ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector

Shang-Tse Chen[1(✉)], Cory Cornelius[2], Jason Martin[2], and Duen Horng (Polo) Chau[1]

[1] Georgia Institute of Technology, Atlanta, GA, USA
{schen351,polo}@gatech.edu
[2] Intel Corporation, Hillsboro, OR, USA
{cory.cornelius,jason.martin}@intel.com

Abstract. Given the ability to directly manipulate image pixels in the digital input space, an adversary can easily generate imperceptible perturbations to fool a Deep Neural Network (DNN) image classifier, as demonstrated in prior work. In this work, we propose *ShapeShifter*, an attack that tackles the more challenging problem of crafting physical adversarial perturbations to fool image-based object detectors like Faster R-CNN. Attacking an object detector is more difficult than attacking an image classifier, as it needs to mislead the classification results in multiple bounding boxes with different scales. Extending the digital attack to the physical world adds another layer of difficulty, because it requires the perturbation to be robust enough to survive real-world distortions due to different viewing distances and angles, lighting conditions, and camera limitations. We show that the *Expectation over Transformation* technique, which was originally proposed to enhance the robustness of adversarial perturbations in image classification, can be successfully adapted to the object detection setting. *ShapeShifter* can generate adversarially perturbed stop signs that are consistently mis-detected by Faster R-CNN as other objects, posing a potential threat to autonomous vehicles and other safety-critical computer vision systems. Code related to this paper is available at: https://github.com/shangtse/robust-physical-attack.

Keywords: Adversarial attack · Object detection · Faster R-CNN

1 Introduction

Adversarial examples are input instances that are intentionally designed to fool a machine learning model into producing a chosen prediction. The success of Deep Neural Network (DNN) in computer vision does not exempt it from this threat. It is possible to bring the accuracy of a state-of-the-art DNN image classifier down to near zero percent by adding imperceptible adversarial perturbations [5,22]. The existence of adversarial examples not only reveals intriguing

M. Berlingerio et al. (Eds.): ECML PKDD 2018, LNAI 11051, pp. 52–68, 2019.
https://doi.org/10.1007/978-3-030-10925-7_4

theoretical properties of DNN, but also raises serious practical concerns on its deployment in security and safety critical systems. Autonomous vehicle is an example application that cannot be fully trusted before guaranteeing the robustness to adversarial attacks. The imperative need to understand the vulnerabilities of DNNs attracts tremendous interest among machine learning, computer vision, and security researchers.

Although many adversarial attack algorithms have been proposed, attacking a real-world computer vision system is difficult. First of all, most of the existing attack algorithms only focus on the image classification task, yet in many real-world use cases there will be more than one object in an image. Object detection, which recognizes and localizes multiple objects in an image, is a more suitable model for many vision-based scenarios. Attacking an object detector is more difficult than attacking an image classifier, as it needs to mislead the classification results in multiple bounding boxes with different scales [14].

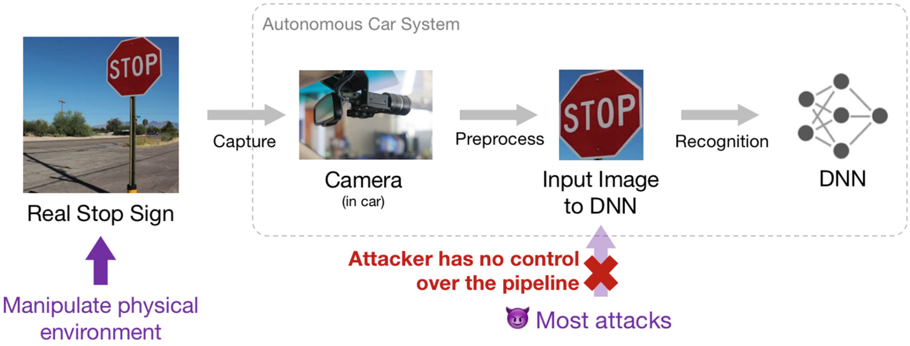


Fig. 1. Illustration motivating the need of physical adversarial attack, from attackers' perspectives, as they typically do not have full control over the computer vision system pipeline.

Further difficulty comes from the fact that DNN is usually only a component in the whole computer vision system pipeline. For many applications, attackers usually do not have the ability to directly manipulate data inside the pipeline. Instead, they can only manipulate the things outside of the system, i.e., those things in the physical environment. Figure 1 illustrates the intuition behind *physical adversarial attacks*. To be successful attacks, physical adversarial attacks must be robust enough to survive real-world distortions due to different viewing distances and angles, lighting conditions, and camera limitations.

There has been prior work that can either attack object detectors digitally [23], or attack image classifiers physically [6,10,19]. However, so far the existing attempts to physically attack object detectors remain unsatisfactory. A perturbed stop sign is shown in [13] that cannot be detected by the Faster R-CNN object detector [18]. However, the perturbation is very large and they tested it with poor texture contrast against the background, making the perturbed stop sign hard to see even by human. A recent short note [7] claims to be able to generate some adversarial stickers that, when attaching to a stop sign, can fool the YOLO object detector [17] and can be transferable to also fool Faster R-CNN. However, they did not reveal the algorithm used to create the sticker and only show a video of indoor experiment with short distance. For other threat models and adversarial attacks in computer vision, we refer the interested readers to the survey of [1].

