domain, where an adversary has the capability of making pixel-specific manipulations to the input. This concept has been further developed with the realization of physically robust AE [12, 15, 29, 37, 38, 43], which are examples that survive real-world environmental conditions, such as varied viewing distances or angles. In order to realize AE, adversaries can either print patches (e.g. as stickers or glasses in the case of face recognition), or replace an entire object by overlaying the object with a printed version of it with subtle changes. However, these techniques have multiple limitations. Firstly, these methods typically generate highly salient areas in the network inputs, which makes them detectable by recent countermeasures [13]. 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.

In this paper, we focus on road safety with autonomous vehicles 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. SLAP-generated AE provide the attacker with multiple benefits over existing patch-based methods, in particular giving fine-grained control over the timing of attacks, allowing them to become *short-lived*.

As part of designing the SLAP attack, we propose a method to 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, which we propagate the derivatives of the projection through during the AE crafting phase. Our method improves the established *non printability score* [37] (NPS) used in patch-based AE by modelling a three-way additive relationship between the projection surface, the projection color, and the camera-perceived output. Furthermore, we improve the robustness of AE in the physical world by systematically identifying and accounting for a large set of environmental changes. We empirically analyze the feasibility of SLAP on two different use-cases: (i) object detection and (ii) traffic sign recognition.

To understand the relationship between ambient light and attack feasibility, we collect extensive measurements in different light conditions, including outdoors. We conduct our attack on four different models: Yolov3, Mask-RCNN, Lisa-CNN, and Gtsrb-CNN, demonstrating the attack can successfully render a stop sign undetected in over 99% of camera frames, depending on ambient light levels.

We also evaluate the transferability of our 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 show that AE generated with Mask-RCNN and Yolov3 transfer onto the proprietary Google Vision API models in up to 100% of cases.

Finally, we evaluate potential defences. We show that SLAP can bypass SentiNet [13], a recent defence tailored to physical AE detection. Since SLAP does not present a locality constraint in the same way as adversarial patches, SLAP AE bypass SentiNet over 95% of the time. We investigate other countermeasures and find that an adaptive defender using adversarial learning can prevent most attacks, but at the cost of reduced accuracy in non-adversarial conditions.

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 the attacker new capabilities compared to existing approaches, including short-livedness and undetectability.
- We propose a method to craft robust AE designed for use with a projector. The method models a three-way additive relationship between a surface, a projection and the camera-perceived image. We enhance the robustness of the attacks by systematically identifying and accounting for varying environmental conditions during the optimization process.
- We evaluate the SLAP attack on two different computer vision tasks related to road safety for autonomous driving: (i) object detection and (ii) traffic sign recognition. We conduct an extensive empirical evaluation, including in- and out-doors, showing that under favourable lighting conditions the attack leads to the target object being undetected.
- We evaluate countermeasures. We firstly show that SLAP AE bypass locality-based detection measures such as SentiNet [13], which is tailored for the detection of physical AE. We then show that an adaptive defender using adversarial learning can thwart most of the attacks.

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, 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 [9]). As lumens only measure the total quantity of 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 [39]). 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 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 [35] and Faster-RCNN [36] 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 [6, 8, 41].

Faster-RCNN is the result of a series of improvements on the initial R-CNN object detector network [16]. 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 [20] 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 [34] or GTSRB [21], 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 [15], one of the earliest works in real-world robust AE.

2.3 Physical Adversarial Examples

Kurakin et al. [25] showed that perturbations computed with the fast gradient descent [40] 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 [10, 23]. Eykholt et al. [15, 38] 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. [37] 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, 28] both by altering the training process or the network parameters post-training.

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 [13] 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 [13].

More recent works have focused on AE for object detection [12, 24, 38, 43]. 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 short-lived and dynamic. This allows adversaries to turn the projection on/off as they please, which can be used to target specific vehicles and allows them to leave no traces of the malicious attack.

Physical AE Detection. Differently from a digital scenario, where input changes are simply limited by L_p-norms, the realization of physically robust AE is more constrained. Adversarial patches are one technique for phsyical AE, however, they have drawbacks which enable their detection. In fact, Chou et al. [13] exploited the locality of adversarial patches to create an AE-detection method named SentiNet, which detects physical AE leveraging the fact that adversarial patches generate localized areas of high saliency in input, as shown in Figure 2. These highly salient areas successfully capture the adversarial patch in input, and therefore can be used for the detection of an AE by using the fact that such salient areas will cause misclassifications when overlaid onto other benign images. For example, Figure 2 shows that an adversarial patch shaped as a flower will cause the stop sign to be misclassified as a warning sign. The same flower patch can be applied to different images and will also cause misclassifications in other classes, which is an unusual behavior which can be detected. SentiNet can capture this behavior just by looking at the saliency masks of benign images, and fitting a curve to the accepted behavior range, rather than fitting a binary model for the detection. This way SentiNet can adapt for unseen attacks

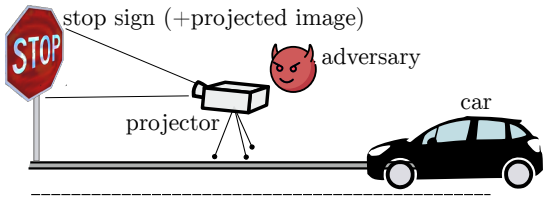


Figure 3: Attack scenario. An adversary points a projector at a stop sign and controls the projection in order to cause the sign to be undetected by an approaching vehicle.

and therefore claims to generalize to different attack methods. In this paper, we show how AE generated 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 adversarial patches to stick on the stop sign is not a suitable attack vector. Patches would lead to the stop sign always being undetected by each passing vehicle and would cause suspicion among the occupants realizing that cars 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 Modelling projectable perturbations

Often, to realize physical AE, researchers use the non-printability-score introduced by Sharif. [37], which models the set of colors a printer is able to print. In our case, when shining light with the projector, the resulting output color as captured by a camera depends on a multitude of factors rather than just the printer (as in NPS). These factors include: (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 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 projecting P on S and θ_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 target object) 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). This is an image of the target object.
2. select a color $c_p = [r, g, b]$, shine an image of that color P_{c_p} over the projection surface, 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

Figure 4: Camera light sensor noise visualized. The first two images show consecutive frames, while the third image shows the absolute pixel-wise difference ($\times 20$) between the two frames. Such sensor noise is accounted for with smoothing over many frames during the data collection step.

from the sensitivity of the light sensor (ISO [33]), shown in Figure 4. In fact, in non-bright lighting conditions, the camera increases the light-sensitivity of image sensor, which generates subtle pixel changes across consecutive (static) frames [19]. 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 camera is static during this process.

Secondly, we found that there is a smoothing over-time effect in the sensor readings while recording the video, so that the sensor does not update immediately when a certain color is being shown. Figure 6 shows how the average pixel color per channel changes over time in relation to the timing of certain projections being shown. The camera does not immediately stabilize to the resulting color when a projection is shown, but adjusts over a few frames. To account for this adaptation, during the data collection, we interleave each projected color with 10 frames of no projection, so that the camera re-adapts to the unaltered image of the projection surface.

