Physical Adversarial Examples for Object Detectors

Kevin Eykholt[1], Ivan Evtimov[2], Earlence Fernandes[2], Bo Li[3],

Amir Rahmati[4,6], Florian Tramèr[5], Atul Prakash[1], Tadayoshi Kohno[2], Dawn Song[3]

[1]University of Michigan
[2]University of Washington
[3]University of California, Berkeley
[4]Stony Brook University
[5]Stanford University
[6]Samsung Research America

Abstract

Deep neural networks (DNNs) are vulnerable to *adversarial examples*—maliciously crafted inputs that cause DNNs to make incorrect predictions. Recent work has shown that these attacks generalize to the physical domain, to create perturbations on physical objects that fool image classifiers under a variety of real-world conditions. Such attacks pose a risk to deep learning models used in safety-critical cyber-physical systems.

In this work, we extend physical attacks to more challenging *object detection models*, a broader class of deep learning algorithms widely used to detect and label multiple objects within a scene. Improving upon a previous physical attack on image classifiers, we create perturbed physical objects that are either ignored or mislabeled by object detection models. We implement a *Disappearance Attack*, in which we cause a Stop sign to "disappear" according to the detector—either by covering the sign with an adversarial Stop sign poster, or by adding adversarial stickers onto the sign. In a video recorded in a controlled lab environment, the state-of-the-art YOLO v2 detector failed to recognize these adversarial Stop signs in over 85% of the video frames. In an outdoor experiment, YOLO was fooled by the poster and sticker attacks in 72.5% and 63.5% of the video frames respectively. We also use Faster R-CNN, a different object detection model, to demonstrate the *transferability* of our adversarial perturbations. The created poster perturbation is able to fool Faster R-CNN in 85.9% of the video frames in a controlled lab environment, and 40.2% of the video frames in an outdoor environment. Finally, we present preliminary results with a new *Creation Attack*, wherein innocuous physical stickers fool a model into detecting nonexistent objects.

1 Introduction

Deep neural networks (DNNs) are widely applied in computer vision, natural language, and robotics, especially in safety-critical tasks such as autonomous driving [9]. At the same time, DNNs have been shown to be vulnerable to *adversarial examples* [3, 6, 7, 14, 17], maliciously perturbed inputs that cause DNNs to produce incorrect predictions. These attacks pose a risk to the use of deep learning in safety- and security-critical decisions. For example, an attacker can add perturbations, which are negligible to humans, to a Stop sign and cause a DNN embedded in an autonomous vehicle to misclassify or ignore the sign.

Early works studied adversarial examples in the digital space only. However, it has recently been shown that it is also possible to create perturbations that survive under various physical conditions (*e.g.,* object distance, pose, lighting, etc.) [1, 2, 4, 8, 20]. These works focus on attacking *classification networks*, *i.e.,* models that produce a single prediction on a static input image. In this work, we start exploring physical adversarial examples for *object detection networks*, a richer class of deep learning algorithms that can detect and label multiple objects in a scene. Object detection networks are a popular tool for tasks that require real-time and dynamic recognition of surrounding objects, autonomous driving being a canonical application. Object detectors are known to be vulnerable to digital attacks [22], but their vulnerability to physical attacks remains an open question.

Compared to classifiers, object detection networks are more challenging to attack: 1) Detectors process an entire scene instead of a single localized object. This allows detectors to use contextual information (*e.g.,* the orientation and relative position of objects in the scene) to generate predictions. 2) Detectors are not limited to producing a single prediction. Instead, they label every recognized object in a scene, usually by combining predictions of the *location* of objects in a scene, and of the labeling of these objects. Attacks on object detectors need to take both types of predictions (presence/absence of an object and nature of the object) into account, whereas attacks on classifiers only focus on modifying the label of a single (presumably present) object.

To create proof-of-concept attacks for object detectors, we start from the existing Robust Physical Perturbations (RP_2) algorithm [4] of Eykholt *et al.*, which was originally proposed to produce robust physical attacks on image classifiers. The approach taken by Eykholt *et al.* (as well as by others [1, 8]) is to sample from a distribution that mimics physical perturbations of an object (*e.g.,* view distance and angle), and find a perturbation that maximizes the probability of mis-classification under this distribution. We find that the physical perturbations considered in their work are insufficient to extend to object detectors.

Indeed, when working with image classifiers, prior works considered target objects that make up a large portion of the image and whose relative position in the image varies little. Yet, when performing object detection in a dynamic environment such as a driving car, the relative size and position of the multiple objects in a scene can change drastically. These changes produce additional constraints that have to be taken into account to produce successful robust physical attacks. Many object detectors, for instance, split a scene into a grid or use a sliding window to identify regions of interest, and produce separate object predictions for each region of interest. As the relative position of an object changes, the grid cells the object is contained in (and the corresponding network weights) change as well. Robust perturbations, thus, have to be applicable to multiple grid cells simultaneously. We show that robustness to these physical modifications can be attained by extending the distribution of inputs considered by Eykholt *et al.* to account for additional synthetic transformations to objects in a scene (*e.g.,* changes in perspective, size, and position).