In this work, we propose *ShapeShifter*, the first robust targeted attack that can fool a state-of-the-art Faster R-CNN object detector. To make the attack robust, we adopt the *Expectation over Transformation* technique [3,4], and adapt it from the image classification task to the object detection setting. As a case study, we generate some adversarially perturbed stop signs that can consistently be mis-detected by Faster R-CNN as the target objects in real drive-by tests. Our contributions are summarized below.

1.1 Our Contributions

- To the best of knowledge, our work presents the first reproducible and robust targeted attack against Faster R-CNN [14]. Recent attempts either can only do untargeted attack and requires perturbations with "extreme patterns" (in the researchers' words) to work consistently [13], or has not revealed the details of the method [7]. We have open-sourced our code on GitHub[1].
- We show that the *Expectation over Transformation* technique, originally proposed for image classification, can be applied in the object detection task and significantly enhance robustness of the resulting perturbation.
- By carefully studying the Faster R-CNN object detector algorithm, we overcome non-differentiability in the model, and successfully perform optimization-based attacks using gradient descent and backpropogation.
- We generate perturbed stop signs that can consistently fool Faster R-CNN in real drive-by tests (videos available on the GitHub repository), calling for imperative need to improve and fortify vision-based object detectors.

2 Background

This section provides background information of adversarial attacks and briefly describes the Faster R-CNN object detector that we try to attack in this work.

2.1 Adversarial Attack

Given a trained machine learning model C and a benign instance $x \in \mathcal{X}$ that is correctly classified by C, the goal of the untargeted adversarial attack is to find another instance $x' \in \mathcal{X}$, such that $C(x') \neq C(x)$ and $d(x, x') \leq \epsilon$ for some

[1] https://github.com/shangtse/robust-physical-attack.

distance metric $d(\cdot,\cdot)$ and perturbation budget $\epsilon > 0$. For targeted attack, we further require $C(x') = y'$ where $y' \neq C(x)$ is the target class. Common distance metrics $d(\cdot,\cdot)$ in the computer vision domain are ℓ_2 distance $d(x,x') = ||x - x'||_2^2$ and ℓ_∞ distance $d(x,x') = ||x - x'||_\infty$.

The work of [22] was the first to discover the existence of adversarial examples for DNNs. Several subsequent works have improved the computational cost and made the perturbation highly imperceptible to human [8,15]. Most adversarial attack algorithms against DNNs assume that the model is differentiable, and use the gradient information of the model to tweak the input instance to achieve the desired model output [5]. Sharif et al. [19] first demonstrated a physically realizable attack to fool a face recognition model by wearing an adversarially crafted pair of glasses.

2.2 Faster R-CNN

Faster R-CNN [18] is one of the state-of-the-art general object detectors. It adopts a 2-stage detection strategy. In the first state, a region proposal network is used to generate several class-agnostic bounding boxes called region proposals that may contain objects. In the second stage, a classifier and a regressor are used to output the classification results and refined bounding box coordinates for each region proposal, respectively. The computation cost is significantly reduced by sharing the convolutional layers in the two stages. Faster R-CNN is much harder to attack, as a single object can be covered by multiple region proposals of different sizes and aspect ratios, and one needs to mislead the classification results in all the region proposals to fool the detection.

3 Threat Model

Existing methods that generate adversarial examples typically yield imperceptible perturbations that fool a given machine learning model. Our work, following [19], generates perturbations that are perceptible but constrained such that a human would not be easily fooled by such a perturbation. We examine this kind of perturbation in the context of object detection (e.g., stop sign). We chose this use case because of object detector's possible uses in security-related and safety-related settings (e.g., autonomous vehicles). For example, attacks on traffic sign recognition could cause a car to miss a stop sign or travel faster than legally allowed.

We assume the adversary has white-box level access to the machine learning model. This means the adversary has access to the model structure and weights to the degree that the adversary can both compute outputs (i.e., the forward pass) and gradients (i.e., the backward pass). It also means that the adversary does not have to construct a perturbation in real-time. Rather, the adversary can study the model and craft an attack for that model using methods like Carlini-Wagner attack [5]. This kind of adversary is distinguished from a black-box adversary who is defined as having no such access to the model architecture or weights. While our choice of adversary is the most powerful one, existing research has shown it is possible to construct imperceptible perturbations without white-box level access [16]. However, whether our method is capable of generating perceptible perturbations with only black-box access remains an open question. Results from Liu et al. [12] suggest that iterative attacks (like ours) tend not to transfer as well as non-iterative attacks.

Unlike previous work, we restrict the adversary such that they cannot manipulate the digital values of pixels gathered from the camera that each use case uses to sense the world. This is an important distinction from existing imperceptible perturbation methods. Because those methods create imperceptible perturbations, there is a high likelihood such perturbations would not fool our use cases when physically realized. That is, when printed and then presented to the systems in our use cases, those perturbations would have to survive both the printing process and sensing pipeline in order to fool the system. This is not an insurmountable task as Kurakin et al. [10] have constructed such imperceptible yet physically realizable adversarial perturbations for image classification systems.

Finally, we also restrict our adversary by limiting the shape of the perturbation the adversary can generate. This is important distinction for our use cases because one could easily craft an odd-shaped "stop sign" that does not exist in the real world. We also do not give the adversary the latitude of modifying all pixels in an image like Kurakin et al. [10], but rather restrict them to certain pixels that we believe are both inconspicuous and physically realistic.

4 Attack Method

Our attack method, *ShapeShifter*, is inspired by the iterative, change-of-variable attack described in [5] and the *Expectation over Transformation* technique [3,4]. Both methods were originally proposed for the task of image classification. We describe these two methods in the image classification setting before showing how to extend them to attack the Faster R-CNN object detector.