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., $c_s \in S_{unq}$ and then create a mask for each color $M^{(c_s)} = \{i_j, ..., i_k\}$ 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 $\{c_s, c_p, c_o^{(s,p)}\}$. A 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:

$$Loss_{\mathcal{P}} = \arg\min_{\theta_1} \sum_{\forall c_s, c_p} \left\| \mathcal{P}(c_s, c_p) - c_o^{(s,p)} \right\|_1, \qquad (2)$$

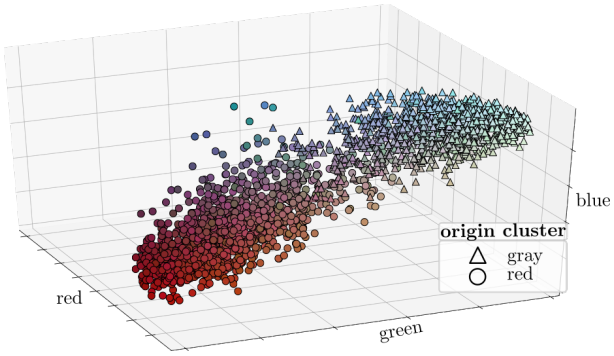


Figure 5: Plot showing the output space of the learned projection model. Each data point correspond to a color in S and its color is the model output $\mathcal{P}(c_s, c_p)$ for a random c_p.

where $\mathcal{P}$ is the model. We optimize the network using gradient descent and Adam optimizer. Using $\mathcal{P}$ we have a differentiable model which can be used to propagate the derivatives through it during the AE generation, see Section 4.2.

Visualizing the Learned Model. When the projection surface S is a stop sign (as mainly investigated in this paper), 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 $\mathcal{P}$, 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:

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

where δ_x is the projected image, f the detection network, $\mathcal{P}$ the projection model, x the input image background, x a stop sign image, and J the detection loss, described later. In the

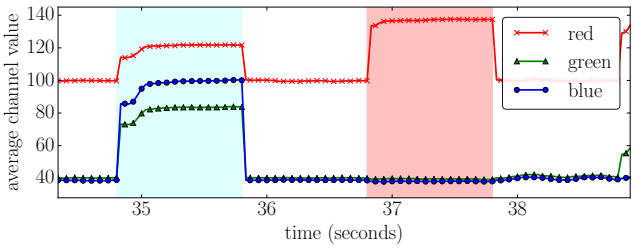


Figure 6: Plot showing how the average value of a pixel (RGB) changes when a certain projection is shown. Immediately after the projection is shown, the camera requires a few frames (the lines are marked every 2 frames) to converge to a stable value. The two shaded areas mark the time the projection is being shown and are colored with the projection color.

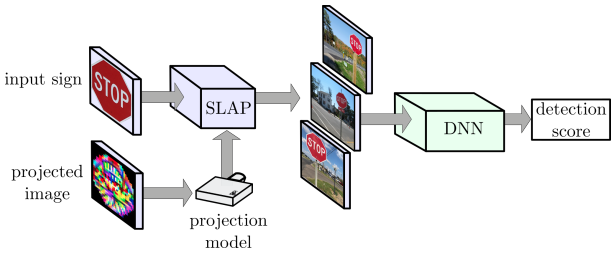


Figure 7: Overview of the adversarial samples generation pipeline. We optimize the projected image which passes through the projection model in order to minimize the target detection score on a given DNN for a set of randomly generated permutations of the input.

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. We improve the physical realizability of the projection with two steps. In order to maintain the physical realizability of the projection we have two two steps. 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 [31].

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 δ_x, we substitute δ_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:

$$\arg\min_w J(f(t+\mathcal{P}(x,w))) + \lambda \|\mathcal{P}(x,w)-x\|_p + \mathrm{TV}(w),$$

where λ 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., $p_j^{(b)}$. 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., $\sum_{b \in B} p_j^{(b)}$;
- **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 [40] do not survive different viewing angles or viewing distances [38]. 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 [15], which consists in reducing the loss over a set of training images computed synthetically. These training images are generated using linear transformations of the 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:

$$\begin{aligned} Loss_f = \arg\min_w \mathbb{E}_{t_i \sim T, m_j \sim M}\ & J(f(t_i + m_j \cdot \mathcal{P}(x,w))) \\ & + \lambda \|\mathcal{P}(x,w)-x\|_p + \mathrm{TV}(w), \end{aligned} \tag{3}$$

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. We report in Figure 7 an overview of the complete optimization used in our method.

Background and Traffic Sign Post. Similarly to [43] we select a set of road backgrounds and carefully place the stop

sign on a post at the edge of the road. In [43] 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 [14], 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 [2], increase the luma component (Y) by a specified delta and then bring the image back into RGB.

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 [7]). This leads to objects in the frames to 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.

Parameter	Yolov3	Mask-RCNN	Lisa-CNN	Gtsrb-CNN
learning rate	0.005	0.005	0.05	0.05
brightness	$[-13, +13]$ (with range [0, 255])			
perspective	x-axis $[-30°, +30°]$, y-axis $[-30°, +30°]$			
rotation	$[-5°, +5°]$			
aspect ratio	from 4:3 to 16:9			
sign size	[25, 90] pixels			
grid size	25×25			

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.

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. We run the optimization for 50 epochs, in one epoch we feed 600 generated images containing a stop sign in the network. 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 [43], 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.

5 Evaluation

In this section, we test the feasibility of the attack in practice.

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 [5], which is a mid-range office projector (roughly $1,500) with 4,000 maximum lumens. We carry out the experiment in a large lecture theatre in our

lux	camera exposure (ms)	$Loss_{\mathcal{P}}$	$Loss_f$ Yolov3	Mask-RCNN	Gtsrb-CNN	Lisa-CNN
120	33	0.020	0.09	0.08	0.01	0.06
180	25	0.023	0.11	0.52	0.00	0.07
300	18	0.017	0.68	0.86	0.89	1.03
440	12	0.015	1.44	4.24	5.31	2.45
600	9	0.011	1.80	5.92	9.12	8.16

Table 2: Preliminary results for the various light settings considered in the experiment. The camera exposure is the exposure of the camera used for profiling (set automatically). The table shows the optimization losses: $Loss_{\mathcal{P}}$ refers to the loss in Equation 2, while $Loss_f$ refers to the loss in Equation 3.