Following Eykholt *et al.*, we consider physical adversarial attacks on the detection and classification of Stop signs, an illustrative example for the safety implications of a successful attack. The perturbations, while large enough to be visible to the human eye, are constrained to resemble human-made graffiti or subtle lighting artifacts that could be considered benign. We consider an untargeted attack specific to object detectors, which we refer to as a *Disappearance Attack*. In a Disappearance Attack, we create either an adversarial poster or physical stickers applied to a real Stop sign (see Figure 2), which causes the sign to be ignored by an object detector in different scenes with varying object distance, location, and perspective. This attack is analogous to the one considered by Eykholt *et al.* for image classifiers, but targets a richer class of deep neural networks.

We further introduce a new *Creation Attack*, wherein physical stickers that humans would ignore as being inconspicuous can cause an object detector into recognizing nonexistent Stop signs. This attack differs from prior attacks that attempt to fool a network into mis-classifying one object into another, in that it creates an entirely new object classification. Specifically, we experiment with creating adversarial stickers (similar to the ones considered in [2]). Such stickers could for instance be used to mount *Denial of Service* attacks on road-sign detectors.

For our experiments, we target the state-of-the-art YOLO v2 (You Only Look Once) object detector [16]. YOLO v2 is a deep convolutional neural network that performs real-time object detection for 80 object classes. Our indoor (laboratory) and outdoor experiments show that up to distances of 30 feet from the target object, detectors can be tricked into *not* perceiving the attacker's target object using poster and sticker perturbations.

Our Contributions:

- We extend the RP_2 algorithm of Eykholt *et al.* to provide proof-of-concept attacks for object detection networks, a richer class of DNNs than image classifiers.
- Using our new and improved algorithm, we propose a new physical attack on object detection networks: the Disappearance Attack that cause physical objects to be ignored by a detector.
- We evaluate our attacks on the YOLO v2 object detector in an indoor laboratory setting and an outdoor setting. Our results show that our adversarial poster perturbation fools YOLO v2 in 85.6% of the video frames recorded in an indoor lab environment and in 72.5% of the video frames recorded in an outdoor environment. Our adversarial stickers fool YOLO v2 in 85% of the video frames recorded in a laboratory environment and in 63.5% of the video frames recorded in an outdoor environment.
- We evaluate the transferability of our attacks using the Faster R-CNN object detector in laboratory and outdoor environments. Our results show that our attacks fool Faster R-CNN in 85.9% of the video frames recorded in a laboratory environment and in 40.2% of the video frames recorded in an outdoor environment.
- We propose and experiment with a new type of *Creation* attack, that aims at fooling a detector into recognizing adversarial stickers as non-existing objects. Our results with this attack type are preliminary yet encouraging.

Our work demonstrates that physical perturbations are effective against object detectors, and leaves open some future questions: 1) Generalization to other physical settings (*e.g.,* moving vehicles, or even real autonomous vehicles). 2) Further exploration of other classes of attacks:

Our work introduces the disappearance and creation attacks which use posters or stickers, yet there are other plausible attack types (*e.g.,* manufacturing physical objects that are not recognizable to humans, but are recognized by DNNs). 3) Physical attacks on segmentation networks. We envision that future work will build on the findings presented here, and will create attacks that generalize across physical settings (*e.g.,* real autonomous vehicles), and across classes of object detection networks (*e.g.,* semantic segmentation [22]).

2 Related Work

Adversarial examples for deep learning were first introduced by Szegedy *et al.* [21]. Since their seminal work, there have been several works proposing more efficient algorithms for generating adversarial examples [3, 6, 12, 14]. All of these works assume that the attacker has "digital-level" access to an input, *e.g.,* that the attacker can make arbitrary pixel-level changes to an input image of a classifier. For uses of deep learning in cyber-physical systems (*e.g.,* in an autonomous vehicle), these attacks thus implicitly assume that the adversary controls a DNN's input system (*e.g.,* a camera). A stronger and more realistic threat model would assume that the attacker only controls the physical layer, *e.g.,* the environment or objects that the system interacts with, but not the internal sensors and data pipelines of the system.

This stronger threat model was first explored by Kurakin *et al.* They generated physical adversarial examples by printing digital adversarial examples on paper [8]. In their work, they found that a significant portion of the printed adversarial examples fooled an image classifier. However, their experiments were done without any variation in the physical conditions such as different viewing angles or distances.

Athalye *et al.* improved upon the work of Kurakin *et al.* by creating adversarial objects that are robust to variations in viewing angle [1]. To account for such variations, they model small scale transformations synthetically when generating adversarial perturbations. They demonstrate several examples of adversarial objects that fool their target classifiers, but it is not clear how many transformations their attack is robust to. In their paper, they state their algorithm is robust to rotations, translations, and noise and suggest their algorithm is robust so long as the transformation can be modeled synthetically.

Eykholt *et al.* also proposed an attack algorithm capable of generating physical adversarial examples [4]. Unlike Athalye *et al.*, they choose to model image transformations both synthetically and physically. Certain image transformations, such as changes in viewing angle and distance, are captured in their victim dataset. They apply other image transformations, such as lighting, synthetically when generating adversarial examples. Their work suggests that sole reliance on synthetic transformations can miss subtleties in the physical environment, thus resulting in a less robust attack. Different from all prior work that focused on *classifiers*, our work focuses on the broader class of object detection models. Specifically, we extend the algorithm of Eykholt *et al.* using synthetic transformations (perspective, position, scale) to attack object detection models.

