

Making an Invisibility Cloak: Real World Adversarial Attacks on Object Detectors

Zuxuan Wu

Facebook AI

University of Maryland

Ser-Nam Lim

Facebook AI

Larry Davis

University of Maryland

Tom Goldstein

Facebook AI

University of Maryland

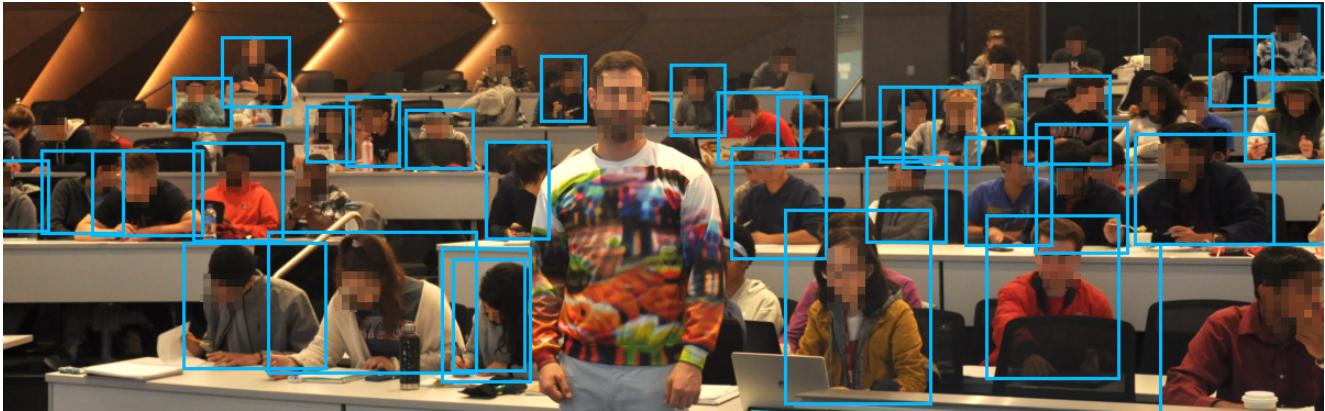


Figure 1: This stylish pullover is a great way to stay warm this winter, whether in the office or on-the-go. It features a stay-dry microfleece lining, a modern fit, and adversarial patterns that evade most common object detectors. In this demonstration, the YOLOv2 detector is evaded using a pattern trained on the COCO dataset with a carefully constructed objective.

Abstract

We present a systematic study of adversarial attacks on state-of-the-art object detection frameworks. Using standard detection datasets, we train patterns that suppress the objectness scores produced by a range of commonly used detectors, and ensembles of detectors. Through extensive experiments, we benchmark the effectiveness of adversarially trained patches under both white-box and black-box settings, and quantify transferability of attacks between datasets, object classes, and detector models. Finally, we present a detailed study of physical world attacks using printed posters and wearable clothes, and rigorously quantify the performance of such attacks with different metrics.

1. Introduction

Adversarial examples are security vulnerabilities of machine learning systems in which an attacker makes small or unnoticeable perturbations to system inputs with the goal of manipulating system outputs. These attacks are most effective in the digital world, where attackers can directly manipulate image pixels. However, most attacks have real

security implications only when they cross into the physical realm. In a “physical” attack, the adversary modifies a real-world *object*, rather than a digital image, so that it confuses systems that observe it. These objects must maintain their adversarial effects when observed with different cameras, resolutions, lighting conditions, distances, and angles.

While a range of physical attacks have been proposed in the literature, these attacks are frequently confined to digital simulations, or are demonstrated against simple classifiers rather than object detectors. Furthermore, many studies of physical attacks present successful examples without quantifying the success rate of the attack, or how different aspects of the training process and environment impact effectiveness. Finally, physical world attacks are usually demonstrated using idealized flat printed patches that do not suffer the complex distortions needed to attack 3D objects like people, airplanes, or cars.

In this paper, we study the art and science of crafting physical attacks against detectors, with the ultimate goal of producing wearable textiles. Our study has the following goals:

- We focus on industrial-strength detectors. Unlike *classifiers*, which output one feature vector per image, object *detectors* output a map of vectors, one for each

prior (i.e., candidate bounding box), centered at each output pixel. Since any of these priors can detect an object, attacks must simultaneously manipulate hundreds or thousands of priors operating at different positions, scales, and aspect ratios.

- We break down the incremental process of getting attacks out of a digital simulation and into the real world. We explore how real-world nuisance variables cause major differences between the digital and physical performance of attacks, and present experiments for quantifying and identifying the sources of these differences.
- We *quantify* the success rate of attacks under various conditions, and measure how algorithm and model choices impact success rates. We rigorously study how attacks degrade classifiers using standard metrics (AP), and also more interpretable success/fail metrics.
- We push physical attacks to their limits by creating wearable adversarial clothing, and quantify the success rate of our attacks under complex fabric distortions.

2. Related Work

Attacks on object detection and semantic segmentation. While there is a plethora of work on attacking image classifiers [18, 8, 16], less work has been done on more complex vision tasks like object detection and semantic segmentation. Metzen *et al.* demonstrate that nearly imperceptible adversarial perturbations can fool segmentation models to produce incorrect outputs [17]. Arnab *et al.* also show that segmentation models are vulnerable to attacks [1], and claim that adversarial perturbations fail to transfer across network architectures. Xie *et al.* introduce Dense Adversary Generation (DAG), a method that produces incorrect predictions for pixels in segmentation models or proposals in object detection frameworks [26]. Wei *et al.* further extend the attack from images to videos [25]. In contrast to [26, 25], which attack the classifier stage of object detectors, Li *et al.* attack region proposal networks by decreasing the confidence scores of positive proposals [13]. Note that all of these studies focus on digital (as opposed to physical) attacks with a specific detector. In this paper, we systematically evaluate a wide range of popular detectors in both the digital and physical world.