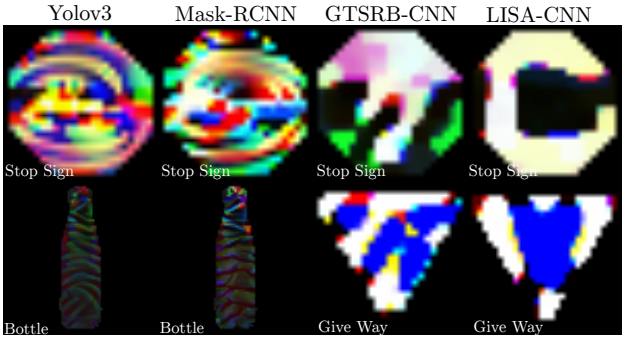


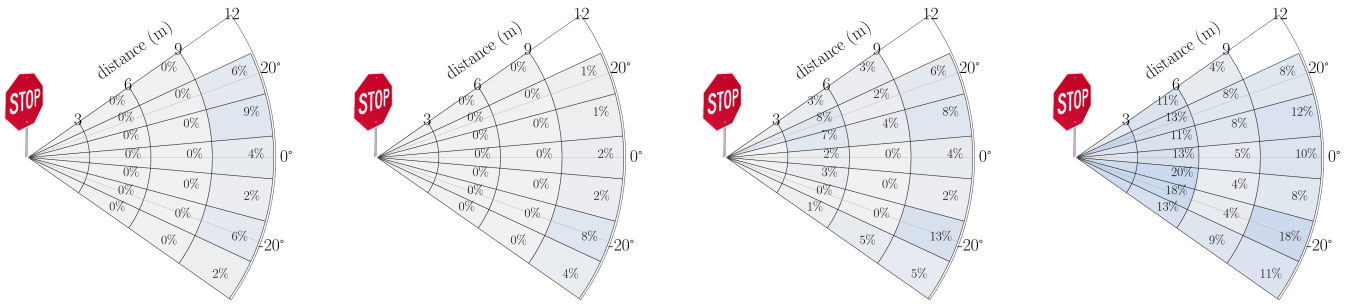
Figure 8: Examples of the projected images computed with the optimization. Bottom-right of each image specifies the target class fed to Equation 3. These images are computed within the 180 lux setting.

institution. We measure the projector light intensity with a Lux Meter Neoteck, following the 9-points measuring procedure used to measure ANSI lumens [39], 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 (white) light on the stop sign surface. We use this 800 lux white value to make considerations on the attack feasibility in Section 6. A similar amount of projected light can be obtained from greater distances by using long throw projectors, available for few thousand dollars (e.g., $3,200 for Panasonic PT AE8000 [4], see Section 6). 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, we conduct our experiments indoor and we control the amount of light hitting the stop sign (Section 5.2). We further evaluate the attack outdoors with a road driving test (Section 5.3). To reproduce various light settings indoors, we use both the ceiling lights mounted in the indoor hall and by using an additional 60 Watts LED floodlight pointed at the sign. We measure the attack in five different light settings: 120, 180, 300, 440 and 600 lux. The darker setting (120 lux) corresponds to slightly dimming the ceiling lights only. The 180 lux setting corresponds to the normal indoor lighting found in the lecture theatre where we carry out the measurements. Higher settings are achieved by adding the LED floodlight pointed directly at the sign at different distances (from roughly 4m away at 300 lux to less than 2m away at 600 lux). 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 [1].

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 [35]. For Mask-RCNN, we use Resnet-101 as a backbone and feature pyramid network [26] for the region proposals. We download the weights for Lisa-CNN and Gtsrb-CNN from the GitHub of the paper authors [15]. 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 [27]. 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. 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 the 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 [30] 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 at different distances and angles, with the projection being shone. The iPhone is mounted on a stabilizing gimbal to avoid excessive blurring.



(a) Yolov3. (b) Mask-RCNN. (c) Gtsrb-CNN. (d) Lisa-CNN.

Figure 9: Baseline mis-detection rate in absence of the adversary for the 180 lux setting at different angles, distances and for different networks, as the percentage of frames where a stop sign is **not** detected. Brighter shades represent higher detection rates. Percentages for 0-3m are omitted for clarity, but the corresponding cone section is colored accordingly.

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.

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); this uses a separate Logitech C920 HD Pro Webcam rather than the iPhone X camera (on which the attack is later evaluated).
- **Step 3:** We use the projection model to run the AE generation (Section 4.2) and optimize the image to project;
- **Step 4:** We shine the image on the sign and we take a set of videos at different distances and angles.

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 various light settings. The table shows that our projection model fits the collected color triples: $Loss_{\mathcal{P}} < 0.03$ shows that the error in the predicted colors, is less than 1% per channel. As expected, we found that the results of the optimization match the reduced capability to reproduce colors for higher ambient light settings: $Loss_f$ goes from roughly 0 to higher values as the light increases. Note that the reported $Loss_f$ is summed over the batches of 20 and computed before non-maximum suppression. We report in Figure 8 examples of projected images output of the optimization process, for three different target objects: stop sign, give way sign and bottle. Whilst we limit the rest of the experiments to attacks on stop signs, we report considerations and results on generalizing the attack to various objects in Appendix A and in Section 6.

Artifacts Availability. The experiments code and data are available online.[1]

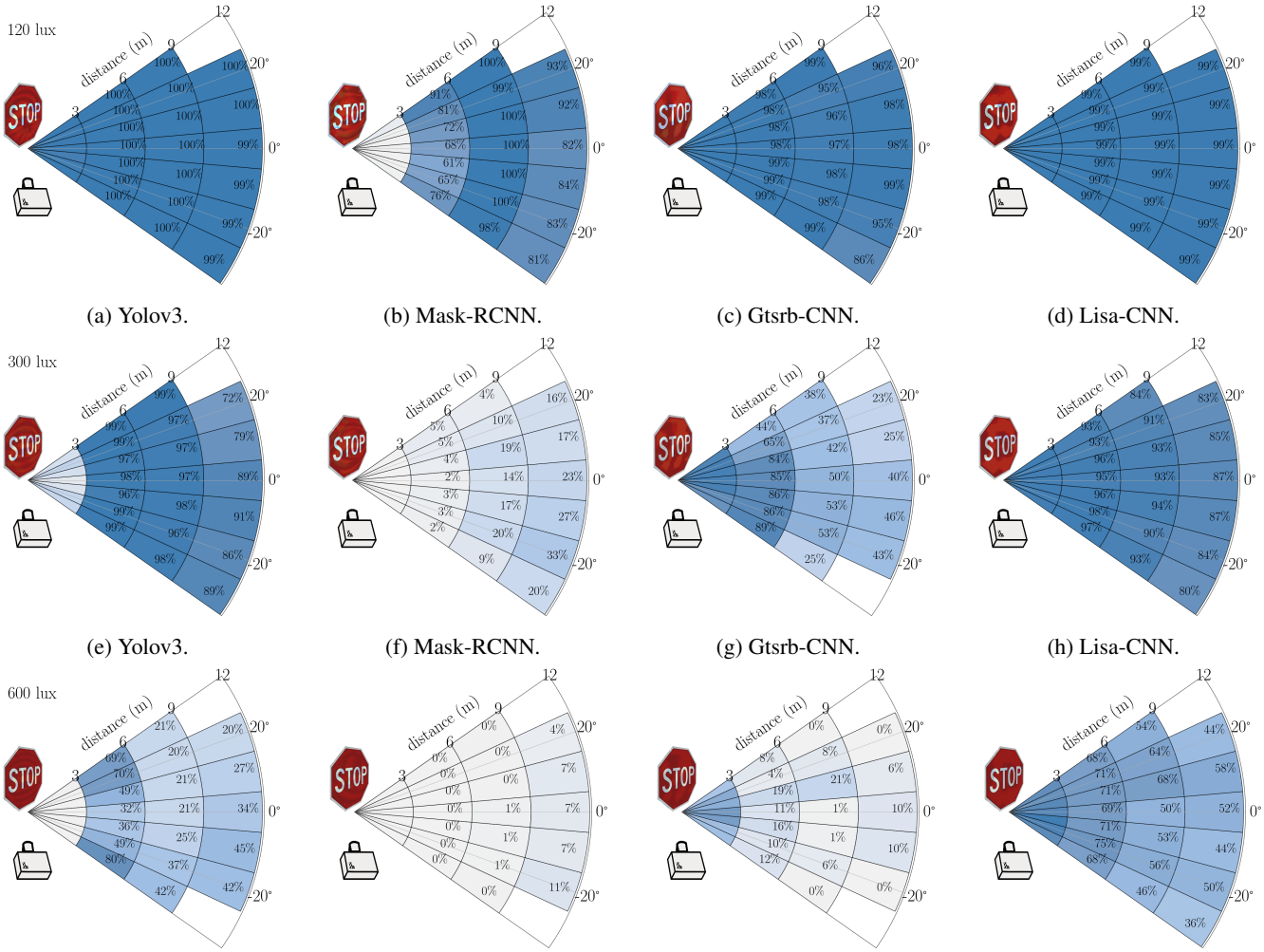
[1] *https://github.com/ssloxford/short-lived-adversarial-perturbations*

