

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

Zuxuan Wu<sup>1,2</sup>, Ser-Nam Lim<sup>2</sup>, Larry S. Davis<sup>1</sup>, and Tom Goldstein<sup>1,2</sup>

<sup>1</sup>University of Maryland, College Park <sup>2</sup>Facebook AI

**Abstract.** We present a systematic study of the transferability 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.



Fig. 1: In this demonstration, the YOLOv2 detector is evaded using a pattern trained on the COCO dataset with a carefully constructed objective.

## 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, many studies assume a white box threat model, in which the attacker knows the dataset, architecture, and model parameters used by the victim. In addition, 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. However, in most realistic situations the attacker has only black or grey box knowledge, and their attack must *transfer* from the digital world into the physical world, from the attacker model to the victim model, or from models trained on one dataset to another.

In this paper, we study the transferability of attacks on object detectors across different architectures, classes, and datasets, with the ultimate goal of generalizing digital attacks to the real-world. Our study has the following goals:

- We focus on industrial-strength detectors under both black-box and white-box settings. 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.
- In the digital setting, we systematically quantify how well attacks on detectors transfer between models, classes and datasets.
- 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 (average precision) that best describe the strength of detectors, and also more interpretable success/failure metrics.
- We push physical attacks to their limits with 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 [25,11,23], 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 [24]. 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 [34]. Wei *et al.* further extend the attack from images to

videos [33]. In contrast to [34,33], which Attacks the classifier stage of object detectors, Li *et al.* attack region proposal networks by decreasing the confidence scores of positive proposals [19]. DPatch causes misclassification of detectors, by placing a patch that does not overlap with the objects of interest [22]. Li *et al.* add imperceptible patches to the background to fool object detectors [18]. Note that all of these studies focus on digital (as opposed to physical) attacks with a specific detector, without studying the transferability of attacks. In this paper, we systematically evaluate a wide range of popular detectors in both the digital and physical world, and benchmark how attacks transfer in different settings.

**Physical attacks in the real world.** Kurakin *et al.* took photos of adversarial images with a camera and input them to a pretrained classifier [16]; 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 [9]. They successfully fool classifiers using both norm bounded perturbations, and also sparse perturbations using carefully placed stickers. Stop signs attacks on object detectors are considered in [8,5]. Lu *et al.* showed that the perturbed sign images from [9] can be reliably recognized by popular detectors like Faster-RCNN [27] and Yolov2 [26], and showed that detectors are much more robust to attacks than classifiers. Note that fooling stop sign detectors differs from fooling person detectors because stop sign perturbations can cover the whole object, whereas our person attacks leave the face, hands, and legs uncovered.

Zeng *et al.* use rendering tools to perform attacks in 3D environments [37]. Sitawarin *et al.* [30] 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 [28]. 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 [15]. Huang *et al.* craft attacks by simulations to cause misclassification of detectors [14]. Thys *et al.* produce printed adversarial patches [31] that deceive person detectors instantiated by Yolov2 [26]. This proof-of-concept study was the first to consider physical attacks on detectors, although it was restricted to the white-box setting. Furthermore the authors did not quantify the performance, or address issues like robustness to distance/distortions and detectors beyond Yolov2, and the transferability of their attacks are limited. Xu *et al.* learn TPS transformations to generate T-shirts [36]. To the best of our knowledge, no paper has conclusively demonstrated the transferability of attacks on object detectors, or quantified the reliability of transfer attacks.

## 2.1 Object detector basics

We briefly review the inner workings of object detectors [20,4,29,17], most of which can be described as two-stage frameworks (*e.g.*, Fast(er) RCNN [10,27], Mask RCNN [12], *etc.*) or one-stage pipelines (*e.g.*, YOLOv2 [26], SSD [21], *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.

**Detection metrics.** In the field of object detection, the standard performance metric is average precision (AP) per class, which balances the trade-off between precision and recall. In contrast, success rates (using a fixed detection threshold) are more often used when evaluating physical attacks, due to their interpretability. However, manually selected thresholds are required to compute success rates. To mitigate this issue, we report both AP (averaging over all confidence thresholds) and success rates.

### 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. Further, 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.