4.1 Attacking an Image Classifier

Let $F : [-1, 1]^{h \times w \times 3} \to \mathbb{R}^K$ be an image classifier that takes an image of height h and width w as input, and outputs a probability distribution over K classes. The goal of the attacker is to create an image x' that looks like an object x of class y, but will be classified as another target class y'.

Change-of-variable Attack. Denote $L_F(x, y) = L(F(x), y)$ as the loss function that calculates the distance between the model output $F(x)$ and the target label y. Given an original input image x and a target class y', the change-of-variable attack [5] propose the following optimization formulation.

$$\underset{x' \in \mathbb{R}^{h \times w \times 3}}{\arg\min}\ L_F(\tanh(x'), y') + c \cdot ||\tanh(x') - x||_2^2. \tag{1}$$

The use of *tanh* ensures that each pixel is between $[-1, 1]$. The constant c controls the similarity between the modified object x' and the original image x. In practice, c can be determined by binary search [5].

Expectation over Transformation. The *Expectation over Transformation* [3,4] idea is simple: adding random distortions in each iteration of the optimization to make the resulting perturbation more robust. Given a transformation t that can be translation, rotation, and scaling, $M_t(x_b, x_o)$ is an operation that transforms an object image x_o using t and then overlays it onto a background image x_b. $M_t(x_b, x_o)$ can also include a masking operation that only keeps a certain area of x_o. This will be helpful when one wants to restrict the shape of the perturbation. After incorporating the random distortions, Eq. (1) becomes

$$\underset{x' \in \mathbb{R}^{h\times w\times 3}}{\arg\min}\ \mathbb{E}_{x\sim X, t\sim T}\left[L_F(M_t(x, \tanh(x')), y')\right] + c \cdot ||\tanh(x') - x_o||_2^2, \tag{2}$$

where X is the training set of background images. When the model F is differentiable, this optimization problem can be solved by gradient descent and back-propagation. The expectation can be approximated by the empirical mean.

4.2 Extension to Attacking Faster R-CNN

An object detector $F : [-1, 1]^{h\times w\times 3} \to (\mathbb{R}^{N\times K}, \mathbb{R}^{N\times 4})$ takes an image as input and outputs N detected objects. Each detection includes a probability distribution over K pre-defined classification classes as well as the location of the detected object, represented by its 4 coordinates. Note that it is possible for an object detector to output more or fewer detected objects, depending on the input image, but for simplicity we select top-N detected objects ranked by confidence.

As described in Subsect. 2.2, Faster R-CNN adopts a 2-stage approach. The region proposal network in the first stage outputs several region proposals, and the second stage classifier performs classification within each of the region proposals. Let $rpn(x) = \{r_1, \ldots, r_m\}$, where each r_i is a region proposal represented as its four coordinates, and let x_r be a sub-image covered by region r. Denote $L_{F_i}(x, y) = L(F(x_{r_i}), y)$, i.e., the loss of the classification in the i-th region proposal. We can simultaneously attack all the classifications in each region proposal by doing the following optimization.

$$\underset{x' \in \mathbb{R}^{h\times w\times 3}}{\arg\min}\ \mathbb{E}_{x\sim X, t\sim T}\left[\frac{1}{m}\sum_{r_i \in rpn(M_t(x'))} L_{F_i}(M_t(x'), y')\right] + c \cdot ||\tanh(x') - x_o||_2^2, \tag{3}$$

where we abuse the notation $M_t(x') = M_t(x, \tanh(x'))$ for simplicity. However, for computational issues, most models prune the region proposals by using heuristics like non-maximum suppression [18]. The pruning operations are usually non-differentiable, making it hard to optimize equation (3) end to end.

Therefore, we approximately solve this optimization problem by first run a forward pass of the region proposal network, and fixed the pruned region proposals as fixed constants to the second stage classification problem in each iteration. We empirically find this approximation sufficient to find a good solution.

5 Evaluation

We evaluate our method by fooling a pre-trained Faster R-CNN model with Inception-v2 [21] convolutional feature extraction component. The model was trained on the Microsoft Common Objects in Context (MS-COCO) dataset [11] and is publicly available in the Tensorflow Object Detection API [9] model zoo repository[2].

The MS-COCO dataset contains 80 general object classes ranging from people and animals to trucks and cars and other common objects. Although our method can potentially be used to attack any classes, we choose to focus on attacking the stop sign class due to its importance and relevance to self-driving cars, where a vision-based object detector may be used to help make decisions. An additional benefit of choosing the stop sign is its flat shape that can easily be printed on a paper. Other classes, like dogs, are less likely to be perceived as real objects by human when printed on a paper. While 3D printing adversarial examples for image recognition is possible [3], we leave 3D-printed adversarial examples against object detectors as future work.

5.1 Digitally Perturbed Stop Sign

We generate adversarial stop signs by performing the optimization process described in Eq. 3. The hyperparameter c is crucial in determining the perturbation strength. A smaller value of c will result in a more conspicuous perturbation, but the perturbation will also be more robust to real-world distortions when we do the physical attack later.