Physical attacks in the real world. Kurakin *et al.* took photos of adversarial images with a camera and input them to a pretrained image classifier [12]; they demonstrate that a large fraction of images are misclassified. Eykholt *et al.* consider physical attacks on stop sign classifiers using images cropped from video frames [6]. They successfully fool classifiers using both norm bounded perturbations, and also sparse perturbations using carefully placed stickers.

Lu *et al.* showed that the perturbed sign images from [6] can be reliably recognized by popular detectors like Faster-RCNN [20] and Yolov2 [19], and showed that detectors are much more robust to attacks than classifiers.

Sitawarin *et al.* [22] propose large out-of-distribution perturbations, producing toxic signs to deceive autonomous vehicles. Athalye *et al.* introduce expectation over transformations (EoT) to generate physically robust adversarial samples, and they produce 3D physical adversarial objects that can attack classifiers in different conditions. Sharif *et al.* explore adversarial eyeglass frames that fool face classifiers [21]. Brown *et al.* placed adversarial patches [3] on raw images, forcing classifiers to output incorrect predictions. Komkov *et al.* generate stickers attached to hats to attack face classifiers [11]. Inspired by the work above, Thys *et al.* produce printed adversarial patches [23] that deceive person detectors instantiated by Yolov2 [19]. This proof-of-concept study was the first to consider physical attacks on detectors, although it was restricted to the white box setting (attacker knows the model and model parameters). Furthermore the authors did not quantify success rates, or address issues like robustness to distance/distortions and detectors beyond Yolov2.

2.1. Object detector basics

We briefly review the inner workings of object detectors, most of which can be described as either two-stage frameworks (*e.g.*, Fast RCNN [7], Faster RCNN [20], Mask RCNN [9], *etc.*) or one-stage frameworks (*e.g.*, YOLOv2 [19], SSD [15], *etc.*).

Two-stage detectors These detectors use a region proposal network (RPN) to identify potential bounding boxes (Stage I), and then classify the contents of these bounding boxes (Stage II). An RPN passes an image through a *backbone* network to produce a stack of 2D feature maps with resolution $W' \times H'$ (or a feature pyramid containing features at different resolutions). The RPN considers k “priors”, or candidate bounding boxes with a range of aspect ratios and sizes, centered on every output pixel. For each of the $W' \times H' \times k$ priors, the RPN produces an “objectness score”, and also the offset to the center coordinates and dimensions of the prior to best fit the closest object. Finally, proposals with high objectness scores are sent to a Stage-II network for classification.

One-stage detectors These networks generate object proposals and at the same time predict their class labels. Similar as RPNs, these networks typically transform an image into a $W' \times H'$ feature map, and each pixel on the output contains the locations of a set of default bounding boxes, their class prediction scores, as well as objectness scores.

Why are detectors hard to fool? A detector usually produces hundreds or thousands of priors that overlap with an

object. Usually, non-maximum suppression (NMS) is used to select the bounding box with highest confidence, and reject overlapping boxes of lower confidence so that an object is only detected once. Suppose an adversarial attack evades detection by one prior. In this case, the NMS will simply select a different prior to represent the object. For an object to be completely erased from an image, the attack must simultaneously fool the ensemble of all priors that overlap with the object—a much harder task than fooling the output of a single classifier.

3. Approach

Our goal is to generate an adversarial pattern that, when placed over an object either digitally or physically, makes that object invisible to detectors. Furthermore, we expect the pattern to be (1) universal (image-agnostic)—the pattern must be effective against a range of objects and within different scenes; (2) transferable—it breaks a variety of detectors with different backbone networks; (3) dataset agnostic—it should fool detectors trained on disparate datasets; (4) robust to viewing conditions—it can withstand field-of-view changes when observed from different perspectives and distances; (5) realizable—patterns should remain adversarial when printed over real-world 3D objects.

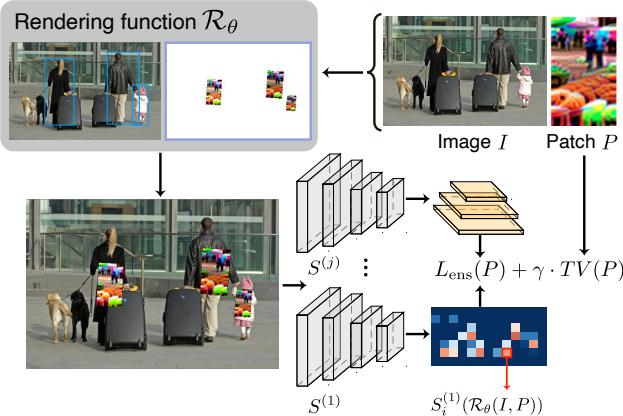


Figure 2: An overview of the framework. Given a patch and an image, the rendering function uses translations and scalings, plus random augmentation transforms, to overlay the patch onto detected persons. The patch is then updated to minimize the objectness scores produced by a detector while maintaining patch smoothness.

3.1. Creating a universal adversarial patch

Our strategy is to “train” a patch using a large set of images containing people. On each training iteration, we draw a random batch of images, and pass them through an object detector to obtain bounding boxes containing people.

We then place a randomly transformed patch over each detected person, and update the patch pixels to minimize the objectness scores in the output feature map.

More formally, we consider a patch $P \in \mathbb{R}^{w \times h \times 3}$ and a randomized rendering function \mathcal{R}_θ . The rendering function takes a patch P and image I , and renders a rescaled copy of P over every detected person in the image I . In addition to scaling and translating the patch to place it into each bounding box, the rendering function also applies an augmentation transform parameterized by the (random) vector θ . These transforms are a composition of brightness, contrast, rotation, translation, and sheering transforms that help make patches robust to variations caused by lighting and viewing angle that occur in the real world. We also consider more complex thin-plate-spline (TPS) transforms to simulate the random “crumpling” of fabrics.