Lu *et al.* performed experiments using adversarial road signs printed on paper with the YOLO object detector [11]. Their results suggested that it is very challenging to fool YOLO with physical adversarial examples. Our work resolves the challenges and shows that existing algorithms can be adapted to produce physical attacks on object detectors in highly variable environmental conditions.

3 Background on Object Detectors

Object classification is a standard task in computer vision. Given an input image and a set of class labels, the classification algorithm outputs the most probable label (or a probability distribution over all labels) for the image. Object classifiers are limited to categorizing a single object per image. If an image contains multiple objects, the classifier only outputs the class of the most dominant object in the scene. In contrast, object detectors both locate and classify multiple objects in a given scene.

The first proposed deep neural network for object detection was Overfeat [18], which combined a sliding window algorithm and convolution neural networks. A more recent proposal, Regions with Convolutional Neural Networks (R-CNN) uses a search algorithm to generate region proposals, and a CNN to label each region. A downside of R-CNN is that the region proposal algorithm is too slow to be run in real-time. Subsequent works—Fast R-CNN [5] and Faster R-CNN [19]—replace this inefficient algorithm with a more efficient CNN.

The above algorithms treat object detection as a two-stage problem consisting of region proposals followed by classifications for each of these regions. In contrast, so-called "single shot detectors" such as YOLO [15] (and the subsequent YOLO v2 [16]) or SSD [10] run a single CNN over the input image to jointly produce confidence scores for object localization and classification. As a result, these networks can achieve the same accuracy while processing images much faster. In this work, we focus on YOLO v2, a state-of-the-art object detector with real-time detection capabilities and high accuracy.

The classification approach of YOLO v2 is illustrated in Figure 1. A single CNN is run on the full input image and predicts object location (bounding boxes) and label confidences for 361 separate grid cells (organized into a

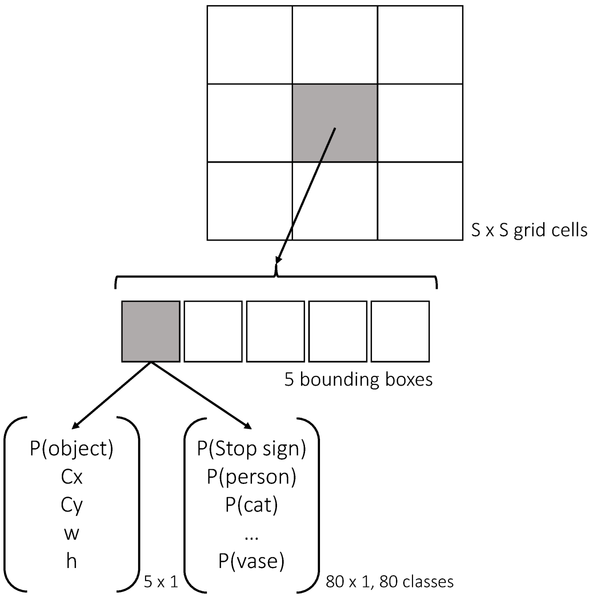


Figure 1: For an input scene, the YOLO v2 CNN outputs a $19 \times 19 \times 425$ tensor. To generate this tensor, YOLO divides the input image into a square grid of S^2 cells ($S = 19$). For each grid cell, there are B bounding boxes ($B = 5$). Each bounding box predicts 5 values: probability of an object in the cell, co-ordinates of the bounding box (center x, center y, width, height). Additionally, for each bounding box the model predicts a probability distribution over all 80 output classes.

19×19 square over the original image). For each cell, YOLO v2 makes a prediction for 5 different boxes. For each box, the prediction contains the box confidence (the probability that this box contains an object), its location and the probability of each class label for that box. A box is discarded if the product of the box confidence and the probability of the most likely class is below some threshold (this threshold is set to 0.1 in our experiments). Finally, the *non-max suppression* algorithm is applied in a post-processing phase to discard redundant boxes with high overlap [16].

Such an object detection pipeline introduces several new challenges regarding physical adversarial examples: First, unlike classification where an object is always assumed present and the attack only needs to modify the class probabilities, attacks on a detector network need to control a combination of box confidences and class probabilities for all boxes in all grid cells of the input scene. Second, classifiers assume the object of interest is centered in the input image, whereas detectors can find objects at arbitrary positions in a scene. Finally, the object's size in the detector's input is not fixed. In classification, the image is usually cropped and resized to focus on the object being classified. Object detectors are meant to reliably detect objects at multiple scales, distances and angles in a scene.

These challenges mainly stem from object detectors being much more flexible and broadly applicable than standard image classifiers. Thus, albeit harder to attack, object detectors also represent a far more interesting attack target than image classifiers, as their extra flexibility makes them a far better candidate for use in reliable cyber-physical systems.