However, it is hard to choose an appropriate c when naively using the ℓ_2 distance to a real stop sign as regularization. To obtain a robust enough perturbation, a very small c needs to be used, which has the consequence of creating stop signs that are difficult for humans to recognize. The ℓ_2 distance is not a perfect metric for human perception, which tends to be more sensitive to color changes on lighter-colored objects. Due to this observation, we only allow the perturbation to change the red part of the stop sign, leaving the white text intact. This allows us to generate larger and more robust perturbation, while providing enough contrast between the lettering and red parts so that a human can easily recognize the perturbation as a stop sign. The adversarial stop sign generated in [13] does not consider this and is visually more conspicuous. Automating this procedure for other objects we leave as future work.

[2] http://download.tensorflow.org/models/object_detection/faster_rcnn_inception_v2_coco_2017_11_08.tar.gz.

We performed two targeted attacks and one untargeted attack. We choose person and sports ball as the two target classes because they are relatively similar in size and shape to stop signs. Our method allows attackers to use any target classes, however the perturbation needs to achieve its means and fool the object detector. For some target classes, this may mean creating perturbations so large in deviation that they may appear radically different from the victim class. We also noticed that some classes are easier to be detected at small scales, such as *kite*, while other classes (e.g., *truck*) could not be detected when the object was too small. This may be an artifact of the MS-COCO dataset that the object detector was trained on. Nevertheless, ultimately the attacker has a choice in target class and, given ample time, can find the target class that best fools the object detector according to their means.

For each attack, we generated a high confidence perturbation and a low level perturbation. The high confidence perturbations were generated using a smaller value of c, thus making them more conspicuous but also more robust. Depending upon the target class, it may be difficult to generate an effective perturbation. We manually chose c for each target class so that the digital attack achieves high success rate while keeping the perturbation not too conspicuous, i.e., we tried to keep the color as red as possible. We used $c = 0.002$ for the high confidence perturbations and $c = 0.005$ for the low confidence perturbations in the “sports ball” targeted attack and the untargeted attack. We used $c = 0.005$ and $c = 0.01$ for the high and low confidence perturbations in the “person” targeted attack, respectively. The 6 perturbations we created are shown in Fig. 2.

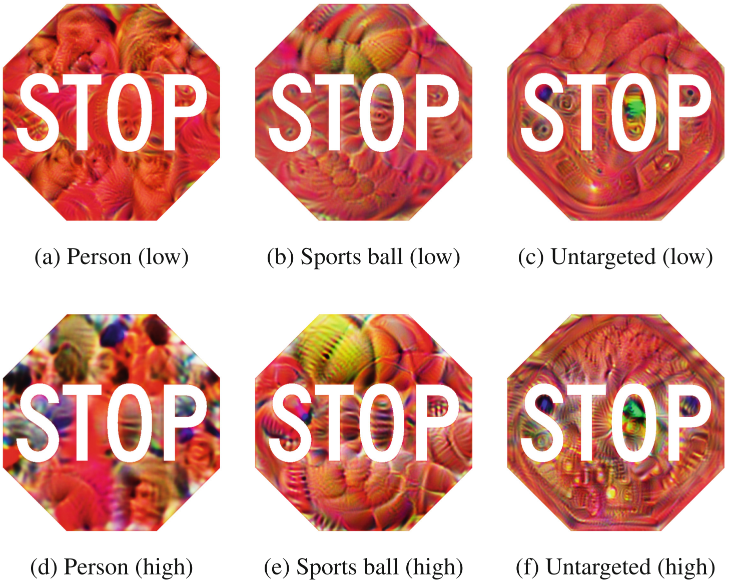
5.2 Physical Attack

We performed physical attacks on the object detector by printing out the perturbed stop signs shown in Fig. 2. We then took photos from a variety of distances and angles in a controlled indoor setting. We also conducted drive-by tests by recording videos from a moving vehicle that approached the signs from a distance. The lightning conditions varied from recording to recording depending upon the weather at the time.

Equipment. We used a Canon Pixma Pro-100 photo printer to print out signs with high-confidence perturbations, and an HP DesignJet to print out those with low-confidence perturbations[3]. For static images, we used a Canon EOS Rebel T7i DSLR camera, equipped with a EF-S 18-55mm IS STM lens. The videos in our drive-by tests are shot using an iPhone 8 Plus mounted on the windshield of a car.

Indoor Experiments. Following the experimental setup of [6], we took photos of the printed adversarial stop sign, at a variety of distances ($5'$ to $40'$) and

[3] We used two printers to speed up our sign production, since a sign can take more than 30 min to produce.



(a) Person (low) (b) Sports ball (low) (c) Untargeted (low)

(d) Person (high) (e) Sports ball (high) (f) Untargeted (high)

Fig. 2. Digital perturbations we created using our method. Low confidence perturbations on the top and high confidence perturbations on the bottom.