A detector network takes a patched image $\mathcal{R}_\theta(I, P)$ as input, and outputs a vector of objectness scores, $\mathcal{S}(\mathcal{R}_\theta(I, P))$ one for each prior. These scores rank general objectness for a two-stage detector, and the strength of the “person” class for a one-stage detectors. A positive score is taken to mean that an object/person overlaps with the corresponding prior, while a negative score denotes the absence of a person. To minimize the objectness scores, we formulate the objectness loss function

$$L_{\text{obj}}(P) = \mathbb{E}_{\theta, I} \sum_i \max\{\mathcal{S}_i(\mathcal{R}_\theta(I, P)) + 1, 0\}^2. \quad (1)$$

Here, i indexes the priors produced by the detector’s score mapping. The loss function penalizes any objectness score greater than -1 . This suppresses scores that are positive, or lie very close to zero, without wasting the “capacity” of the patch on decreasing scores that are already far below the standard detection threshold. We minimize the expectation over the transform vector θ as in [2] to promote robustness to real-world distortions, and also the expectation over the random image I drawn from the training set.

Finally, we add a small total-variation penalty to the patch. We do this because there are pixels in the patch that are almost never used by the rendering function \mathcal{R}_θ , which almost always sub-samples the patch before rendering it onto the image. The TV penalty helps ensure a smooth patch in which all pixels in the patch get optimized. The final optimization problem we solve is

$$\underset{P}{\text{minimize}} L_{\text{obj}}(P) + \gamma \cdot TV(P), \quad (2)$$

where γ was chosen to be small enough to prevent outlier pixels without visibly distorting the patch.

Ensemble training To help adversarial patterns generalize to detectors that were not used for training (*i.e.*, to create a black-box attack), we also consider training patches that

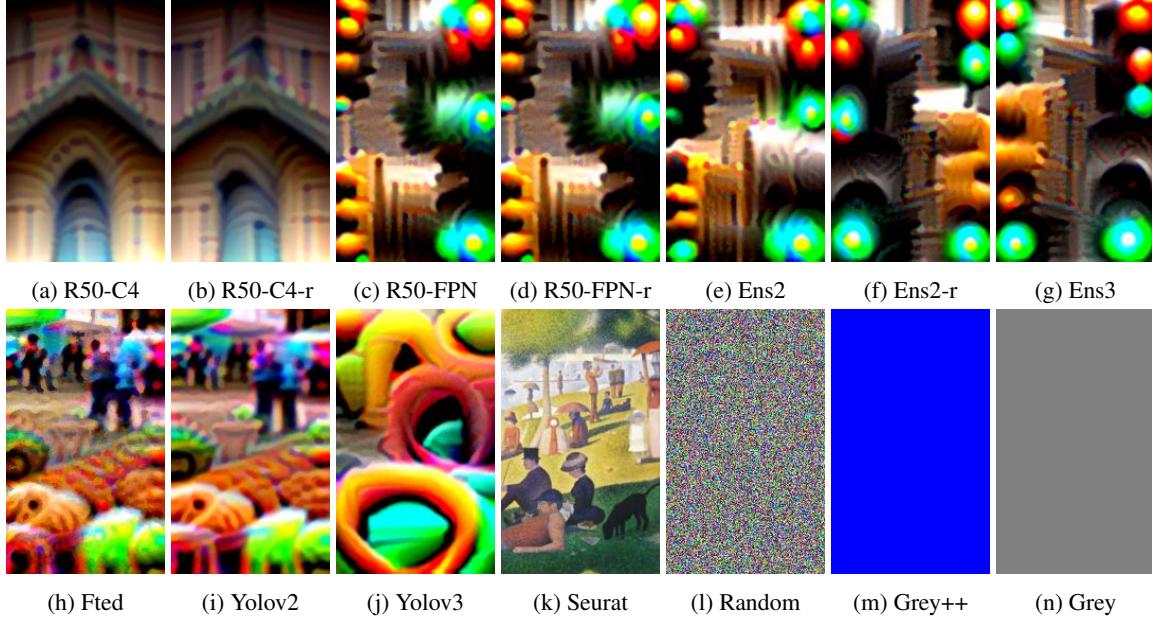


Figure 3: **Adversarial patches** crafted using a range of object detectors.

fool an ensemble of detectors. In this case we replace the objectness loss (1) with the ensemble loss

$$L_{\text{ens}}(P) = \mathbb{E}_{\theta, I} \sum_{i,j} \max\{\mathcal{S}_i^{(j)}(\mathcal{R}_\theta(I, P)) + 1, 0\}^2, \quad (3)$$

where $\mathcal{S}_i^{(j)}$ denotes the j th detector in an ensemble.

4. Crafting attacks in the digital world

Datasets and metrics We craft attack patches using the COCO dataset,¹ which contains a total of 123,000 images. After removing images from the dataset that do not contain people, when then chose a random subset of 10,000 images for training. For both physical and digital attacks, we compute average precision (AP) for the category of interest to measure the effectiveness of patches. For physical attacks, we further compute success rates to quantify the performance of patches, as will be explained in Sec 5.