4 Physical Adversarial Examples for Object Detectors

We will first summarize the original RP_2 algorithm, before discussing the modifications necessary to adapt the algorithm to attack object detectors.

4.1 The RP_2 Algorithm

The RP_2 algorithm proposed by Eykholt *et al.* optimizes the following objective function:

$$\begin{aligned}\underset{\delta}{\operatorname{argmin}}\ & \lambda||M_x \cdot \delta||_p + NPS(M_x \cdot \delta) \\ & + \mathbb{E}_{x_i \sim X^V} J(f_\theta(x_i + T_i(M_x \cdot \delta)), y^*)\end{aligned} \tag{1}$$

The first term of the objective function is the ℓ_p norm (with scaling factor λ) of the perturbation δ masked by M_x. The mask is responsible for spatially constraining the perturbation δ to the surface of the target object. For example, in Figure 2, the mask shape is two horizontal bars on the sign.

The second term of the objective function measures the printability of an adversarial perturbation. Eykholt *et al.* borrow this term from prior work [20]. The printability of a perturbation is affected by two factors. First, the colors the computed perturbation must reproduce. Modern printers have a limited color gamut, thus certain colors that appear digitally may not be printable. Second, a printer may not faithfully reproduce a color as it is shown digitally (see Figure 3).

The last term of the objective function is the value of the loss function, $J(\cdot,\cdot)$ averaged across all of the images sampled from X^V. In practice, this is a set of victim images. The victim dataset is composed of multiple images of the object taken under a variety of physical conditions such as changes in viewing angle, viewing distance and lighting. T_i is an "alignment function" that applies a digital transformation that mimics the physical conditions of victim object x_i. For example, if the victim object x_i is a rotated version of the "canonical" target object, then the perturbation $M_x \cdot \delta$ should also be rotated appropriately.

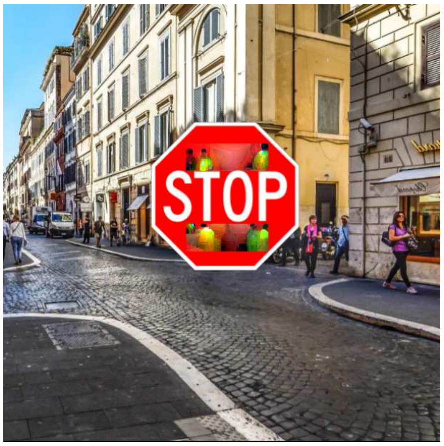
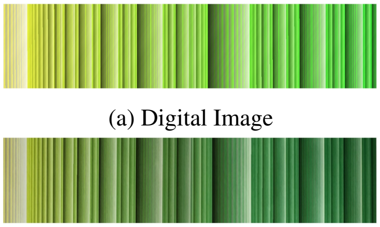


Figure 2: An example of an adversarial perturbation overlaid on a synthetic background. The Stop sign in the image is printed such that it is the same size as a U.S. Stop sign. Then, we cut out the two rectangle bars, and use the original print as a stencil to position the cutouts on a real Stop sign.



(a) Digital Image

(b) Printer Result of Digital Image

Figure 3: The image in (a) shows the image as it is stored digitally. The result of printing and taking a picture of the image in (a) is shown in (b).

Thus, to simulate physical consistency of the perturbed object, we apply the alignment function T_i to the masked perturbation. $f_\theta(\cdot)$ is the output of the classifier network, and y^* is the adversarial target class.

4.2 Extensions to RP$_2$ for Object Detectors

Our modified version of RP$_2$ contains three key differences from the original algorithm proposed by Eykholt *et al.* First, due to differences in the output behavior of classifiers and object detectors, we make modifications to the adversarial loss function. Second, we observed additional constraints that an adversarial perturbation must be robust to and model these constraints synthetically. Finally, we introduce a smoothness constraint into the objective, rather than using the ℓ_p norm. In the following, we discuss each of these changes in detail.

4.2.1 Modified Adversarial Loss Function

An object detector outputs a set of bounding boxes and the likelihood of the most probable object contained within that box given a certain confidence threshold. See Figure 1 for a visualization of this output. By contrast, a classifier outputs a single vector where each entry represents the probability that the object in the image is of that type. Attacks on image classifiers typically make use of the cross-entropy loss between this output vector, and a one-hot representation of the adversarial target. However, this loss function is not applicable to object detectors due to their richer output structure. Thus, we introduce a new adversarial loss function suitable for use with detectors. This loss function is tailored to the specific attacks we introduce in this work.

Disappearance Attack Loss. The goal of the attacker is to prevent the object detector from detecting the target object. To achieve this, the adversarial perturbation must ensure that the likelihood of the target object in any bounding box is less than the detection threshold (the default is 25% for YOLO v2). In our implementation of the attack, we used the following loss function:

$$J_d(x,y) = \max_{s \in S^2, b \in B} P(s,b,y,f_\theta(x)) \qquad (2)$$

Where $f_\theta(x)$ represents the output of the object detector (for YOLO v2, this is a $19 \times 19 \times 425$ tensor). $P(\cdot)$ is a function that extracts the probability of an object class from this tensor, with label y (in our case, this is a Stop sign) in grid cell s and bounding box b. We denote x as the input scene containing our perturbed target object.