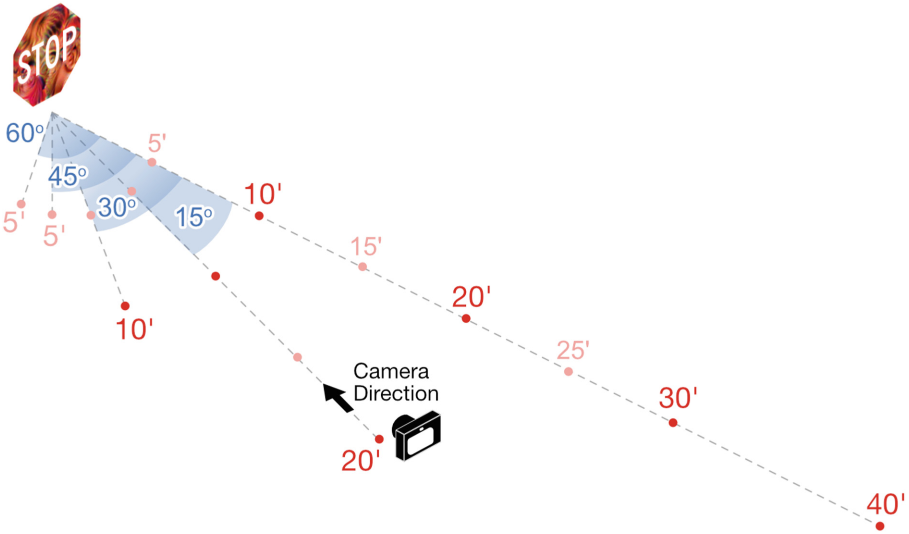


Fig. 3. Indoor experiment setup. We take photos of the printed adversarial sign, from multiple angles (0°, 15°, 30°, 45°, 60°, from the sign's tangent), and distances (5′ to 40′). The camera locations are indicated by the red dots, and the camera always points at the sign. (Color figure online)

Table 1. Our **high-confidence** perturbations succeed at attacking at a variety of distances and angles. For each distance-angle combination, we show the detected class and the confidence score. If more than one bounding boxes are detected, we report the highest-scoring one. Confidence values lower than 30% is considered undetected.

Distance	Angle	Person	(Conf.)	Sports ball	(Conf.)	Untargeted	(Conf.)
5′	0°	person	(.77)	sports ball	(.61)	clock	(.35)
5′	15°	person	(.91)	cake	(.73)	clock	(.41)
5′	30°	person	(.93)	cake	(.66)	cake	(.39)
5′	45°	person	(.69)	cake	(.61)	*stop sign*	(.62)
5′	60°	*stop sign*	(.93)	*stop sign*	(.70)	*stop sign*	(.88)
10′	0°	person	(.55)	cake	(.34)	clock	(.99)
10′	15°	person	(.63)	cake	(.33)	clock	(.99)
10′	30°	person	(.51)	cake	(.55)	clock	(.99)
15′	0°	undetected	—	cake	(.49)	clock	(.99)
15′	15°	person	(.57)	cake	(.53)	clock	(.99)
20′	0°	person	(.49)	sports ball	(.98)	clock	(.99)
20′	15°	person	(.41)	sports ball	(.96)	clock	(.99)
25′	0°	person	(.47)	sports ball	(.99)	*stop sign*	(.91)
30′	0°	person	(.49)	sports ball	(.92)	undetected	—
40′	0°	person	(.56)	sports ball	(.30)	*stop sign*	(.30)
Targeted success rate		87%		40%		N/A	
Untargeted success rate		93%		93%		73%	

Table 2. As expected, low-confidence perturbations achieve lower success rates.

Distance	Angle	Person	(Conf.)	Sports ball	(Conf.)	Untargeted	(Conf.)
5′	0°	*stop sign*	(.87)	cake	(.90)	cake	(.41)
5′	15°	*stop sign*	(.63)	cake	(.93)	cake	(.34)
5′	30°	person	(.83)	cake	(.84)	*stop sign*	(.48)
5′	45°	*stop sign*	(.97)	*stop sign*	(.94)	*stop sign*	(.82)
5′	60°	*stop sign*	(.99)	*stop sign*	(.99)	*stop sign*	(.89)
10′	0°	*stop sign*	(.83)	*stop sign*	(.99)	undetected	—
10′	15°	*stop sign*	(.79)	*stop sign*	(.94)	undetected	—
10′	30°	*stop sign*	(.60)	*stop sign*	(.98)	*stop sign*	(.78)
15′	0°	*stop sign*	(.52)	*stop sign*	(.94)	*stop sign*	(.31)
15′	15°	*stop sign*	(.33)	*stop sign*	(.93)	undetected	—
20′	0°	*stop sign*	(.42)	sports ball	(.73)	undetected	—
20′	15°	person	(.51)	sports ball	(.83)	cell phone	(.62)
25′	0°	*stop sign*	(.94)	sports ball	(.87)	undetected	—
30′	0°	*stop sign*	(.94)	sports ball	(.95)	*stop sign*	(.79)
40′	0°	*stop sign*	(.95)	undetected	—	*stop sign*	(.52)
Targeted success rate		13%		27%		N/A	
Untargeted success rate		13%		53%		53%	

angles (0°, 15°, 30°, 45°, 60°, from the sign's tangent). This setup is depicted in Fig. 3 where camera locations are indicated by red dots. The camera always pointed at the sign. We intended these distance-angle combinations to mimic a vehicle's points of view as it would approach the sign from a distance [13]. Tables 1 and 2 summarize the results for our *high-confidence* and *low-confidence* perturbations, respectively. For each distance-angle combination, we show the detected class and the detection's confidence score. If more than one bounding boxes are detected, we report the highest-scoring one. Confidence values lower than 30% were considered undetected; we decided to use the threshold of 30%, instead of the default 50% in the Tensorflow Object Detection API [9], to impose a stricter requirement on ourselves (the "attacker"). Since an object can be detected as a stop sign and the target class simultaneously, we consider our attack to be successful only when the confidence score of the target class is the highest among all of the detected classes.