Object detectors attacked We experiment with both one-stage detectors, *i.e.*, YOLOv2 and YOLOv3, and two-stage detectors, *i.e.*, R50-C4 and R50-FPN, both of which are based on Faster RCNN with a ResNet-50 [10] backbone. R50-C4 and R50-FPN use different features for region proposal—R50-C4 uses single-resolution features, while R50-FPN uses a multi-scale feature pyramid. For all these detectors, we adopt standard models pre-trained on COCO, in addition to our own models retrained from scratch (models denoted with “-r”) to test for attack transferability across

network weights. Finally, we consider patches crafted using three different ensembles of detectors—ENS2: YOLOv2 + R50-FPN, ENS2-r: YOLOv2 + R50-FPN-r, and ENS3-r: YOLOv2 + YOLOv3 + R50-FPN-r.

Implementation details We use PyTorch for implementation, and we initialize training with a random uniform patch of size $3 \times 250 \times 150$ (note, the patch is dynamically resized by the rendering function during the forward pass). We use the Adam optimizer with learning rate 10^{-3} , and decay the rate every 100 epochs until 400 is reached. For YOLOv2/v3, images are resized to 640×640 for both training and testing. For Faster RCNN detectors, the shortest side of images is 250^2 for training, and 800 for testing.

4.1. Evaluation of digital attacks

We begin by evaluating patches in digital simulated settings: we consider white-box attacks (detector weights are used for patch learning) and black-box attacks (patch is crafted on a surrogate model and tested on a victim model with different parameters).

Effectiveness of learned patches for white-box attack We optimize patches using the aforementioned detectors, and denote the learned patch with the corresponding model it is trained on. We further compare with the following alternative patches: (1) FTED, a learned YOLOv2 patch that is further fine-tuned on a R50-FPN model; (2) SEURAT, a crop from the famous painting “A Sunday Afternoon on the Island of La Grande Jatte” by Georges Seurat, which is visually

¹We focus on the COCO dataset for its wide diversity of scenes, although we consider the effect of the dataset later.

²We found that using a lower resolution for training produced more effective attacks on these detectors.

Victim Patch \	R50-C4	R50-C4-r	R50-FPN	R50-FPN-r	YOLOv2	YOLOv2-r	YOLOv3	YOLOv3-r
R50-C4	24.5	24.5	31.4	31.4	37.9	42.6	57.6	48.3
R50-C4-r	25.4	23.9	30.6	30.2	37.7	42.1	57.5	47.4
R50-FPN	20.9	21.1	23.5	19.6	22.6	12.9	40.2	40.3
R50-FPN-r	21.5	21.7	25.4	18.8	17.6	11.2	37.5	36.9
YOLOv2	21.1	19	21.5	21.4	10.7	7.5	18.1	25.7
YOLOv3	28.3	28.9	31.5	27.2	20	15.9	17.8	36.1
FTED	25.6	23.9	24.2	24.4	18.9	16.4	31.6	28.2
ENS2	20	20.3	23.2	19.3	17.5	11.3	39	38.8
ENS2-r	19.7	20.2	23.3	16.8	14.9	9.7	36.3	34.1
ENS3-r	21.1	21.4	24.2	17.4	13.4	9.0	29.8	33.6
SEURAT	47.9	52	51.6	52.5	43.4	39.5	62.6	57.1
RANDOM	53	58.2	59.8	59.7	52	52.5	70	63.5
GREY	45.9	49.6	50	50.8	48	47.1	65.6	57.5
GREY++	46.5	49.8	51.4	52.7	48.5	49.4	64.8	58.6
CLEAN	78.7	78.7	82.2	82.1	63.6	62.7	81.6	74.5

Table 1: **Impact of different patches on various detectors, measured using average precision (AP).** The left axis lists patches created by different methods, and the top axis lists different victim detectors. Here, “r” denotes retrained weights instead of pretrained weights downloaded from model zoos.

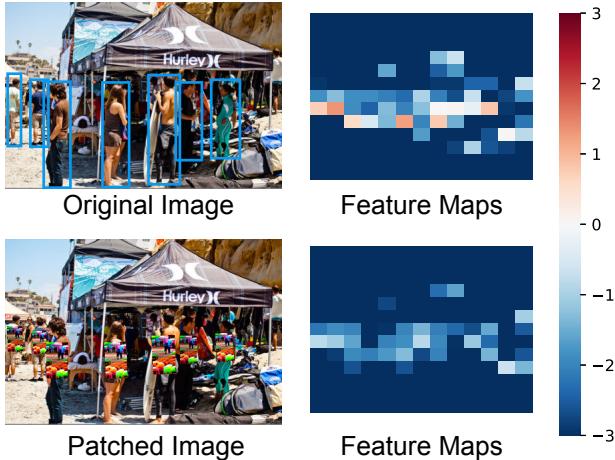


Figure 4: **Images and their corresponding feature maps,** with and without patches, using YOLOv2. Each pixel in the feature map represents an objectness score.

similar to the top-performing YOLOv2 patch with objects like persons, umbrellas *etc.* (See Figure 7d); (3) GREY, a grey patch; (4) GREY++, the most powerful RGB value for attacking Yolov2 using COCO; (5) RANDOM, a randomly initialized patch; (6) CLEAN, which corresponds to the oracle performance of detectors when patches are not applied.

Patches are shown in Fig 3, and results are summarized in Table 1. We can observe that all adversarially learned patches are highly effective in digital simulations, where

the AP of all detectors degrades by at least 29%, going as low as 7.5% AP when the YOLOv2 patch is tested on the retrained YOLOv2 model (YOLOv2-r). Interestingly, all patches transferred well to the corresponding retrained models. In addition, the ensemble patches perform better compared to Faster RCNN patches but are worse than YOLO patches. It is also interesting to see that YOLO patches can be effectively transferred to Faster RCNN models, while Faster RCNN patches are not very effective at attacking YOLO models. Although the SEURAT patch is visually similar to the learned YOLOv2 patch, it does not consistently perform better than GREY.