Therefore, the loss function outputs the maximum probability of a Stop sign if it occurs within the scene. Using this loss function, the goal of the adversary is to directly minimize that probability until it falls below the detection threshold of the network.

Creation Attack Loss. We propose a new type of *Creation Attack*, wherein the goal is to fool the model into recognizing nonexistent objects. Similar to the "adversarial patch" approach of [2], our goal is to create a physical sticker that can be added to any existing scene. Contrary to prior work, rather than causing a misclassification our aim is to create a new classification (*i.e.,* a new object detection) where non existed before.

For this, we use a composite loss function, that first aims at creating a new object localization, followed by a targeted "mis-classification." The mask M_x is sampled randomly so that the adversarial patch is applied to an arbitrary location in the scene. As above, let $f_\theta(x)$ represent the full output tensor of YOLO v2 on input scene x, and let $P(s,b,y,f_\theta(x))$ represent the probability assigned to class y in box b of grid cell s. Further let $P_{\text{box}}(s,b,f_\theta(x))$ represent the probability of the box only, *i.e.,* the model's confidence that the box contains *any* object. Our loss is then

$$\begin{aligned} \text{object} &= P_{\text{box}}(s,b,f_\theta(x)) > \tau \\ J_c(x,y) &= \text{object} + (1-\text{object}) \cdot P(s,b,y,f_\theta(x)) \end{aligned} \tag{3}$$

Here, τ is a threshold on the box confidence (set to 0.2 in our experiments), after which we stop optimizing the box confidence and focus on increasing the probability of the targeted class. As our YOLO v2 implementation uses a threshold of 0.1 on the product of the box confidence and class probability, any box with a confidence above 0.2 and a target class probability above 50% is retained.

4.2.2 Synthetic Representation of New Physical Constraints

Generating physical adversarial examples for detectors requires simulating a larger set of varying physical conditions than what is needed to trick classifiers. In our initial experiments, we observed that the generated perturbations would fail if the object was moved from its original position in the image. This is likely because a detector has access to more contextual information when generating predictions. As an object's position and size can vary greatly depending on the viewer's location, perturbations must account for these additional constraints.

To generate physical adversarial perturbations that are positionally invariant, we chose to synthetically model two environmental conditions: object rotation (in the Z plane) and position (in the X-Y plane). In each epoch of the optimization, we randomly place and rotate the object. Our approach differs from the original approach used by Eykholt *et al.*, in that they modeled an object's rotation physically using a diverse dataset. We avoided this approach because of the added complexity necessary for the alignment function, T_i, to properly position the adversarial perturbation on the sign. Since these transformations are done synthetically, the alignment function, T_i, simply needs to use the same process to transform the adversarial perturbation.

4.2.3 Noise Smoothing using Total Variation

The unmodified RP$_2$ algorithm uses the ℓ_p norm to smooth the perturbation. However, in our initial experiments, we observed that the ℓ_p norm results in very pixelated perturbations. The pixelation hurts the success rate of the attack, especially as the distance between the viewer and the object increases. We found that using the total variation norm in place of the ℓ_p norm gave smoother perturbations, thus increasing the effective range of the attack. Given a mask, M_x, and noise δ, the total variation norm of the adversarial perturbation, $M_x \cdot \delta$, is:

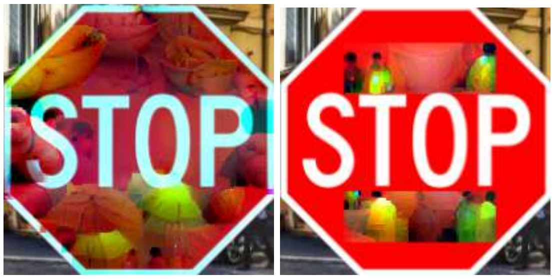


Figure 4: Output of the extended RP$_2$ algorithm to attack YOLO v2 using poster and sticker attacks.

$$\begin{aligned} &TV(M_x \cdot \delta) = \\ &\sum_{i,j} |(M_x \cdot \delta)_{i+1,j} - (M_x \cdot \delta)_{i,j}| \\ &\quad + |(M_x \cdot \delta)_{i,j+1} - (M_x \cdot \delta)_{i,j}| \end{aligned} \tag{4}$$

where i,j are the row and column indices for the adversarial perturbation. Thus our final modified objective function is:

$$\begin{aligned} &\underset{\delta}{\text{argmin}}\ \lambda TV(M_x \cdot \delta) + NPS \\ &\quad + \mathbb{E}_{x_i \sim X^V} J_d(x_i + T_i(M_x \cdot \delta), y^*) \end{aligned} \tag{5}$$

where $J_d(\cdot, y^*)$ is the loss function (discussed earlier) that measures the maximum probability of an object with the label y^* contained in the image. In our attack, y^* is a Stop sign.

5 Evaluation

We first discuss our experimental method, where we evaluate attacks in a whitebox manner using YOLO v2, and in a blackbox manner using Faster-RCNN. Then, we discuss our results, showing that state-of-the-art object detectors can be attacked using physical posters and stickers. Figure 4 shows the digital versions of posters and stickers used for disappearance attacks, while Figure 5 shows a digital version of the sticker used in a creation attack.