5.2 Indoor Results

In this section, we present the results of the detection for the controlled indoor experiment. We also report in Figure 9 the baseline results of using the networks to detect/classify the stop sign, by recording videos of the stop sign unaltered. Figure 9 shows that all networks work quite well in non-adversarial conditions, with the exception of Lisa-CNN which shows a few misdetections.

We report in Figure 10 the results of the detection for the 120, 300 and 600 lux setting for the different networks, as the percentage of frames where the stop sign was not detected by the network. The minimum number of frames tested for a single model is 3,438, see Table 5 for exact figures. Figure 10 shows that the attack is extremely successful in dimmer lighting conditions, obtaining >99% success rate for all networks except Mask-RCNN, which presents additional resilience at shorter distances. The figure also shows how our method is able to create AE that generalize extremely well across all the measured distances from 1 to 12m and viewing angles -30° to 30°. As the ambient increases, the success rate quickly decreases accordingly. Already at 300 lux, the attack success rate is greatly reduced for Mask-RCNN and Gtsrb-CNN, while Yolov3 and Lisa-CNN still remain vulnerable, but the attack degradation becomes evident at 600 lux.

Overall, we found that Mask-RCNN is consistently more resilient than the other networks in the detection. In 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 could be a combination of Mask-RCNN learning more robust features for the detection (possibly thanks to the higher model complexity) and of using a region proposal network for the detection [20]. Nevertheless, such robustness comes at the cost of execution speed: Mask-RCNN requires up to 14 times the execution time of Yolov3 (300ms vs 22ms).



(a) Yolov3. (b) Mask-RCNN. (c) Gtsrb-CNN. (d) Lisa-CNN.

(e) Yolov3. (f) Mask-RCNN. (g) Gtsrb-CNN. (h) Lisa-CNN.

(i) Yolov3. (j) Mask-RCNN. (k) Gtsrb-CNN. (l) Lisa-CNN.

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 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.

5.3 Road Driving Test

To further test the feasibility of the attack, we carry out the attack outdoors in moving vehicle settings.

Setup. The experiment is carried out on a section of private road at our institution. We mount the stop sign at 2m height and set the projector in front of it at a distance of approximately 2 metres. The experiment was conducted shortly prior to sunset in early October, at coordinates 51.7520° N, 1.2577° W. At the time of the experiment the ambient light level measured at the surface of the sign is $\sim$ 120 lux. We use a car to approach the stop sign at 10-15km/h, with the car headlights on during the approach. Videos are recorded using the same iPhone X mounted inside the car at 240fps. We follow the same pipeline described in Section 5.1. However, rather than carrying out the profiling step (Step 2 of the Experimental Procedure), we *re-use* the 120 lux projections that were optimized for the controlled indoor conditions.

Results. We report the results from the driving test in Figure 11, which shows the probability of detection for stop sign as the car approaches the sign. The experiment measures up to 18m away to roughly 7m, when the stop sign exits the video frame (we keep the camera angle fixed during the approach). The results closely match the findings indoor, with the attack being successful for most networks along the whole approach: we obtain 100% success rate for Lisa-CNN and Gtsrb-CNN and over 77% for Mask-RCNN and Yolov3. These results also confirm the generalizability of optimized projections: simply re-using projections without having to re-execute Step 2 and

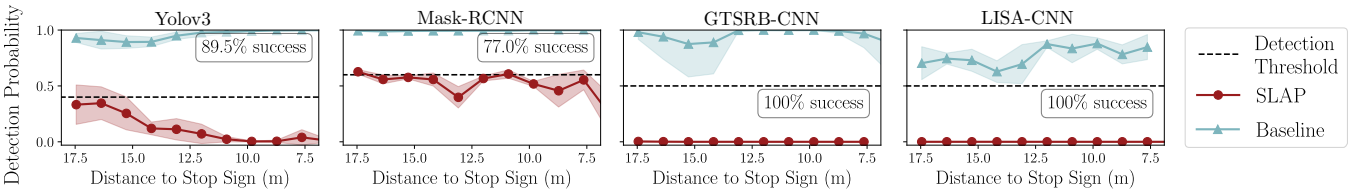


Figure 11: Detection probability for the stop sign during the road driving test. During the test the car approaches the stop sign while the attack is being carried out, the ambient light during the measurements is ∼120 lux, the car headlights are on. The data are grouped into 10 distance bins, the shaded areas indicate the standard deviation of the probability within that distance bin.

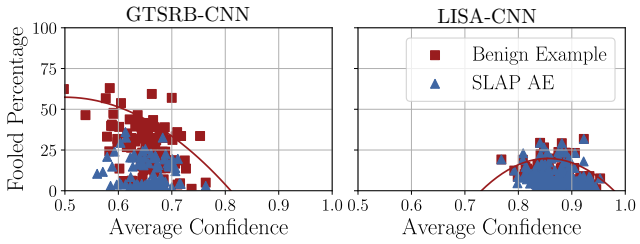


Figure 12: Visualization of the SentiNet detection results (from the 180 lux setting). The plot shows that the SLAP AE have a behaviour similar to benign examples across the two dimensions used by SentiNet, preventing detection.

Step 3 of the experimental procedure at the time of attack led to similar success rates. This means that adversaries could easily pre-compute a set of projections and quickly swap between them depending on the current light conditions.

5.4 Defences

Generally, AE defences are aimed at detecting AE in a digital scenario, where adversaries have the capability to arbitrarily manipulate inputs, but are limited to an L_p-norm constraint. In the case of physical AE, adversaries are not directly limited by an L_p-norm constraint but by the physical realizability of their AE. Defences that are tailored to physical AE have not been investigated as much as general-scope AE defences. For this reason, we specifically choose to evaluate our AE against Sentinet [13]: it is one of the few published works that addresses physical AE detection. Additionally, we evaluate our attack against two other defences which could be used in the autonomous driving scenario as they do not entail additional running time: the input randomization by Xie et al. [42] and adversarial learning [40]. In the following we describe our evaluation setup and results.

