CY 7790, Lecture 2021-10-28 and 2021-11-04 Notes:

Manjit Ullal
Northeastern University

December 14, 2021

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

[3]

presented by: Alina Oprea

1.1 Problem Statement

Object detection is a technology in computer vision of detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. This paper studies transferability of attacks on object detectors. Both black-box and white box attacks are studied under digital and physical domain and the attack success is measured under various conditions. Goal of the attack is to create a patch which when placed digitally or physically on the object in the image makes it invisible to the detector. Here the patch is 1) universal (works for any image); 2) transferable across different models; 3) dataset agnostic; 4) robust to viewing conditions; 5) Work digitally and physically.

1.2 Threat Model

Digital attack is performed under both white-box setting where model parameters are used to learn the patch and black-box setting where a surrogate model is used to learn the patch. There is presumption on the architecture of the model. For example whether it is single step or two step detector.

1.3 Methodology

1.3.1 Background

How does object detector work ?
In Object detection there are two objectives. One is to locate the object in an image. This is done by predicting a box around the object called a bounding box in the image. And other objective is identify the object in the image.
Currently there are models which does this in two stages like Fast RCNN and there are other like YOLO which does it in one stage.

1.3.2 Steps to create patch

As described in the goals, the aim is to create a universal patch which makes object (object here is only applicable to people) in an image invisible to the detector. On each iteration draw random images from images containing people and pass them through the object detector to get the bounding box. Then place a random patch on each detected person. Apply augmentations such as brightness, contrast, rotation and translation to the patch.
Improve the patch by updating it via minimizing the below objective.

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

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

In equation 1, R_θ is a render function that applies the path P to the images I after performing the transformations, post which objective score is found of these images which has positive value if object has been detected in the image and negative if not. And the objective is to minimize the expectation of the loss function L_{obj} over all the priors i.
In order to make the patch smooth such that all the pixels are used a small TV penalty is incorporated in equation 2.

1.4 Results

1.4.1 Digital attacks

Patches in Fig 3 below were learned using the above method were learned on COCO dataset and patches were applied to single step and two step object detectors and the results can be seen in Table 1 below.
We can notice that there is clear trend where patch reduces the average precision of object detectors regardless of whether its single step or multiple step. The patch performs differently across models. Ensemble patch works the best.

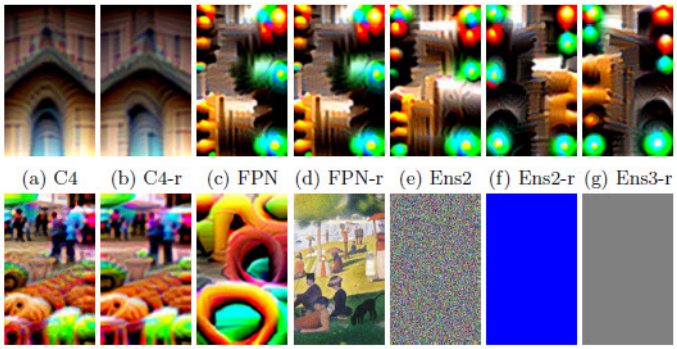


Fig. 3: **Adversarial patches**, and comparisons with control patches. Here, (a)-(d) are based on R50, and G++ denotes Grey++.

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.