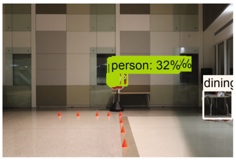
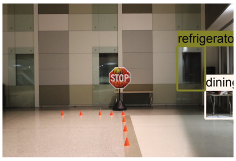
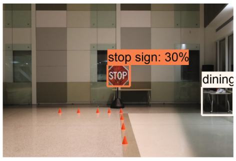
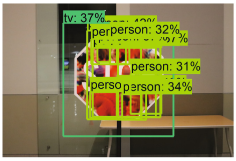
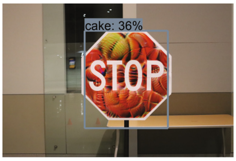
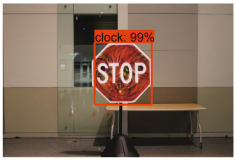
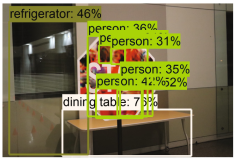
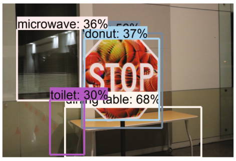
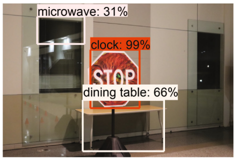
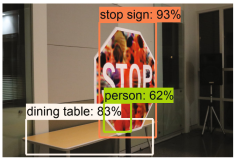
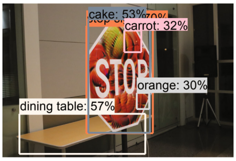
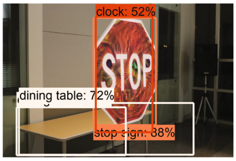
Table 1 shows that our high-confidence perturbations achieve a high attack success rate at a variety of distances and angles. For example, we achieved a targeted success rate 87% in misleading the object detector into detecting the stop sign as a *person*, and an even higher untargeted success rate of 93% when our attack goal is to cause the detector to either fail to detect the stop sign (e.g., at 15′ 0°) or to detect it as a class that is *not* a stop sign. The *sports ball* targeted attack has a lower targeted success rate but achieves the same untargeted success rate. Our untargeted attack consistently misleads the detection into the *clock* class in medium distances, but is less robust for longer distances. Overall, the perturbation is less robust to very high viewing angle (60° from the sign's tangent), because we did not simulate the viewing angle distortion in the optimization.

The low-confidence perturbations (Table 2), as expected, achieve a much lower attack success rate, suggesting the need to use higher-confidence perturbations when we conduct the more challenging drive-by tests (as we shall describe in the next section). Table 3 shows some sample high-confidence perturbations from our indoor experiments.

Drive-By Tests. We performed drive-by tests at a parking lot so as not to disrupt other vehicles with our stop signs. We put a purchased real stop sign as a control and our printed perturbed stop sign side by side. Starting from about 200 feet away, we slowly drove (between 5 mph to 15 mph) towards the signs while simultaneously recording video from the vehicle's dashboard at 4 K resolution and 24 FPS using an iPhone 8 Plus. We extracted all video frames, and for each frame, we obtained the detection results from Faster R-CNN object detection model. Because our low confidence attacks showed relatively little robustness indoors, we only include the results from our high-confidence attack. Similar to our indoor experiments, we only consider detections that had a confidence score of at least 30%.

In Fig. 4, we show sample video frames (rectangular images) to give the readers a sense of the size of the signs relative to the full video frame; we also

Table 3. Sample high-confidence perturbations from indoor experiments. For complete experiment results, please refer to Table 1.

Dist.	Angle	Target: person	Target: sports ball	Untargeted
40'	0°			
10'	0°			
10'	30°			
5'	60°		 cake: 53% carrot: 32% STOP orange: 30% dining table: 57%	

show zoomed-in views (square images) that more clearly show the Faster R-CNN detection results.

The *person-perturbation* in Fig. 4a drive-by totaled 405 frames as partially shown in the figure. The real stop sign in the video was correctly detected in every frame with high confidence. On the other hand, the perturbed stop sign was only correctly detected once, while 190 of the frames identified the perturbed stop sign as a person with medium confidence. For the rest of the 214 frames the object detector failed to detect anything around the perturbed stop sign.

The video we took with the *sports-ball-perturbation* shown in Fig. 4b had 445 frames. The real stop sign was correctly identified all of the time, while the perturbed stop sign was never detected as a stop sign. As the vehicle (video camera) moved closer to the perturbed stop sign, 160 of the frames were detected as a sports ball with medium confidence. One frame was detected as *apple* and *sports ball* and the remaining 284 frames had no detection around the perturbed stop sign.

Finally, the video of the untargeted perturbation (Fig. 4c) totaled 367 frames. While the unperturbed stop sign was correctly detected all of the time, the perturbed stop sign was detected as *bird* 6 times and never detected for the remaining 361 frames.