Setup and Remarks. We evaluate the three considered AE defences applied to Gtsrb-CNN and Lisa-CNN, as all three defences are designed to work in image classification scenario; at the time of writing, defences for object detectors are not as well explored. For SentiNet, we use 100 *benign* images taken from the GTSRB and LISA dataset to compute the threshold function, 100 *test* images where we overlay the suspected adversarial regions and 100 random frames containing a SLAP AE from the collected videos. For our Sentinet implementation, we use XRAI [22] to compute saliency masks as the original method used (GradCam [11]) led to too coarse grained masks (see Appendix C). For the input randomization of [42], we set the maximum size of the padded image to be 36 (from 32). For adversarial learning [40], we re-write the Lisa-CNN and Gtsrb-CNN models and we train them on the respective datasets from scratch adding an FGSM-adversarial loss to the optimization. We use Adam with learning rate 0.001, the weight of the adversarial loss is set to 0.2, the FGSM step size to 0.2, we use L_{inf}-norm and train for 50 epochs. Adversarially trained models present a slight accuracy degradation on the test set compared to training them with categorial cross-entropy, Gtsrb-CNN goes from 98.47% to 98.08% (-.39%) while Lisa-CNN from 95.9% to 95.55% (-.35%). For input randomization and adversarial learning we run the inference on all the collected video frames of the experiment.

Results. We report the results in Table 3. The table shows the attack success rate computed as the percentages of frames where a stop sign was not detected. We also report the legitimate attack success for comparison. We found that input randomization does not detect our attack. This is expected given that any type of input augmentation-defence is intrinsically compensated for by our optimization (see Section 4.3). Even worse, such method actually degrades the accuracy of the model, showing that the original models for Lisa-CNN and Gtsrb-CNN taken from [15] were not trained with sufficient data augmentation. As expected, thanks to the larger affected areas of the SLAP AE, these adversarial samples can bypass detection by SentiNet in over 95% of the evaluated frames, with no significant difference across the overlay pattern used (either Random or Checkerboard). We also report a visualization of the threshold function fit in SentiNet in Figure 12, showing that the behaviour of SLAP AE resembles those of normal examples. We found that adversarial learning is a more suitable way to defend against SLAP, stopping a good portion of the attacks. Nevertheless, the fact that we only evaluate an adaptive defender (not an adaptive adversary) and that adversarially-trained models suffer from benign accu-

Network	*Ambient Light* (lx)	*Attack Success*	*Adversarial Learning [40]*	*Input Randomization [42]*	*SentiNet [13]* Random	*SentiNet [13]* Checkerboard
Gtsrb-CNN	120	99.96%	20.23% (-79.73%)	99.55% (-0.40%)	93.43%	95.45%
	180	90.53%	23.57% (-66.97%)	90.02% (-0.51%)	93.19%	93.72%
	300	56.51%	48.18% (-8.33%)	86.78% (+30.27%)	96.97%	96.46%
	440	56.34%	40.24% (-16.10%)	82.96% (+26.61%)	95.81%	96.34%
	600	12.79%	10.91% (-1.88%)	51.37% (+38.58%)	95.29%	95.29%
Lisa-CNN	120	100.00%	0.06% (-99.94%)	100.00% (+0.00%)	94.24%	95.29%
	180	99.95%	0.88% (-99.07%)	99.90% (-0.05%)	100.00%	100.00%
	300	99.81%	0.00% (-99.81%)	99.98% (+0.17%)	94.76%	96.86%
	440	98.44%	0.59% (-97.85%)	99.95% (+1.51%)	100.00%	100.00%
	600	69.05%	0.04% (-69.01%)	95.71% (+26.67%)	95.81%	96.86%

Table 3: Attack success rate across the various evaluated defences, models and lux settings. Figures are reported as the percentage of frames in which the attack is successful, i.e., a stop sign is not detected. Differently, (*) figures for SentiNet are reported as percentage out of the 100 adversarial frames extracted from the videos, both overlaying patterns Random and Checkerboard are reported.

racy degradation (performance of the model with no attack in place) highlights how SLAP still remains a potential threat.

5.5 Attack Transferability

Setup. In this section, we test the transferability of our attack across networks, testing all pairwise combinations of our models, including adversarially trained ones. We also use the Google Vision API [17] 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.

Results. We report the results in Table 4. 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, taken from the videos from the indoor experiment. Table 4 reports success rates of the attack as a percentage of the frames where the stop sign was undetected. Table 4 shows that our attack transfers well for low light settings, but the transferability degrades quickly for the 300 lux setting and above. We find that Mask-RCNN transfers better to Yolov3 compared to the opposite direction, the same happens for Gtsrb-CNN and Lisa-CNN, suggesting that fitting AE on complex models favours the attacker. Table 4 also shows that adversarially trained model have benefits by reducing the transferability of attacks fit on surrogate models.

6 Discussion

In this section, we discuss the attack feasibility.

Attack Feasibility. Our experiments demonstrate that increasing ambient light quickly stops the feasibility of the attack in bright conditions. In practice, during daytime, the attack could be conducted on non-bright days, e.g., dark overcast days or close to sunset or sunrise, when the ambient light is low (<400 lux). Regarding the effect of car headlights, our outdoor experiments show that the car headlights-emitted light is negligible compared to the projection luminosity and does not influence the attack success. While car headlights on high-beam would compromise the projection appearance and degrade the attack success rates, we did not consider these lights to be on as stop signs would be mainly present in urban areas, where high-beam headlights would be off. In general, the amount of projector-emitted light that reaches the sign depends on three factors: (i) the distance between projector and sign, (ii) the throw ratio 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 additional projectors with long throw distance, the Pana-

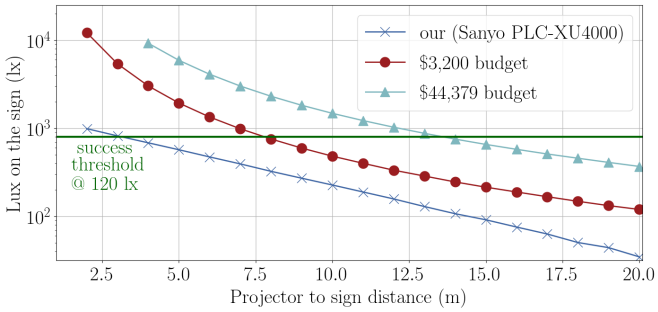


Figure 13: Amount of lux achievable on the stop sign surface for increasing projection distances and different projectors. The horizontal line shows the threshold for success measured in our experiments (800 lux at 120 lux ambient light).

lux	*Source Model*	*no. frames*	*Target Model* Yolov3	Mask-RCNN	Gtsrb-CNN	Gtsrb-CNN$^{(a)}$	Lisa-CNN	Lisa-CNN$^{(a)}$	Google Vision*
120	Yolov3	4587	**100.0%**	73.4%	0.0%	0.0%	21.5%	0.0%	100.0%
	Mask-RCNN	3765	98.7%	**97.1%**	0.0%	0.0%	15.5%	0.0%	100.0%
	Gtsrb-CNN	3760	40.5%	37.0%	**99.9%**	16.1%	51.4%	0.0%	72.4%
	Lisa-CNN	4998	29.4%	28.1%	6.8%	0.0%	**100.0%**	0.0%	77.1%
300	Yolov3	5169	**96.5%**	3.6%	2.5%	0.0%	2.3%	0.0%	72.3%
	Mask-RCNN	3543	32.0%	**14.0%**	0.1%	0.0%	10.4%	0.0%	65.9%
	Gtsrb-CNN	3438	2.0%	2.9%	**48.0%**	43.1%	44.0%	0.0%	47.6%
	Lisa-CNN	4388	0.7%	4.9%	8.6%	0.0%	**100.0%**	0.0%	25.0%
600	Yolov3	5507	**17.8%**	0.2%	32.5%	0.0%	27.4%	0.0%	23.7%
	Mask-RCNN	5058	0.1%	**0.4%**	5.3%	0.0%	4.6%	0.0%	16.7%
	Gtsrb-CNN	4637	0.0%	0.9%	**7.2%**	7.5%	4.9%	0.0%	21.1%
	Lisa-CNN	4714	0.0%	0.9%	8.6%	0.0%	**57.5%**	0.0%	15.8%