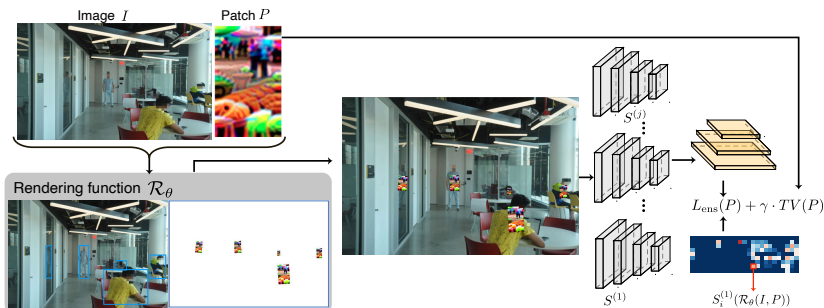


Fig. 2: **An overview of the framework.** Given a patch and an image, the rendering function uses translations and scalings, plus random augmentations, to overlay the patch onto detected persons. The patch is then updated to minimize the objectness scores produced by a detector while maintaining its 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 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  $\mathcal{R}_\theta$ . 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  $\theta$ .

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 its 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 objectness scores, we use 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  $\theta$  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  $\mathcal{R}_\theta$ , which down-samples the high-resolution patch (using linear interpolation) when rendering it onto the image. A small TV penalty ensures a smooth patch in which all pixels in the patch get optimized. A comparison to results without this TV penalty is shown in the supplementary material. The final optimization problem we solve is

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

where  $\gamma$  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 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  $\mathcal{S}^{(j)}$  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,<sup>1</sup> which contains a total of 123,000 images. After removing images from the dataset that do not contain people, we then chose a random subset of 10,000 images for training. We compute average precision (AP) for the category of interest to measure the effectiveness of patches.

**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 [13] 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.

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

Patch	Victim							
	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.

**Implementation details.** We use PyTorch for implementation, and we start with a random uniform patch of size  $3 \times 250 \times 150$  (the patch is dynamically re-sized by  $\mathcal{R}_\theta$  during the forward pass). We use the size since the aspect ratio is similar to that of a body trunk and it has sufficient capacity to reduce objectness scores. We use the Adam optimizer with a learning rate  $10^{-3}$ , and decay the rate every 100 epochs until 400 is reached. For YOLOv2/v3, images are resized to  $640 \times 640$  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 similar to the top-performing YOLOV2 patch with objects like persons, umbrellas *etc.* (see Fig. 3k); (3) GREY, a grey patch; (4) GREY++, the most powerful

<sup>2</sup> We found that using a lower resolution produced more effective attacks.

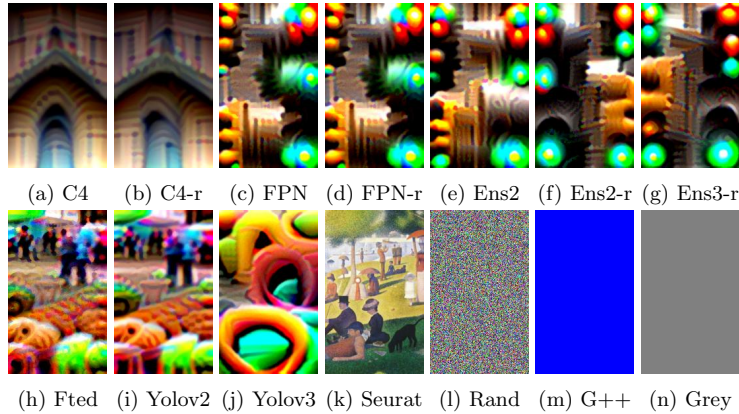


Fig. 3: **Adversarial patches**, and comparisons with control patches. Here, (a)-(d) are based on R50, and G++ denotes Grey++.

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.

Adversarial and control patches are shown in Fig. 3, and results are summarized in Table 1. We 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). 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 (cf. Fig. 3i and Fig. 3k), it does not consistently perform better than GREY. We visualize the impact of the patch in Fig 4, which shows objectness maps from the YOLOv2 model with and without patches. We see that when patches are applied to persons, the corresponding pixels in the feature maps are suppressed.