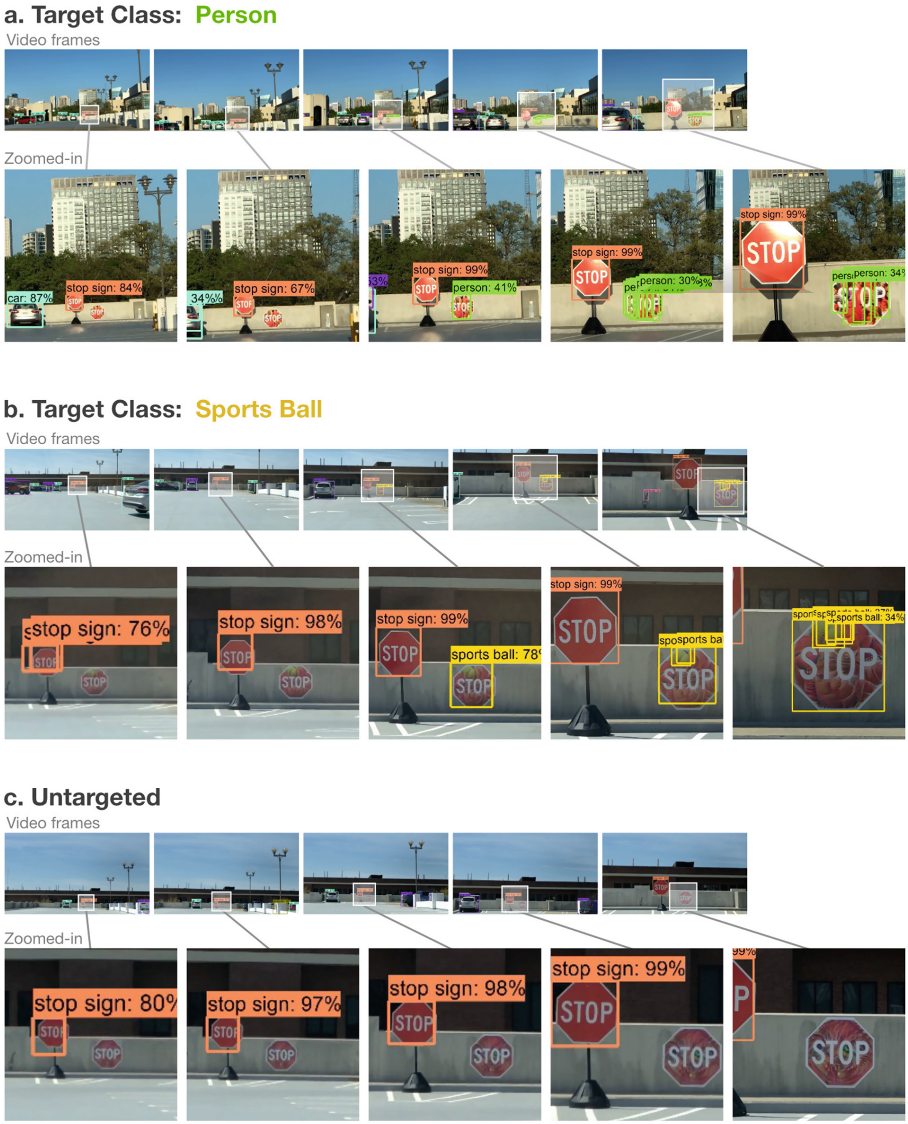


Fig. 4. Snapshots of the drive-by test results. In (a), the person perturbation was detected 47% of the frames as a person and only once as a stop sign. The perturbation in (b) was detected 36% of the time as a sports ball and never as a stop sign. The untargeted perturbation in (c) was detected as *bird* 6 times and never detected as a stop sign or anything else for the remaining frames.

Exploring Black-Box Transferability. We also sought to understand how well our high-confidence perturbations could fool other object detection models. For image recognition, it is known that high-confidence targeted attacks fail to transfer [12].

To this end, we fed our high-confidence perturbations into 8 other MS-COCO-trained models from the Tensorflow detection model zoo[4]. Table 4 shows how well our perturbation generated from the Faster R-CNN Inception-V2 transfer to other models. To better understand transferability, we examined the worse case. That is, if a model successfully detects a stop sign in the image, we say the perturbation has failed to transfer or attack that model. We report the number of images (of the 15 angle-distance images in our indoor experiments) where a model successfully detected a stop sign with at least 30% confidence. We also report the maximum confidence of all of those detected stop sign.

Table 4. Black-box transferability of our 3 perturbations. We report the number of images (of the 15 angle-distance images) that failed to transfer to the specified model. We consider the detection of any stop sign a “failure to transfer.” Our perturbations fail to transfer for most models, most likely due to the iterative nature of our attack.

Model	Person	(Conf.)	Sports ball	(Conf.)	Untargeted	(Conf.)
Faster R-CNN Inception-V2	3	(.93)	1	(.70)	5	(0.91)
SSD MobileNet-V2	2	(.69)	8	(.96)	15	(1.00)
SSD Inception-V2	11	(1.00)	14	(.99)	15	(1.00)
R-FCN ResNet-101	4	(.82)	10	(.85)	15	(1.00)
Faster R-CNN ResNet-50	13	(.00)	15	(1.00)	15	(1.00)
Faster R-CNN ResNet-101	15	(.99)	13	(.97)	15	(1.00)
Faster R-CNN Inc-Res-V2	1	(.70)	0	(.00)	12	(1.00)
Faster R-CNN NASNet	14	(1.00)	15	(1.00)	15	(1.00)

Table 4 shows the lack of transferability of our generated perturbations. The untargeted perturbation fails to transfer most of the time, followed by the sports ball perturbation, and finally the person perturbation. The models most susceptible to transferability were the Faster R-CNN Inception-ResNet-V2 model, followed by the SSD MobileNet-V2 model. Iterative attacks on image recognition also usually fail to transfer [12], so it is not surprising that our attacks fail to transfer as well. We leave the thorough exploration of transferability as future work.