Table 4: Transferability results. We test all the frames from the collected videos with a certain projection being shone against a different target model, figures in bold are white-box pairs. (*) For Google Vision we only test one frame every 30 frames, i.e., one per second. We also remove all frames that are further than 6m away as Google Vision does not detect most of them in a baseline scenario. $\_^{(a)}$ indicates adversarially trained models.

sonic PT-RZ570BU and the NEC PH1202HL1, 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 from increasing distances. We consider the success as measured in 120 lux ambient light, where obtaining 800 lux of light on the sign with the projector is sufficient to achieve consistent attack success (see Section 5.1). 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. Additionally, adversaries could also use different lenses to increase the throw ratio of cheaper projectors (similarly to [32]).

Attack Generalizability. We show results for attacks on other objects (give way sign, bottle) in Appendix A, however, to extend the attack to *any* object, the adversary will have to consider the distortion introduced by the projection surface (not necessary for flat traffic signs). The attacker will have to augment the projection model used in this paper with differentiable transformations which model the distortion caused by the non-flat surface. In general, the size of the projectable area limits the feasibility of the attack against certain objects (e.g., hard to project on a bike); this drawback is shared across all vectors that create physically robust AE, including adversarial patches. We also found that the properties of the material where the projection is being shone will impact the attack success: traffic signs are an easier target because of their high material reflectivity. When executing the attack on other objects, we found that certain adaptations lead to marginal attack improvements, in particular context information (e.g., the pole for the stop sign, the table where the bottle is placed). Generally, for object detectors, adversaries will have to tailor certain parameters of the optimization to the target object.

7 Conclusions

In this paper we presented SLAP, a new attack vector to realize short-lived physical adversarial examples by using a light projector. We investigate the attack in the context of road safety, where the attacker's goal is to change the appearance of a stop sign by shining a crafted projection onto it so that it is undetected by the DNNs mounted on autonomous vehicles.

Given the non-trivial physical constraints of projecting specific light patterns on various materials in various conditions, we proposed a method to generate projections based on fitting a predictive three-way color model and using an AE generation pipeline that enhances the AE robustness. We evaluated the proposed attack in a variety of light conditions, including outdoors, and against state-of-the-art object detectors Yolov3 and Mask-RCNN and traffic sign recognizers Lisa-CNN and Gtsrb-CNN. Our results show that SLAP generates AEs that are robust in the real-world. We evaluated defences, highlighting how existing defences tailored to physical AE will not work against AE generated by SLAP, while finding that an adaptive defender using adversarial learning can successfully hamper the attack effect, at the cost of reduced accuracy.

Nevertheless, the novel capability of modifying how an object is detected by DNN models, combined with the capability of carrying out opportunistic attacks, makes SLAP a powerful new attack vector that requires further investigation. This paper makes an important step towards increasing the awareness and further research of countermeasures against light-projection adversarial examples.

Acknowledgements

This work was supported by grants from armasuisse, Mastercard, and by the Engineering and Physical Sciences Research Council [grant numbers EP/N509711/1, EP/P00881X/1].

References

[1] "Daylight", [Online] Accessed: 2020-02-20. https://en.wikipedia.org/wiki/Daylight.

[2] "JPEG File Interchange Format", [Online] Accessed: 2020-01-15. http://www.w3.org/Graphics/JPEG/jfif3.pdf.

[3] "Manual of Uniform Traffic Control Devices for Street and Highways", [Online] Accessed: 2020-01-08. http://mutcd.fhwa.dot.gov/pdfs/2009r1r2/mutcd2009r1r2edition.pdf.

[4] "Panasonic PT-AE8000 Projector", [Online] Accessed: 2020-10-12. http://www.projectorcentral.com/Panasonic-PT-AE8000.htm.

[5] "Sanyo PLC-XU4000 Projector", [Online] Accessed: 2020-10-12. http://www.projectorcentral.com/Sanyo-PLC-XU4000.htm.

[6] Apollo. "ApolloAuto - An open autonomous driving platform", [Online] Accessed: 2021-02-19. http://github.com/apolloauto.

[7] Aptina. "1/3-Inch CMOS Digital Image Sensor AR0132AT Data Sheet", [Online] Accessed: 2021-02-19. http://datasheetspdf.com/pdf/829321/AptinaImagingCorporation/AR0132AT/1f.

[8] BMW. "BMW TechOffice Munich", [Online] Accessed: 2020-02-19. http://github.com/BMW-InnovationLab.

[9] Andy Boxall. "From robots to projection mapping: Inside Panasonic's Tokyo 2020 Olympic tech", [Online] Accessed: 2021-02-19. http://www.digitaltrends.com/mobile/panasonic-tokyo-2020-technology-interview/.

[10] Tom Brown, Dandelion Mane, Aurko Roy, Martin Abadi, and Justin Gilmer. "Adversarial Patch". *arXiv preprint arXiv:1712.09665v2*, 2018.

[11] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. "Grad-cam++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks". In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018.

[12] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. "Shapeshifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector". In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68, 2018.

[13] E. Chou, F. Tramèr, and G. Pellegrino. "SentiNet: Detecting Localized Universal Attacks Against Deep Learning Systems". In *Proceedings of the IEEE Security and Privacy Workshops (SPW)*, pages 48–54, 2020.

[14] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. "Exploring the Landscape of Spatial Robustness". In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1802–1811, 2019.

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

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[17] Google. "Google Vision API", Accessed: 2020-10-12. https://cloud.google.com/vision.

[18] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain". *arXiv preprint arXiv:1708.06733*, 2017.

[19] Phil Hall. "The Exposure Triangle: Aperture, Shutter Speed and ISO explained", [Online] Accessed: 2021-02-19. http://www.techradar.com/uk/how-to/the-exposure-triangle.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask R-CNN". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2969, 2017.

[21] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. "Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark". In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013.

[22] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. "Xrai: Better Attributions through Regions". In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 4948–4957, 2019.

[23] Danny Karmon, Daniel Zoran, and Yoav Goldberg. "Lavan: Localized and visible adversarial noise". In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2507–2515, 2018.

[24] Sebastian Köhler, Giulio Lovisotto, Simon Birnbach, Richard Baker, and Ivan Martinovic. "They See Me Rollin': Inherent Vulnerability of the Rolling Shutter in CMOS Image Sensors". *arXiv preprint arXiv:2101.10011*, 2021.

[25] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. "Adversarial Examples in the Physical World". *arXiv preprint arXiv:1607.02533*, 2016.