**Transferability across backbones.** We also investigate whether the learned patches transfer to detectors with different backbones. We evaluate the patches on the following detectors: (1) R101-FPN, which uses ResNet-101 with FPN as its backbone; (2) X101-FPN, replaces the R101-FPN with ResNeXt-101 [35]; (3) R50-FPN-M, a Mask RCNN model [12] based on R50-FPN; (4) X101-FPN-M, a Mask RCNN model based on X101-FPN; (5) RETINANET-R50, a RetinaNet [20] with a backbone of ResNet-50; (6) FCOS, a recent anchor-free framework [32] based on R50-FPN. The results are shown in Fig 5. We observe that all these learned patches can significantly degrade the performance of the person category even using models that they have not been trained on.



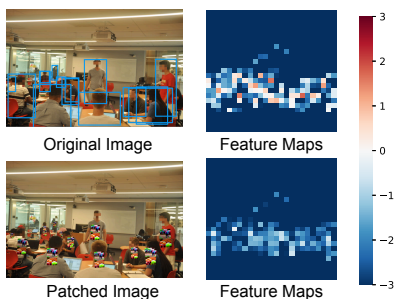


Fig. 4: **Images and their feature maps**, w. and w/o patches, using YOLOv2. Each pixel in the feature map represents an objectness score.

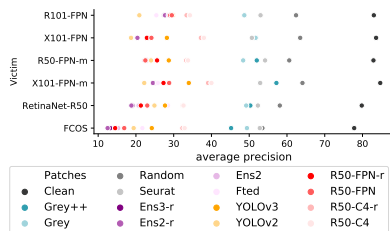


Fig. 5: **Performance of different patches**, when tested on detectors with different backbones.

Patch \ Class	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).

**Transferability across datasets.** We further demonstrate the transferability of patches learned on COCO to other datasets including Pascal VOC 2007 [7] and the Inria person dataset [6]. We evaluate the patches on the person category using R50-FPN and R50-C4, and the results are presented in Fig. 6. The top two panels correspond to results of different patches when evaluated on VOC and Inria, respectively, with both the patches and the models trained on COCO; the bottom panel shows the APs of these patches when applied to Inria images using models trained on VOC. We 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 bottom 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 “bus” and “horse,” and then evaluate these patches on all 20 categories in VOC.<sup>3</sup> Table 2 summarizes the results. We can see that the “person” patch transfers to almost all categories, possibly be-

<sup>3</sup> We observe similar trends on COCO.

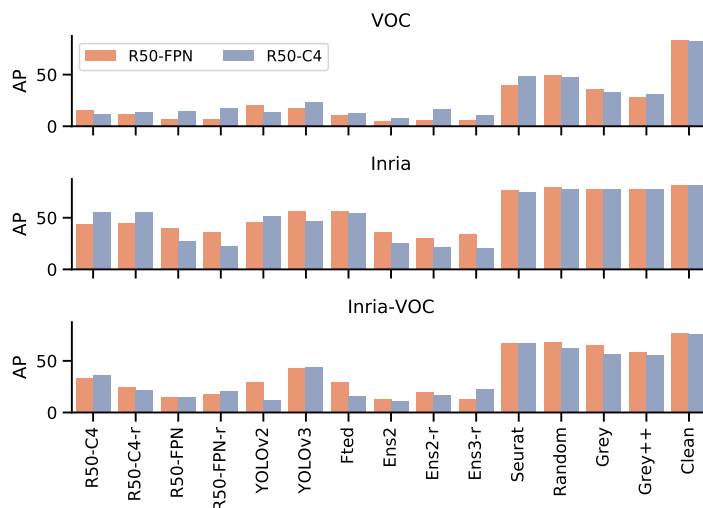


Fig. 6: **Results of different patches, trained on COCO, tested on the person category of different datasets.** Top two panels: COCO patches tested on VOC and Inria, respectively, using backbones learned on COCO; The bottom panel: COCO patches tested on Inria with backbones trained on VOC.

cause 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 discuss the results of physical world attacks with printed posters. In addition to the standard average precision,<sup>4</sup> 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 Fig. 7. To compute these scores, we use cutoff 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