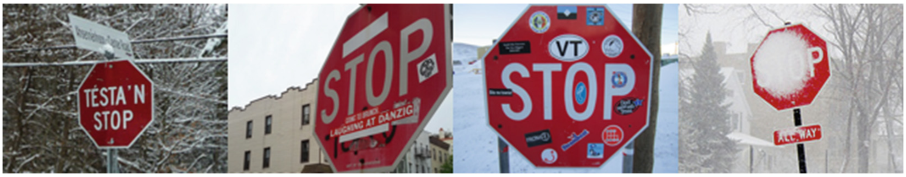


Fig. 5. Example stop signs from the MS-COCO dataset. Stop signs can vary by language, by degree of occlusion by stickers or modification by graffiti, or just elements of the weather. Each stop sign in the images is correctly detected by the object detector with high confidence (99%, 99%, 99%, and 64%, respectively).

[4] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md.

6 Discussion and Future Work

There is considerable variation in the physical world that real systems will have to deal with. Figure 5 shows a curated set of non-standard examples of stop signs from the MS-COCO dataset[5]. The examples show stop signs in a different language, or that have graffiti or stickers applied to them, or that have been occluded by the elements. In each of these cases, it is very unlikely a human would misinterpret the sign as anything else but a stop sign. They each have the characteristic octagonal shape and are predominantly red in color. Yet, the object detector sees something else.

Unlike previous work on adversarial examples for image recognition, our adversarial perturbations are overt. They, like the examples in Fig. 5, exhibit large deviations from the standard stop sign. A human would probably notice these large deviations, and a trained human might even guess they were constructed to be adversarial. But they probably would not be fooled by our perturbations. However an automated-system using an off-the-shelf object detector would be fooled, as our results show. Our digital perturbation shown in Fig. 2e does look like a baseball or tennis ball has been painted on the upper right hand corner. Figure 4b shows how the object detector detects this part of the image as a sports ball with high confidence. This might seem unfair, but attackers have much latitude when these kind of models are deployed in automated systems. Even in non-automated systems a human might not think anything of Fig. 2d because it does not exhibit any recognizable person-like features.

Attackers might also generate perturbations without restricting the shape and color, and attach them to some arbitrary objects, like a street light or a trash bin. An untrained eye might see these perturbations as some kind of artwork, but the autonomous system might see something completely different. This attack, as described in [20], could be extended to object detectors using our method.

Defending against these adversarial examples has proven difficult. Many defenses fall prey to the so-called "gradient masking" or "gradient obfuscating" problem [2]. The most promising defense, adversarial training, has yet to scale up to models with good performance on the ImageNet dataset. Whether adversarial training can mitigate our style of overt, large-deviation (e.g., large ℓ_p distance) perturbations is also unclear.

7 Conclusion

We show that the state-of-the-art Faster R-CNN object detector, while previously considered more robust to physical adversarial attacks, can actually be

[5] Full resolution images of the examples in Fig. 5 can be found at: http://cocodataset.org/#explore?id=315605, http://cocodataset.org/#explore?id=214450, http://cocodataset.org/#explore?id=547465, and http://cocodataset.org/#explore?id=559484.

attacked with high confidence. Our work demonstrates vulnerability in MS-COCO-learned object detectors and posits that security and safety critical systems need to account for the potential threat of adversarial inputs to object detection systems.

Many real-world systems probably do not use an off-the-shelf pre-trained object detector as in our work. Why would a system with safety or security implications care to detecting sports balls? Most probably do not. Although it remains to be shown whether our style of attack can be applied to safety or security critical systems that leverage object detectors, our attack provides the means to test for this new class of vulnerability.

Acknowledgements. This work is supported in part by NSF grants CNS-1704701, TWC-1526254, and a gift from Intel.

References

1. Akhtar, N., Mian, A.S.: Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access **6**, 14410–14430 (2018)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In: Proceedings of the 35th International Conference on Machine Learning (2018)
3. Athalye, A., Sutskever, I.: Synthesizing robust adversarial examples. In: Proceedings of the 35th International Conference on Machine Learning (2018)
4. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch. arXiv preprint arXiv:1712.09665 (2017)
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: Proceedings of the 38th IEEE Symposium on Security and Privacy, pp. 39–57 (2017)
6. Evtimov, I., et al.: Robust physical-world attacks on machine learning models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
7. Eykholt, K., et al.: Note on attacking object detectors with adversarial stickers. arXiv preprint arXiv:1712.08062 (2017)
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015). https://openreview.net/forum?id=BJm4T4Kgx
9. Huang, J., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3296–3297 (2017)
10. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. In: International Conference on Learning Representations (Workshop) (2017). https://openreview.net/forum?id=HJGU3Rodl
11. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
12. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. In: International Conference on Learning Representations (2017). https://openreview.net/forum?id=Sys6GJqxl

13. Lu, J., Sibai, H., Fabry, E.: Adversarial examples that fool detectors. arXiv preprint arXiv:1712.02494 (2017)
14. Lu, J., Sibai, H., Fabry, E., Forsyth, D.: No need to worry about adversarial examples in object detection in autonomous vehicles. arXiv preprint arXiv:1707.03501 (2017)
15. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
16. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 12th ACM on Asia Conference on Computer and Communications Security, pp. 506–519 (2017)
17. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6517–6525 (2017)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
19. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security, pp. 1528–1540 (2016)
20. Sitawarin, C., Bhagoji, A.N., Mosenia, A., Chiang, M., Mittal, P.: Darts: Deceiving autonomous cars with toxic signs. arXiv preprint arXiv:1802.06430 (2018)
21. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
22. Szegedy, C., et al.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014). https://openreview.net/forum?id=kklr_MTHMRQjG
23. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.L.: Adversarial examples for semantic segmentation and object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1378–1387 (2017)