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature Pyramid Networks for Object Detection". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft COCO: Common Objects in Context". In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[28] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Authors Yingqi Liu, Weihang Wang, and Xiangyu Zhang. "Trojaning Attack on Neural Networks". In *Proceedings of the Network and Distributed System Symposium (NDSS)*, 2018.

[29] Giulio Lovisotto, Simon Eberz, and Ivan Martinovic. "Biometric Backdoors: A Poisoning Attack Against Unsupervised Template Updating". In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 184–197, 2020.

[30] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. "Discriminative Correlation Filter with Channel and Spatial Reliability". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6309–6318, 2017.

[31] Aravindh Mahendran and Andrea Vedaldi. "Understanding Deep Image Representations by Inverting Them". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, 2015.

[32] Yanmao Man, Ming Li, and Ryan Gerdes. "GhostImage: Perception Domain Attacks against Vision-based Object Classification Systems". *arXiv preprint arXiv:2001.07792*, 2020.

[33] Massimo Mancuso and Sebastiano Battiato. "An Introduction to the Digital Still Camera Technology". *ST Journal of System Research*, 2(2), 2001.

[34] Andreas Mogelmose, Mohan Manubhai Trivedi, and Thomas B. Moeslund. "Vision-based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey". *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012.

[35] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". *arXiv preprint arXiv:1804.02767*, 2018.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

[37] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. "Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition". In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1528–1540, 2016.

[38] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. "Physical Adversarial Examples for Object Detectors". In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, 2018.

[39] David Stone. "Spotlight on Lumens: How They're Measured, and Why They're Not All the Same", [Online] Accessed: 2020-02-20. http://www.projectorcentral.com/Lumens-Explained.htm.

[40] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Explaining and Harnessing Adversarial Examples". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[41] Adam Van Etten. "You Only Look Twice: Rapid Multi-Scale Object Detection in Satellite Imagery". *arXiv preprint arXiv:1805.09512*, 2018.

[42] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. "Mitigating Adversarial Effects Through Randomization". *arXiv preprint arXiv:1711.01991*, 2017.

[43] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. "Seeing isn't Believing: Towards More Robust Adversarial Attack against Real World Object Detectors". In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 1989–2004, 2019.

A Attack on Different Objects

The introduced attack can generalize in principle to any kind of deep neural network which uses RGB-camera inputs to make decisions. To show how our attack generalizes, we also investigate the feasibility of the attack on different objects.

Setup. For Lisa-CNN and Gtsrb-CNN, we choose another traffic sign, "give way", while for Yolov3 and Mask-RCNN we choose the "bottle" class. For the give way sign and the bottle, we run a reduced evaluation: we execute all the experiment procedure steps reported in Section 5.1 and we test the correct (mis-)classification across a set of photos of the altered objects. Extending our method to other objects is straightforward, it only requires to change the input mask of the projection and re-profile the projectable colors. When projecting on non-flat surfaces, the adversary will also have to consider the distortion introduced by those surfaces, this is briefly discussed in Section 6.

Results. We report example frames of successful attack on other objects in Figure 15 and in Figure 16. These include legitimate frames where the classification works correctly. All the pictures are taken in 180 lux ambient light. For Mask-RCNN and Yolov3 we restricted the bottle size to [150, 250], meaning that the bottle is generally in the foreground.

B Additional Results

We report extended results for the transferability-based cross-network attack in Table 5. This includes each pair of the evaluated models, including Lisa-CNN$^{(s)}$ and Gtsrb-CNN$^{(s)}$ which are trained with cross-entropy loss from scratch. We report an example frame from the outdoor experiment in Figure 14.

C SentiNet Description

Rationale. We picked SentiNet for the evaluation because it was one of the few defences that was *specifically* designed to detect physical adversarial examples (AE). In fact, there is a plethora of works that creates physical adversarial examples by using stickers (or patches) that are placed on the targeted objects. The insight behind SentiNet is that these patches are the most common way to create physical AE, but generate small image areas with large saliency. This is not only a

Figure 14: Sample video frame extracted from the outdoor experiment, for the 120 lux setting. The image shows the stop sign undetected under the threshold set in the experiments. The blue 'tracker' box is set manually and tracks the location of the sign.

detectable behavior in general, but it is also *unavoidable* for the attacker to escape such behavior when creating a physical AE (without replacing the entire object).

Description. SentiNet is a system designed to detect AE leveraging the intuition of locality of patches. 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 onto them. To compute the salient areas in input SentiNet uses GradCam++ [11], 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 [22], a newer and more accurate method to compute salient areas. We found that using GradCam made the output masks unusable as a resolution of 4x4 leads to coarse block like regions where salient areas cannot be accurately identified (resolution of this layer also is pointed out as a problem in the original paper [13]). 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 [22].

SentiNet computes a threshold function which separates AE from benign images. The threshold function is computed using: (i) 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 (ii) 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 benign data. Our SentiNet implementation is also available with the rest of the source code in the project repository.

lux	*Source Model*	*no. frames*	*Target Model*								
			Yolov3	*Mask-RCNN*	*Gtsrb-CNN*	*Gtsrb-CNN$^{(a)}$*	*Gtsrb-CNN$^{(s)}$*	*Lisa-CNN*	*Lisa-CNN$^{(a)}$*	*Lisa-CNN$^{(s)}$*	*Google Vision**
120	Yolov3	4587	**100.0%**	73.4%	0.0%	0.0%	0.0%	21.5%	0.0%	0.0%	100.0%
	Mask-RCNN	3765	98.7%	**97.1%**	0.0%	0.0%	0.0%	15.5%	0.0%	0.0%	100.0%
	Gtsrb-CNN	3760	40.5%	37.0%	**99.9%**	0.0%	16.1%	51.4%	0.0%	0.0%	72.4%
	Lisa-CNN	4998	29.4%	28.1%	6.8%	0.0%	0.0%	**100.0%**	0.0%	0.0%	77.1%
180	Yolov3	7862	**99.9%**	4.0%	14.6%	0.0%	0.0%	17.5%	0.0%	0.0%	90.6%
	Mask-RCNN	4083	96.3%	**91.0%**	0.2%	0.0%	0.0%	54.8%	0.0%	0.0%	98.9%
	Gtsrb-CNN	7426	12.3%	2.0%	**85.7%**	0.0%	27.7%	13.4%	0.0%	0.0%	44.4%
	Lisa-CNN	6268	9.0%	0.6%	35.7%	0.0%	0.0%	**100.0%**	0.0%	0.0%	26.2%
300	Yolov3	5169	**96.5%**	3.6%	2.5%	0.0%	0.0%	2.3%	0.0%	0.0%	72.3%
	Mask-RCNN	3543	32.0%	**14.0%**	0.1%	0.0%	0.0%	10.4%	0.0%	0.0%	65.9%
	Gtsrb-CNN	3438	2.0%	2.9%	**48.0%**	0.0%	43.1%	44.0%	0.0%	0.0%	47.6%
	Lisa-CNN	4388	0.7%	4.9%	8.6%	0.0%	0.0%	**100.0%**	0.0%	0.0%	25.0%
440	Yolov3	6716	**49.5%**	0.8%	40.3%	0.0%	0.0%	40.9%	0.0%	0.0%	35.3%
	Mask-RCNN	6023	5.4%	**3.3%**	41.1%	0.0%	0.0%	35.6%	0.0%	0.0%	33.6%
	Gtsrb-CNN	6565	0.7%	0.7%	**43.7%**	0.0%	44.1%	35.6%	0.0%	0.0%	33.1%
	Lisa-CNN	6287	1.0%	2.4%	26.4%	0.0%	0.0%	**97.4%**	0.0%	0.0%	26.8%
600	Yolov3	5507	**17.8%**	0.2%	32.5%	0.0%	0.0%	27.4%	0.0%	0.0%	23.7%
	Mask-RCNN	5058	0.1%	**0.4%**	5.3%	0.0%	0.0%	4.6%	0.0%	0.0%	16.7%
	Gtsrb-CNN	4637	0.0%	0.9%	**7.2%**	0.0%	7.5%	4.9%	0.0%	0.0%	21.1%
	Lisa-CNN	4714	0.0%	0.9%	8.6%	0.0%	0.0%	**57.5%**	0.0%	0.0%	15.8%