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

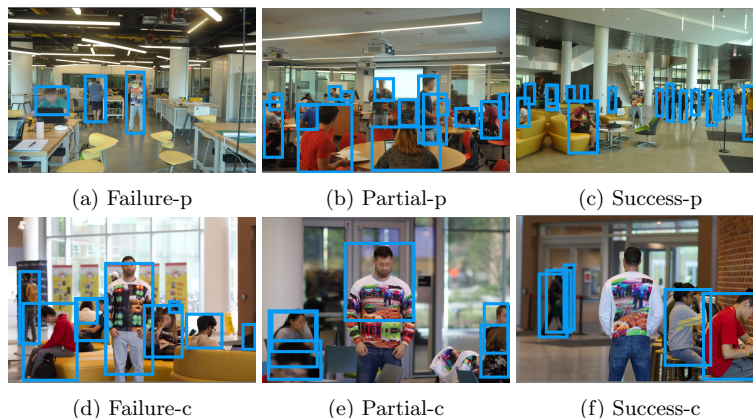


Fig. 7: **Examples of attack failure, partial success, and full success**, using posters (top) and shirts (bottom).

held. We also took photos without printed posters as controls (CONTROL). In total, we collected 630 photos (see the top 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 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 optimized 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 op-

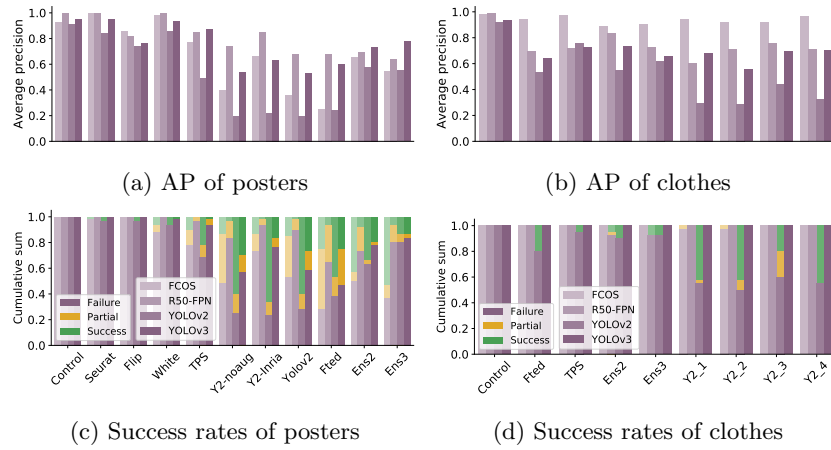


Fig. 8: **AP and success rates for physical attacks.** Top: AP of different printed posters (left) and clothes (right). Lower is better. Bottom: success rates of different printed posters (left) and clothes (right). Y2 denotes YOLOV2.

timization 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 lose 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 useful technique for crafting physical attacks was to make “paper dolls”—printouts of images that we could dress up with different patches at different scales. This facilitates quick experiments with physical world effects and camera distortions without the time and expense of fabricating textiles. 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 assist digital attacks; (3) there exists differences in appearance and texture between large-format digital patches and the digital patches.



Fig. 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 Fig. 9).

We compute success rates of different patches when tested with YOLOv2 and present the results in Fig. 10. Comparing across Fig. 10 and the left side of Fig. 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 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.

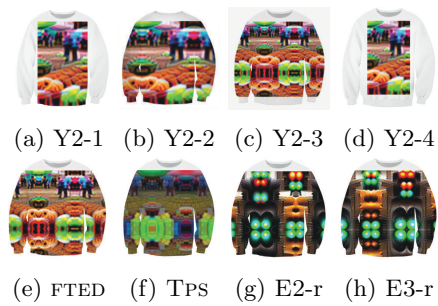
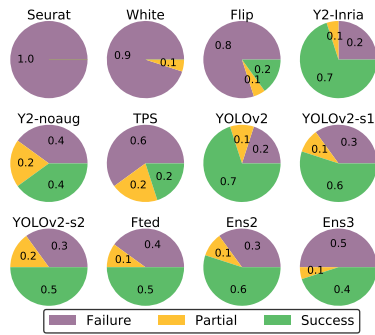


Fig. 10: **Effectiveness of different patches on paper dolls.**

Fig. 11: **Adversarial shirts tested in Section 6.** Y2 denotes YOLOV2.

## 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 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 (see Supple. for a gallery of samples).