We visualize the impact of the patch in Figure 4, which shows objectness maps from the YOLOv2 model with and without patches. We can see that when patches are applied to persons, the corresponding pixels in the feature maps are indeed suppressed.

Transferability across backbones We also investigate whether the learned patches transfer to detectors with a range of backbones. We evaluate the patches on the following detectors: (1) R101-FPN, which uses ResNet-101 together with FPN as its backbone; (2) X101-FPN, replaces the feature extractor of R101-FPN with ResNeXt-101 [27]; (3) R50-FPN-M, a Mask RCNN model [9] based on R50-FPN; (4) X101-FPN-M, a Mask RCNN model based on X101-FPN; (5) RETINANET-R50, a RetinaNet [14] with a backbone of ResNet-50; (6) FCOS, a recent anchor-free framework [24] based on R50-FPN. The results are shown in Figure 5. We observe that all these learned adversarial

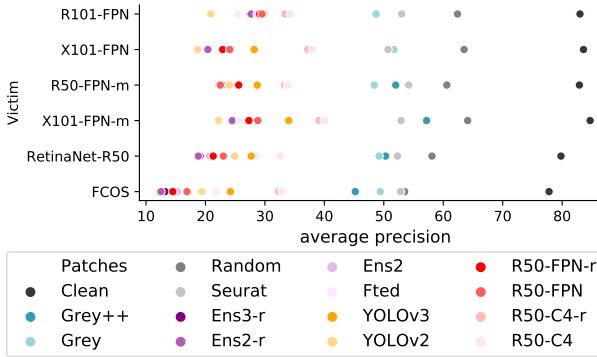


Figure 5: **Performance of different patches**, when tested on detectors with different backbones.

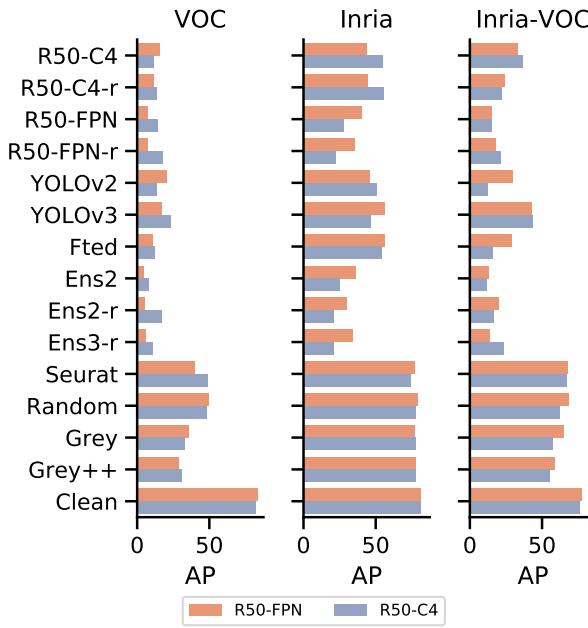


Figure 6: **Results of different patches, trained on COCO, tested on the person category of different datasets.** Left two panels: COCO patches tested on VOC and Inria, respectively, using backbones learned on COCO; The right-most panel: COCO patches tested on Inria with backbones trained on VOC.

patches can significantly degrade the performance of the person category even using models that they have not been trained on.

Transferability across datasets We further demonstrate the transferability of patches learned on COCO to other datasets including Pascal VOC 2007 [5] and the Inria person dataset [4]. We evaluate the patches on the person category using R50-FPN and R50-C4, and the results are presented in Figure 6. The left two panels correspond to results of differ-

ent patches when evaluated on VOC and Inria, respectively, with both the patches and the models trained on COCO; the rightmost panel shows the APs of these patches when applied to Inria images using models trained on VOC. We can see that ensemble patches offer the most effective attacks, degrading the AP of the person class by large margins. This confirms that these patches can transfer not only to different datasets but also backbones trained with different data distributions. From the right two panels, we can see that weights learned on COCO are more robust than on VOC.

Transferability across classes We find that patches effectively suppress multiple classes, even when they are trained to suppress only one. In addition to the “person” patch, we also train patches on the “bus” and “horse” classes of COCO, and then evaluate these patches on all 20 categories in VOC³. Table 2 summarizes the results. We can see that the “person” patch transfers to almost all categories, possibly because they co-occur with most classes. We also compare with the GREY patch to rule out the possibility that the performance drops are due to occlusion.

5. Physical world attacks

We now move on to discuss the results of physical world attacks with printed posters. In addition to the standard average precision⁴, we also quantify the performance of attacks with “success rates,” which we define as (1) a SUCCESS attack: when there is no bounding box predicted for the person with adversarial patterns; (2) a PARTIAL SUCCESS attack: when there is a bounding box covering less than 50% of a person; (3) a FAILURE attack: when the person is successfully detected. Examples of detections in each category are shown in Figure 7. For computing these scores, we use a cutoff rate zero for YOLOv2 and we tune the threshold of other detectors to achieve the best F-1 score on the COCO minival set.

5.1. Printed posters

We printed posters and took photos at 15 different locations using 10 different patches. At each location, we took four photos for each printed patch corresponding to two distances from the camera and two heights where the patch is held. We also took photos without printed posters as controls (CONTROL). In total, we collected 630 photos (see the bottom row of Figure 7 for examples). We use four patches that perform well digitally (*i.e.*, YOLOv2, ENS2, ENS3, FTED), and three baseline patches (SEURAT patch, FLIP patch, WHITE).

To better understand the impact of the training process on attack power, we also consider several variants of

³We observe similar trends on COCO.

⁴We only consider the person with adversarial patterns to calculate AP by eliminating boxes without any overlapping with the GT box.