Table 5: Transferability results. We test all the frames from the collected videos with a certain projection being shone against a different target model, figures in bold are white-box pairs. (*) For Google Vision we only test one frame every 30 frames, i.e., one per second. We also remove all frames that are further than 6m away as Google Vision does not detect most of them in a baseline scenario. $\_^{(a)}$ indicates adversarially trained models. $\_^{(s)}$ indicates models we re-trained from scratch.

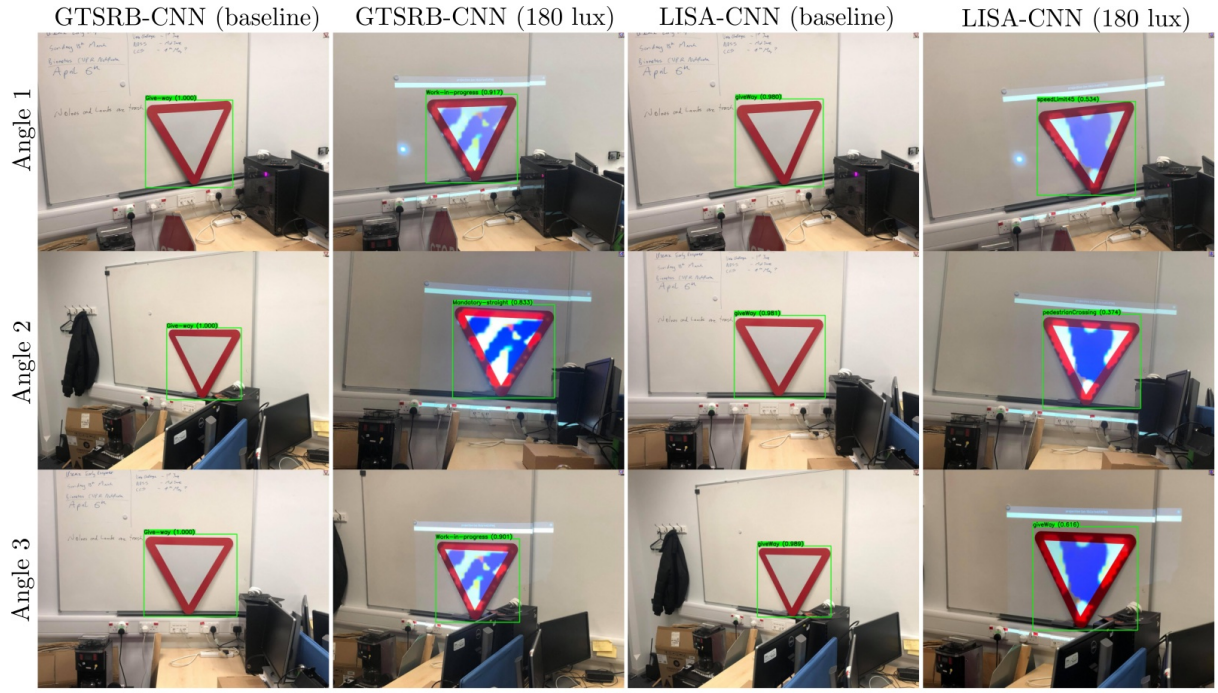


Figure 15: Attack on class “Give Way” for Gtsrb-CNN and Lisa-CNN.

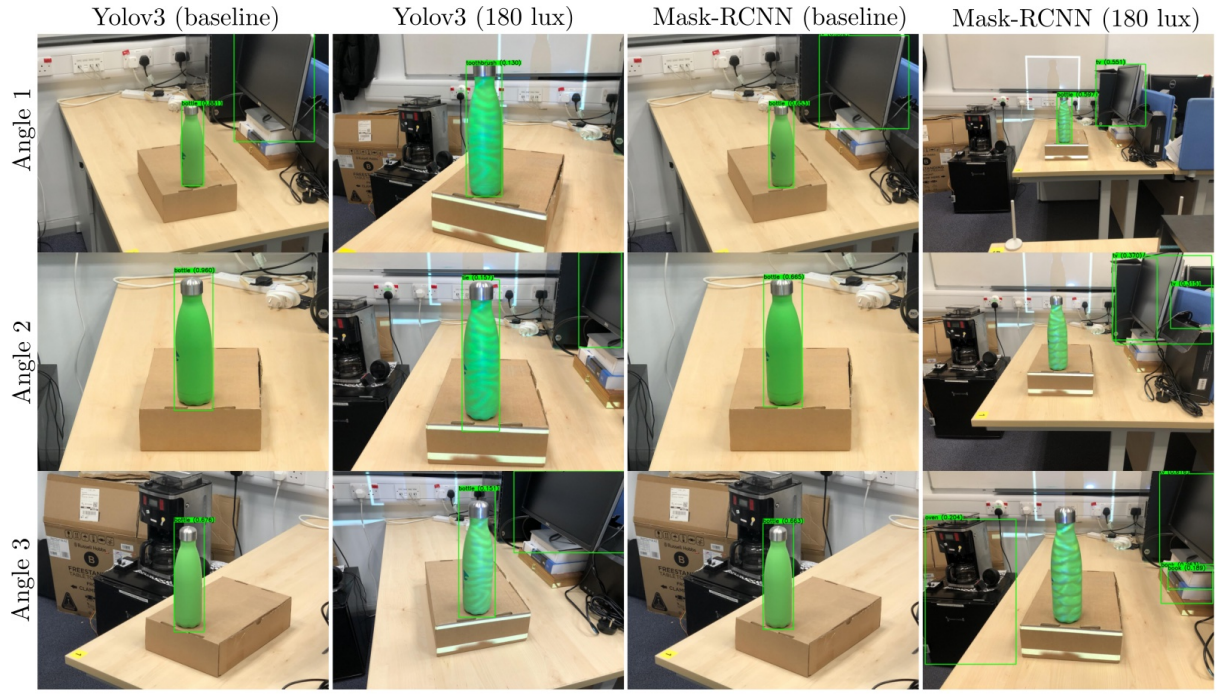


Figure 16: Attack on class “Bottle” for Yolov3 and Mask-RCNN. The detection thresholds used in the paper are 0.4 and 0.6, respectively.