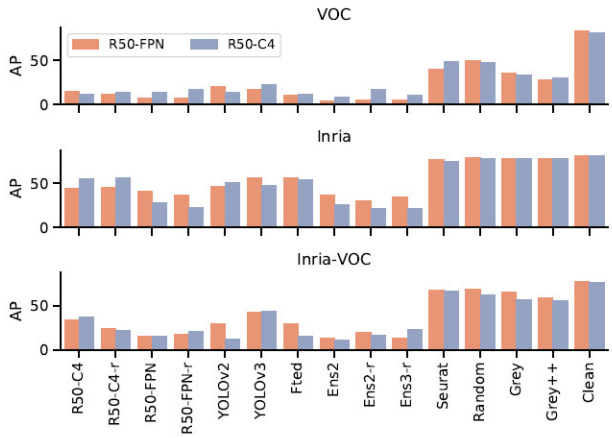
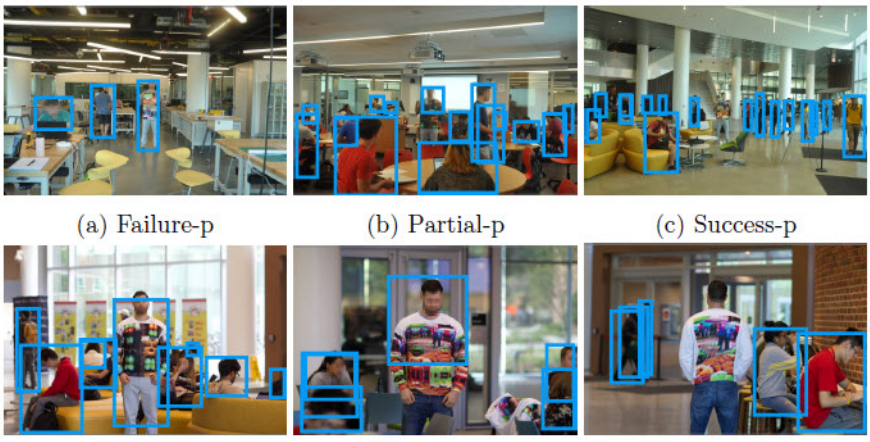


Fig. 6: **Results of different patches, trained on COCO, tested on the person category of different datasets.** Top two panels: COCO patches tested on VOC and Inria, respectively, using backbones learned on COCO; The bottom panel: COCO patches tested on Inria with backbones trained on VOC.

Transferability of attacks
In Fig 6 we can see that patches trained on COCO dataset are tested on Models trained on different dataset and the patches do work.

1.4.2 Physical attacks

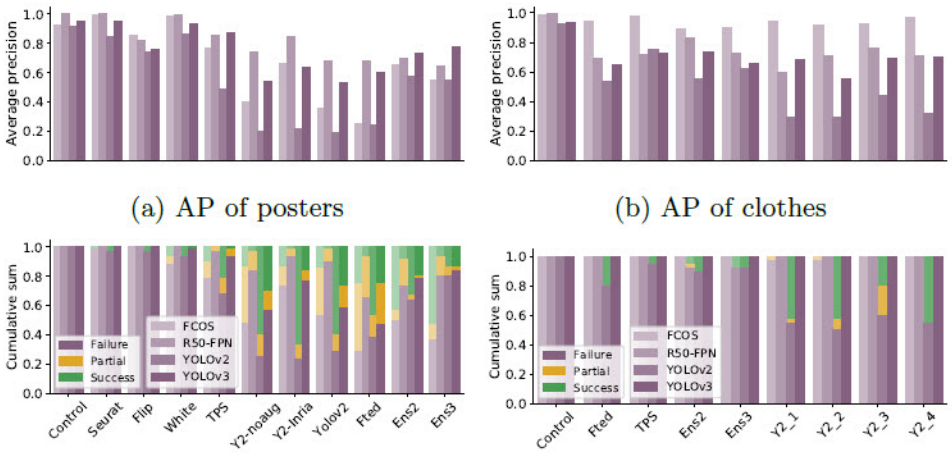


(a) Failure-p (b) Partial-p (c) Success-p

(d) Failure-c (e) Partial-c (f) Success-c

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

Here we can see that the patches work on physical domain as well. A person wearing a T-Shirt with the patch on is invisible to the object detector in Fig 7. Fig 8 shows the histogram of performance of different types of physical attack like wearing a T-Shirt or holding a poster with patch. And we see that the patch does work with varying degree of success across models.



(a) AP of posters (b) AP of clothes

(c) Success rates of posters (d) Success rates of clothes

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.

1.5 Class discussion

What are the labels to the model ? Labels in object detection model is an array/ vector having locations (x,y) of the bounding box and the corresponding label in that bounding box.
Does the patch work for different dataset ? Yes this has been demonstrated in the paper.
Does the patch work for different model ? It has been shown in the paper that the patches do transfer across Models with different backbone.
What are the inputs to the Object detection Model? Images
Does the patch transfer well for other Models? The Patch works across different types of object detectors like YOLO, faster RCNN etc.