5.1 Experimental Setup

We evaluated our disappearance attack in a mix of lab and outdoor settings. For both the poster and sticker attacks, we generated adversarial perturbations and recorded several seconds of video. In each experiment, recording began 30 feet from the sign and ended when no part of the sign was in the camera's field of view. Then, we fed the video into the object detection

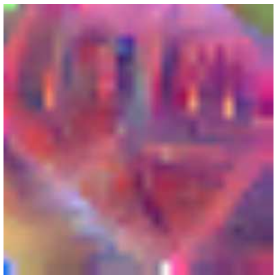


Figure 5: Patch created by the Creation Attack, aimed at fooling YOLO v2 into detecting nonexistent Stop signs.

YOLO v2	Poster	Sticker
Indoors	202/236 (85.6%)	210/247 (85.0%)
Outdoors	156/215 (72.5%)	146/230 (63.5%)

Table 1: Attack success rate for the disappearance attack on YOLO v2. We tested a poster perturbation, where a true-sized print is overlaid on a real Stop sign, and a sticker attack, where the perturbation is two rectangles stuck to the surface of the sign. The table cells show the ratio: number of frames in which a Stop sign was *not* detected / total number of frames, and a success rate, which is the result of this ratio.

network for analysis. We used the YOLO v2 object detector as a white-box attack. We also ran the same videos through the Faster-RCNN network to measure black-box transferability of our attack.

For the creation attack, we experimented with placing stickers on large flat objects (*e.g.,* a wall or cupboard), and recording videos within 10 feet of the sticker.

5.2 Experimental Results

We evaluated the perturbations for a disappearance attack using two different masks and attacked a Stop sign. First, we tested a poster perturbation, which used an octagonal mask to allow adversarial noise to to be added anywhere on the surface of the Stop sign. Next, we tested a sticker perturbation. We used the mask to create two rectangular stickers positioned at the top and bottom of the sign. The results of our attack are shown in Table 1.

In indoor lab settings, where the environment is relatively stable, both the poster and sticker perturbation demonstrate a high success rate in which at least 85% of the total video frames do not contain a Stop sign bounding box. When we evaluated our perturbations in an outdoor environment, we notice a drop in success rate for both attacks. The sticker perturbation also appears to be slightly weaker. We noticed that the sticker perturbation did especially poorly when only a portion of the sign was in the camera's field of view. Namely, when the sticker perturbation began to leave the camera's field of view, the Stop sign bounding boxes appear very frequently. In contrast, this behavior was not observed in the poster perturbation experiments, likely because some part of the adversarial noise is always present in the video due to the mask's shape. Figure 7 shows some frame captures of our adversarial Stop sign videos.

FR-CNN	Poster	Sticker
Indoors	189/220 (85.9%)	146/248 (58.9%)
Outdoors	84/209 (40.2%)	47/249 (18.9%)

Table 2: Attack success rate for the disappearance attack on Faster R-CNN. We tested a poster perturbation, where the entire Stop sign is replaced with a true-sized print, and a sticker attack, where the perturbation is two rectangles stuck to the surface of the sign. The table cells show the ratio: number of frames in which a Stop sign was *not* detected / total number of frames, and a success rate, which is the result of this ratio.

To measure the transferability of our attack, we also evaluated the recorded videos using the Faster R-CNN object detection network.[1]. The results for these experiments are shown in Table 2.

We see from these results that both perturbations transfer with a relatively high success rate in indoor lab settings where the environment conditions are stable. However, once outdoors, the success rate for both perturbations decreases significantly, but both perturbations retain moderate success rates. We observe that our improved attack algorithm can generate an adversarial poster perturbation, which transfers to other object detection frameworks, especially in stable environments.

Finally, we report on some preliminary results for creation attacks (the results are considered preliminary in that we have spent considerably less time optimizing these attacks compared to the disappearance attacks—it is thus likely that they can be further improved). When applying multiple copies of the sticker in Figure 5 to a cupboard and office wall, YOLO v2 detects stop signs in 25%–79% of the frames over multiple independent videos. A sample video frame is shown in Figure 6. Compared to the disappearance attack, the creation attack is more sensitive to the sticker's size, surroundings, and camera movement in the video. This results in highly variable success rates and is presumably because (due to resource constraints) we applied fewer physical and digital transformations when generating the attack. Enhanc-

[1] We used the Tensorflow-Python implementation of Faster R-CNN found at `https://github.com/endernewton/tf-faster-rcnn` It has a default detection threshold of 80%

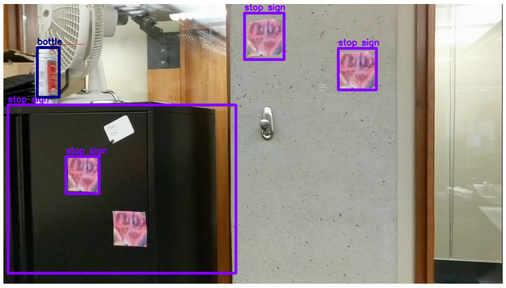


Figure 6: Sample frame from our creation attack video after being processed by YOLO v2. The scene includes 4 adversarial stickers reliably recognized as Stop signs.

ing the reliability and robustness of our creation attack is an interesting avenue for future work, as it presents a novel attack vector (*e.g.,* DOS style attacks) for adversarial examples.