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 Fig. 8b and Fig. 8d. 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  $\sim 50\%$  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 attacks with non-rigid TPS transforms did not improve transferability, 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 valuable insights into object detectors, and for helping us improve our experiments. This work is partially supported by Facebook AI.

## References

1. Arnab, A., Miksik, O., Torr, P.H.: On the robustness of semantic segmentation models to adversarial attacks. In: CVPR (2018) [2](#)
2. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. In: ICML (2018) [6](#)
3. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch. arXiv preprint arXiv:1712.09665 (2017) [3](#)
4. Cai, Z., Vasconcelos, N.: Cascade r-cnn: High quality object detection and instance segmentation. arXiv preprint arXiv:1906.09756 (2019) [3](#)
5. Chen, S.T., Cornelius, C., Martin, J., Chau, D.H.P.: Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In: ECML (2018) [3](#)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005) [9](#)
7. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. IJCV (2015) [9](#)
8. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramer, F., Prakash, A., Kohno, T., Song, D.: Physical adversarial examples for object detectors. In: WOOT (2018) [3](#)
9. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust physical-world attacks on deep learning models. In: CVPR (2018) [3](#)
10. Girshick, R.: Fast r-cnn. In: ICCV (2015) [3](#)
11. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015) [2](#)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017) [3](#), [8](#)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) [6](#)
14. Huang, L., Gao, C., Zhou, Y., Xie, C., Yuille, A.L., Zou, C., Liu, N.: Universal physical camouflage attacks on object detectors. In: CVPR (2020) [3](#)
15. Komkov, S., Petiushko, A.: Advhat: Real-world adversarial attack on arcface face id system. arXiv preprint arXiv:1908.08705 (2019) [3](#)
16. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. In: ICLR Workshop (2017) [3](#)
17. Li, Y., Chen, Y., Wang, N., Zhang, Z.: Scale-aware trident networks for object detection. In: ICCV (2019) [3](#)
18. Li, Y., Bian, X., Chang, M.C., Lyu, S.: Exploring the vulnerability of single shot module in object detectors via imperceptible background patches. In: BMVC (2019) [3](#)
19. Li, Y., Tian, D., Chang, M., Bian, X., Lyu, S.: Robust adversarial perturbation on deep proposal-based models. In: BMVC (2018) [3](#)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) [3](#), [8](#)
21. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016) [3](#)
22. Liu, X., Yang, H., Liu, Z., Song, L., Li, H., Chen, Y.: Dpatch: An adversarial patch attack on object detectors. arXiv preprint arXiv:1806.02299 (2018) [3](#)
23. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017) [2](#)

24. Metzen, J.H., Kumar, M.C., Brox, T., Fischer, V.: Universal adversarial perturbations against semantic image segmentation. In: ICCV (2017) 2
25. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: CVPR (2017) 2
26. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: CVPR (2017) 3
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS (2015) 3
28. 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: ACM CCS (2016) 3
29. Singh, B., Najibi, M., Davis, L.S.: Sniper: Efficient multi-scale training. In: NeurIPS (2018) 3
30. Sitawarin, C., Bhagoji, A.N., Mosenia, A., Chiang, M., Mittal, P.: Darts: Deceiving autonomous cars with toxic signs. arXiv preprint arXiv:1802.06430 (2018) 3
31. Thys, S., Van Ranst, W., Goedemé, T.: Fooling automated surveillance cameras: adversarial patches to attack person detection. In: CVPR Workshop (2019) 3
32. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully convolutional one-stage object detection. In: ICCV (2019) 8
33. Wei, X., Liang, S., Chen, N., Cao, X.: Transferable adversarial attacks for image and video object detection. In: IJCAI (2019) 3
34. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: ICCV (2017) 2, 3
35. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017) 8
36. Xu, K., Zhang, G., Liu, S., Fan, Q., Sun, M., Chen, H., Chen, P.Y., Wang, Y., Lin, X.: Adversarial t-shirt! evading person detectors in a physical world. In: ECCV (2020) 3
37. Zeng, X., Liu, C., Wang, Y.S., Qiu, W., Xie, L., Tai, Y.W., Tang, C.K., Yuille, A.L.: Adversarial attacks beyond the image space. In: CVPR (2019) 3