Patch	Class	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
PERSON		2.0	14.6	1.0	1.8	2.7	13.5	10.7	2.3	0.1	2.4	6.4	2.3	8.3	12.3	5.5	0.3	2.2	1.3	3.8	12.4
HORSE		5.0	31.9	4.7	4.1	2.5	26.4	17.6	10.6	2.3	26.0	24.7	9.5	27.9	26.6	16.0	7.6	12.4	13.4	13.2	35.3
BUS		3.1	30.6	8.5	4.4	1.9	18.4	15.6	7.8	2.7	25.7	39.8	5.3	20.8	20.7	16.0	8.9	12.3	9.5	9.3	29.5
GREY		3.0	19.0	6.4	14.6	8.5	26.9	19.6	9.9	9.8	28.6	24.4	7.4	22.7	15.9	35.8	6.1	18.7	8.7	11.4	61.8
CLEAN		77.5	82.2	76.3	63.6	64.5	82.9	86.5	83.0	57.2	83.3	66.2	84.9	84.5	81.4	83.3	48.0	76.7	70.1	80.1	75.4

Table 2: **Transferability of patches across classes from VOC**, measured with average precision (AP).

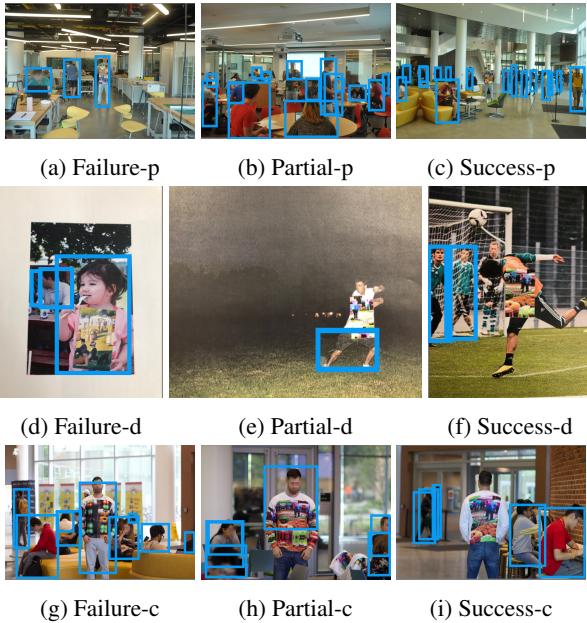


Figure 7: **Examples of attack failure, partial success, and full success**, using posters (top) paper dolls (middle), and shirts (bottom).

the YOLOv2 patch (the best digital performer). To assess whether the learned patterns are “truly” adversarial, or whether any qualitatively similar pattern would work, we add the FLIP patch, which is the YOLOv2 patch held upside-down. We compare to a TPS patch, which uses thin plate spline transformations to potentially enhance robustness to warping in real objects. We consider a YOLOv2-noaug patch, which is trained to attack the YOLOv2 model without any augmentations/transformations beyond resizing. To observe the effect of the dataset, we add the YOLOv2-Inria patch, which is trained on the Inria dataset as opposed to COCO.

Poster results Figure 8a and 8c summarize the results. We can see that compared to baseline patches, adversarially learned patches successfully degrade the performance of detectors measured by both AP and success rates. The YOLOv2 patch achieves the best performance measured by AP among all patches. R50-FPN is the most robust model with slight degradation when patches are applied. FCOS is

the most vulnerable network; it fell to the YOLOv2 patch even though we never trained on an anchor-free detector, let alone FCOS. This may be because anchor-free models predict the “center-ness” of pixels for bounding boxes, and the performance drops when center pixels of persons are occluded by printed posters. Interestingly though, simply using baseline patches for occlusion fails to deceive FCOS.

Beyond the choice of detector, several other training factors impact performance. Surprisingly, the TPS patch is worse than YOLOv2, and we believe this results from the fact that adding such complicated transformation makes optimization more difficult during training. It is also surprising to see that the YOLOv2-Inria patch offers impressive success rates on YOLOv2, but it does not transfer as well to other detectors. Not surprisingly, the YOLOv2 patch outperforms the YOLOv2-noaug in terms of AP, however these gains shrink when measured in terms of success rates.

We included the FLIP patch to evaluate whether patches are generic, i.e., any texture with similar shapes and scales would defeat the detector, or whether they are “truly adversarial.” The poor performance of the FLIP patch seems to indicate that the learned patches are exploiting specialized behaviors learned by the detector, rather than a generic weakness of the detector model.

From the left column of Figure 8 and Table 1, we see that performance in digital simulations correlates well with physical world performance. However, we observe that patches loose effectiveness when transferring from the digital world into the physical world, demonstrating that physical world attacks are more challenging.

5.2. Paper dolls

We found that a very useful technique for crafting physical attacks was to make “paper dolls”—printouts of test images that we could dress up with different patches at different scales. Paper dolls facilitate quick experiments with physical world effects and camera distortions without the time and expense of fabricating textiles.

In this section, we use paper dolls to gain insights into why physical attacks are not as effective as digital attacks. The reasons might be three-fold: (1) Pixelation at the detector and compression algorithms incur subtle changes; (2) the rendering artifacts around patch borders assists digital attacks; (3) there exists differences in appearance and