6 Discussion

In the process of generating physical adversarial examples for object detectors, we note several open research questions that we leave to future work.

Lack of detail due to environmental conditions. We noticed physical conditions (*e.g.,* poor lighting, far distance, sharp angles), which only allowed macro features of the sign (*i.e.,* shape, general color, lettering) to be observed clearly. Due to such conditions, the details of the perturbations were lost, causing it to fail. This is expected as our attack relies on the camera being able to perceive the adversarial perturbations somewhat accurately. When extreme environmental conditions prevent the camera from observing finer details of the perturbation on the sign, the adversarial noise is lost. We theorize that in order to successfully fool object detectors under these extreme conditions, the macro features of the sign need to be attacked. For example, we could create attachments on the outside edges of the sign in order to change its perceived shape.

Alternative attacks on object detectors. In this work, we explored attacking the object detector such that it fails to locate an object, or that it detects non-existent objects. There are several alternative forms of attack we could consider. One alternative is to attempt to generate physical perturbations that preserve the bounding box of an object, but alter its label (this is similar to targeted attacks for classifiers). Another option is to generate further 2D or even 3D objects that appear nonsensical to a human, but are detected and labeled by the object detector. The success of either of these attacks, which have been shown to work digitally [13, 22], would have major safety implications.

Extensions to semantic segmentation. A broader task than object detection is semantic segmentation—where the network labels every pixel in a scene as belonging to an object. Recent work has shown digital attacks against semantic segmentation [22]. An important future work question is how to extend current attack techniques for classifiers, and detectors (as this work shows) to create physical attacks on segmentation networks.

Impact on Real Systems. Existing cyber-physical systems such as cars and drones integrate object detectors into a control pipeline that consists of pre- and post-processing steps. The attacks we show only target the object detection component in isolation (specifically YOLO v2). Understanding whether these attacks are capable of compromising a full control pipeline in an end-to-end manner is an important open question. Although YOLO v2 does recognize a Stop sign in some frames from our attack videos, a real system would generally base its control decisions on a majority of predictions, rather than a few frames. Our attack manages to trick the detector into not seeing a Stop sign in a majority of the tested video frames.

Despite these observations, we stress that a key step towards understanding the vulnerability of the broad class of object detection models to physical adversarial examples is to create algorithms that can attack state-of-the-art object detectors. In this work, we have shown how to can extend the existing RP_2 algorithm with positional and rotational invariance to attack object detectors in relatively controlled settings.

7 Conclusion

Starting from an algorithm to generate robust physical perturbations for *classifiers*, we extend it with positional and rotational invariance to generate physical perturbations for state-of-the-art object *detectors*—a broader class of deep neural networks that are used in dynamic settings to detect and label objects within scenes. Object detectors are popular in cyber-physical systems such as autonomous vehicles. We experiment with the YOLO v2 object detector, showing that it is possible to physically perturb a Stop sign such that the detector ignores it. When presented with a video of the adversarial poster perturbation, YOLO failed to recognize the sign in 85.6% of the video frames in a controlled lab environment, and in 72.5% of the video frames in an outdoor environment. When presented with a video of the adversarial sticker perturbation, YOLO failed to recognize the sign in 85% of the video frames in a controlled lab environment, and in 63.5% of the video frames in an outdoor environment. We also observed limited blackbox transferability to the Faster-RCNN detector. The poster perturbation fooled Faster R-CNN in 85.9% of the video frames in a controlled lab environment, and in 40.2% of the video frames in an outdoor environment. Our work, thus, takes steps towards developing a more informed understanding of the vulnerability of object detectors to physical adversarial examples.

Acknowledgements

We thank the reviewers for their insightful feedback. This work was supported in part by NSF grants 1422211, 1565252, 1616575, 1646392, 1740897, Berkeley Deep Drive, the Center for Long-Term Cybersecurity, FORCES (which receives support from the NSF), the Hewlett Foundation, the MacArthur Foundation, a UM-SJTU grant, and the UW Tech Policy Lab.

References

[1] Athalye, A., and Sutskever, I. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397* (2017).

[2] Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. Adversarial patch. *arXiv preprint arXiv:1712.09665* (2017).

[3] Carlini, N., and Wagner, D. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on* (2017), IEEE, pp. 39–57.

[4] Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., Rahmati, A., and Song, D. Robust physical-world attacks on machine learning models. *CVPR '18* (2018).

[5] Girshick, R. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2015).

[6] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[7] Kos, J., Fischer, I., and Song, D. Adversarial examples for generative models. *arXiv preprint arXiv:1702.06832* (2017).

[8] Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).

[9] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. Ssd: Single shot multibox detector. In *European conference on computer vision* (2016), Springer, pp. 21–37.

[11] Lu, J., Sibai, H., Fabry, E., and Forsyth, D. A. NO need to worry about adversarial examples in object detection in autonomous vehicles. *CoRR abs/1707.03501* (2017).