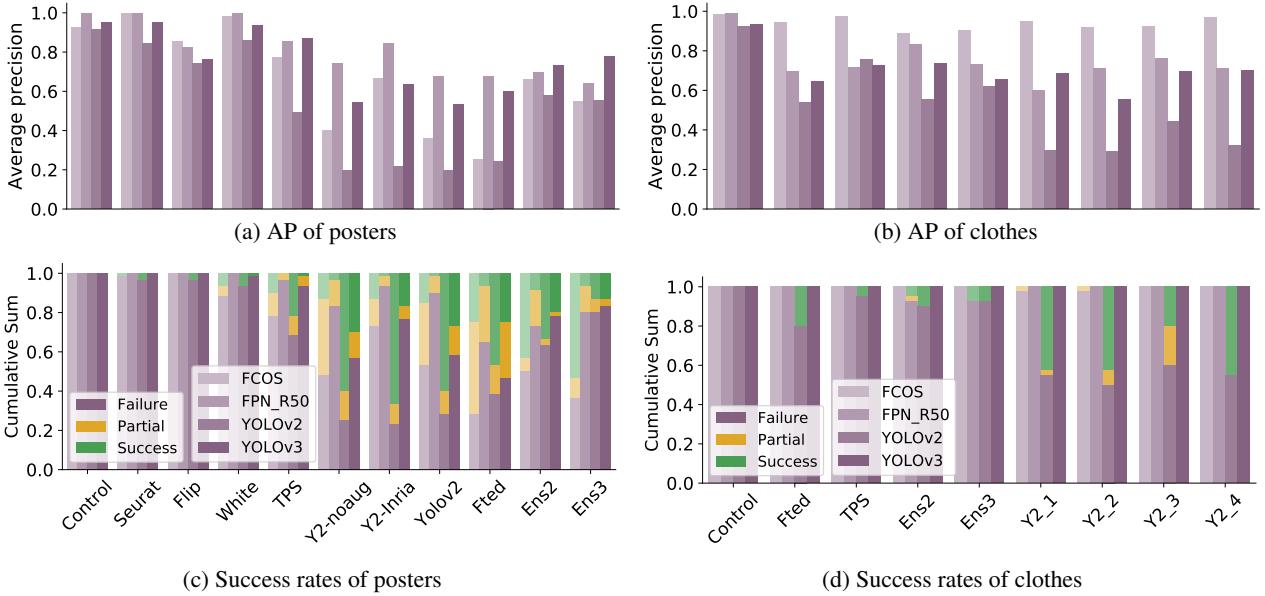


Figure 8: **AP and success rates for physical attacks.** Top: average precision of different printed posters (left) and clothes (right). Lower is better. Bottom: success rates of different printed posters (left) and clothes (right). Y2 denotes YOLOV2.

texture between large-format digital patches and the original digital patch.



Figure 9: **Paper dolls** are made by dressing up printed images with paper patches. We use dolls to observe the effects of camera distortions, and “scrumpled” patches to test against physical deformations that are not easily simulated.

In our paper doll study, we print out patches and photos separately. We then overlay patches onto objects and photograph them. We used the first 20 images from the COCO minival set. We use the same patches from the poster experiment, we also compare with “scrumpled” versions of YOLOV2, *i.e.*, YOLOV2-s1 and YOLOV2-s2, to test for robustness to physical deformation, where “-s1” and “-s2” denote the level of crumpling (“s1” < “s2”, see Figure 9).

We compute success rates of different patches when tested with YOLOv2 and present the results in Figure 10. Comparing across Figure 10 and the left side of Figure 8, we see that paper dolls perform only slightly better than large-format posters. The performance drop of paper dolls compared to digital simulations, combined with the high

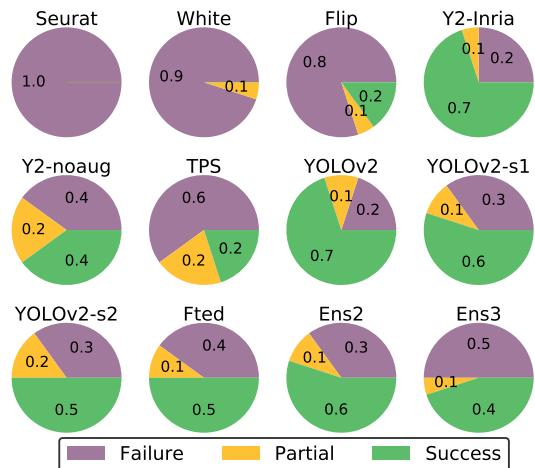


Figure 10: **Effectiveness of different patches on paper dolls.** Y2 denotes YOLOV2.

fidelity of the paper printouts, leads us to believe that the dominant factor in the performance difference between digital and physical attacks can be attributed to the imaging process, like camera post-processing, pixelation, and compression.

6. Wearable adversarial examples

Printed posters provide a controlled setting under which to test the real-world transferability of adversarial attacks. However the success of printed posters does not inform us

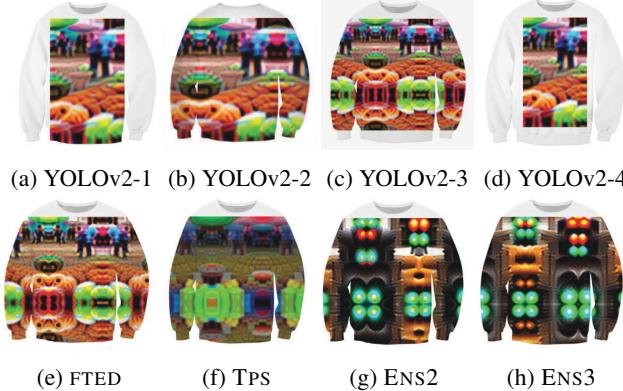


Figure 11: **Adversarial shirts tested in Section 6.**

about whether attacks can survive the complex deformations and textures of real objects.

To experiment with complex real-world transfer, we printed adversarial patterns on shirts using various strategies. We consider four versions of the YOLOV2 patch representing two different scalings of the patch, both with and without boundary reflections to cover the entire shirt (see Figure 11). We also consider the TPS patch to see if complex data augmentation can help the attack survive fabric deformations. Finally, we include the FTED, ENS2, ENS3 patches to see if these more complex crafting methods facilitate transfer. We collected photos of a person wearing these shirts at ten different locations. For each location and shirt, we took 4 photos with two orientations (front and back) and two distances from the camera. We also took control photos where the person was not wearing an attack. We collected 360 photos in total.