[12] Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. *arXiv preprint arXiv:1511.04599* (2015).

[13] Nguyen, A., Yosinski, J., and Cluneand, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *In Computer Vision and Pattern Recognition (CVPR)* (2015).

[14] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on* (2016), IEEE, pp. 372–387.

[15] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 779–788.

[16] Redmon, J., and Farhadi, A. YOLO9000: better, faster, stronger. *CoRR abs/1612.08242* (2016).

[17] Sabour, S., Cao, Y., Faghri, F., and Fleet, D. J. Adversarial manipulation of deep representations. *arXiv preprint arXiv:1511.05122* (2015).

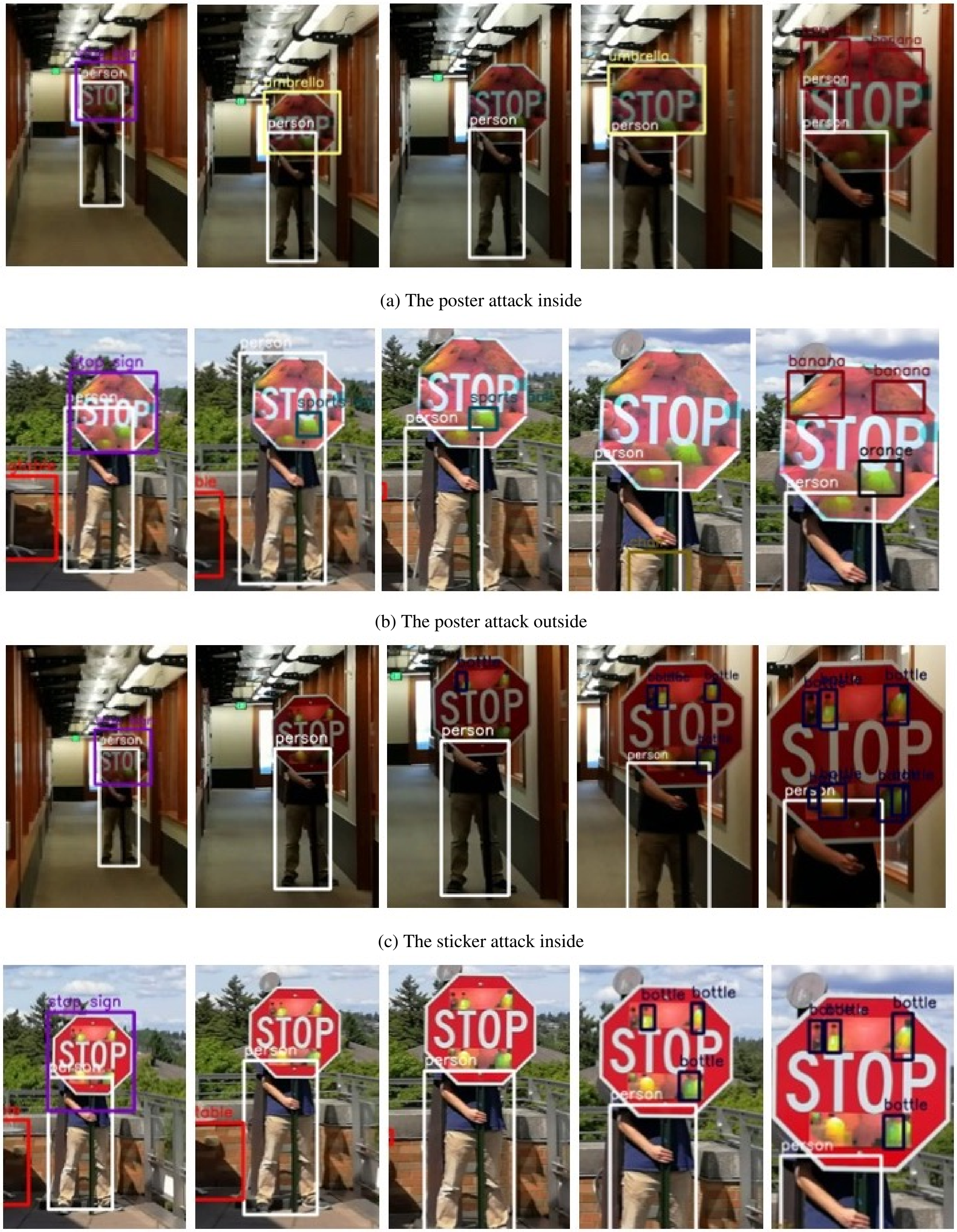
[18] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and Lecun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR) (Banff)* (2013).

[19] Shaoqing Ren, Kaiming He, R. G. J. S. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497* (2015).

[20] Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), ACM, pp. 1528–1540.

[21] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations* (2014).

[22] Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., and Yuille, A. L. Adversarial examples for semantic segmentation and object detection. *CoRR abs/1703.08603* (2017).



(a) The poster attack inside

(b) The poster attack outside

(c) The sticker attack inside

(d) The sticker attack outside

Figure 7: Sample frames from our attack videos after being processed by YOLO v2. In the majority of frames, the detector fails to recognize the Stop sign.