We tested the collected images under the same settings as the poster study, and measure the performance of the patches using both AP and success rates. The results are shown in Figure 8b and Figure 8d. A gallery of selected images is shown in Figure 12. We can see that these wearable attacks significantly degrade the performance of detectors. This effect is most pronounced when measured in AP because, when persons are detected, they tend to generate multiple fragmented boxes. It is also interesting to see that FCOS, which is vulnerable to printed posters, is quite robust with wearable attacks, possibly because shirts more closely resemble the clothing that appears in the training set. When measured in success rates, sweatshirts with YOLOV2 patterns achieve ~50% success rates, yet they do not transfer well to other detectors. Among all YOLOV2 shirts, smaller patterns (*i.e.*, YOLOv2-2) perform worse as compared to larger patterns. We also found that tiling/reflecting a patch to cover the whole shirt did not negatively impact performance, even though the patch was not designed for this use. Finally, we found that augmenting adversarial patterns with non-rigid TPS transforms did not improve transferability,

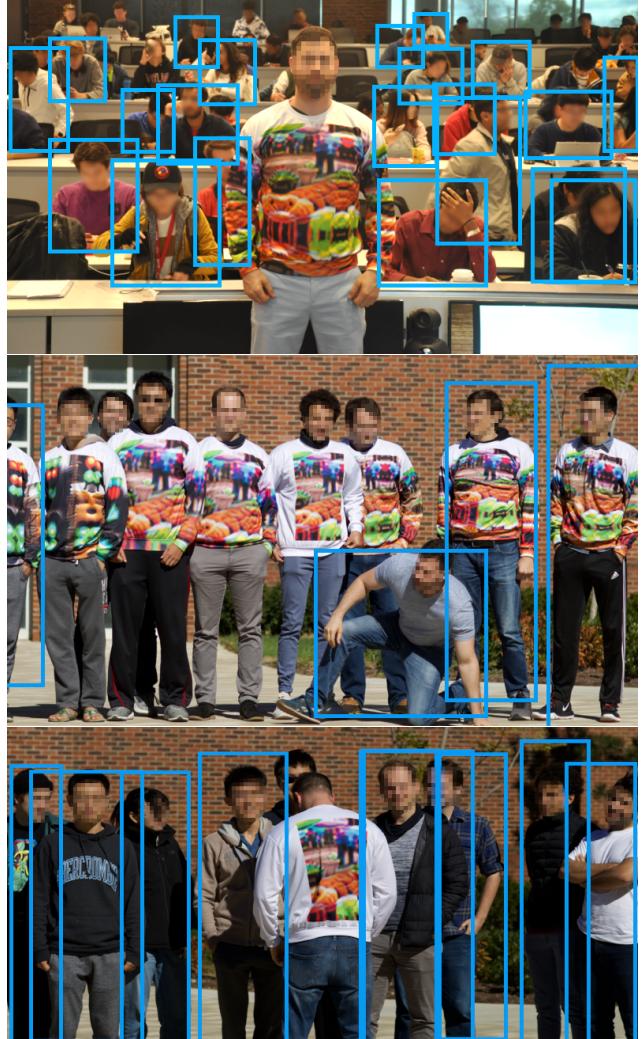


Figure 12: **Selected images of adversarial clothes.**

and in fact was detrimental. This seems to be a result of the difficulty of training a patch with such transformations, as the patch also under-performs other patches digitally.

7. Conclusion

It is widely believed that fooling detectors is a much harder task than fooling classifiers; the ensembling effect of thousands of distinct priors, combined with complex texture, lighting, and measurement distortions in the real world, makes detectors naturally robust. Despite these complexities, the experiments conducted here show that digital attacks can indeed transfer between models, classes, datasets, and also into the real world, although with less reliability than attacks on simple classifiers.

Acknowledgements Thanks to Ross Girshick for his invaluable insights into object detectors, and for helping us refine and improve our experiments.

References

- [1] Anurag Arnab, Ondrej Miksik, and Philip HS Torr. On the robustness of semantic segmentation models to adversarial attacks. In *CVPR*, 2018. [2](#)
- [2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*, 2018. [3](#)
- [3] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. [2](#)
- [4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. [6](#)
- [5] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. [6](#)
- [6] 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 models. In *CVPR*, 2018. [2](#)
- [7] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. [2](#)
- [8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. [2](#)
- [9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. [2](#), [5](#)
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [4](#)
- [11] Stepan Komkov and Aleksandr Petushko. Advhat: Real-world adversarial attack on arcface face id system. *arXiv preprint arXiv:1908.08705*, 2019. [2](#)
- [12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR Workshop*, 2017. [2](#)
- [13] Yuezun Li, Daniel Tian, Ming-Ching Chang, Xiao Bian, and Siwei Lyu. Robust adversarial perturbation on deep proposal-based models. In *BMVC*, 2018. [2](#)
- [14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. [5](#)
- [15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. [2](#)
- [16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. [2](#)
- [17] Jan Hendrik Metzen, Mummadri Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *ICCV*, 2017. [2](#)
- [18] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017. [2](#)
- [19] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. [2](#)
- [20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. [2](#)
- [21] 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 CCS*, 2016. [2](#)
- [22] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Moseinia, Mung Chiang, and Prateek Mittal. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*, 2018. [2](#)
- [23] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPR Workshop*, 2019. [2](#)
- [24] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019. [5](#)
- [25] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. Transferable adversarial attacks for image and video object detection. In *IJCAI*, 2019. [2](#)
- [26] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, 2017. [2](#)
- [27] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. [5](#)