Information Sciences 556 (2021) 459–471

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Towards a physical-world adversarial patch for blinding object detection models

Yajie Wang [a], Haoran Lv [a,c], Xiaohui Kuang [b], Gang Zhao [b], Yu-an Tan [a], Quanxin Zhang [a], Jingjing Hu [a,*]

[a] School of Computer Science, Beijing Institute of Technology, Beijing 100081, China
[b] National Key Laboratory of Science and Technology on Information System Security, Beijing 100192, China
[c] Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 22 April 2020
Received in revised form 12 August 2020
Accepted 23 August 2020
Available online 26 October 2020

Keywords:
Adversarial patch
Adversarial attack
Object detection model
Deep neural network

ABSTRACT

As one of the core components of the computer vision, the object detection model plays a vital role in various security-sensitive systems. However, it has been proved that the object detection model is vulnerable to the adversarial attack. In this paper, we propose a novel adversarial patch attack against object detection models. Our attack can make the object of a specific class invisible to object detection models. We design the detection score to measure the detection model's output and generate the adversarial patch by minimizing the detection score. We successfully suppress the model's inference and fool several state-of-the-art object detection models. We triumphantly achieve a minimum recall of 11.02% and a maximum fooling rate of 81.00% and demonstrates the high transferability of adversarial patch between different architecture and datasets. Finally, we successfully fool a real-time object detection system in the physical world, demonstrating the feasibility of transferring the digital adversarial patch to the physical world. Our work illustrates the vulnerability of the object detection model against the adversarial patch attack in both the digital and physical world.

1. Introduction

Deep learning has shown amazing performance in many tasks in the field of computer vision. As one of the essential tasks of computer vision, object detection has been widely used in face recognition, object tracking, robot vision, and other tasks. Especially in safety–critical systems such as autonomous vehicles and video surveillance, object detection models are used for perception, such as identifying pedestrians, traffic signs, and other essential objects. Although deep neural networks have achieved great success in solving visual problems, they are still vulnerable to deception of the adversarial example. The adversarial example means well-designed malicious input with imperceptible noise. The adversary can manipulate the output of the deep learning model with the adversarial example. In recent years, the security and robustness of object detection models have attracted more and more attention. Research on the adversarial example has become an important part of the security field, just like traditional themes such as encryption technology [12,18] and covert access pattern [14,25]. Similar to the adversarial example problem bring by the application of deep learning models [47,40], more and more security problems in new scenarios continue to rise, such as the attack on Android devices [5,39], the attack on wireless sensor networks [46],

* Corresponding author.
E-mail address: hujingjing@bit.edu.cn (J. Hu).

https://doi.org/10.1016/j.ins.2020.08.087
0020-0255/

and the PDF malware detection [19]. It is an increasingly vital task to study the vulnerability of the object detection model when subjected by the adversarial example.

Most research works about the adversarial example focus on the adversarial example in digital space [11,26,4,2], that is, calculating a limited perturbation and making a small amount of modification for each pixel then feed it to the model to attack. Although the perturbation added by these attack strategies is tiny, it requires the attacker to manipulate the entire image at the pixel level, which is completely impossible in the physical world attack. Therefore, Brown et al. propose Adversarial Patch as a practical method for the adversarial attack in the physical world [3]. In the adversarial patch attack, the adversary attacks by only modifying pixels in a confined region. Adversarial Patch attacks the black-box model successfully; black-box means the model with unknown structure and parameters. Recent research has demonstrated that adversarial patches pose a huge threat to computer vision systems in the physical world. Carefully crafted patches can fool high-precision classification models [15] or face recognition models [34]. The adversarial patch attack has become one of the most practical threat models for computer vision models in the physical world.

We propose an efficient adversarial patch attack against a specific target class in the object detection model, called Invisibility Patch. Specifically, we generate an adversarial patch to attack. Our attack adds the patch to a specific object, making the object detection model blind to the object, realizing an attack in the physical world (Fig. 1). Our attack can hide any class of the object. Here we mainly consider "person" as our target class because the misjudgment of the "person" class in safety-sensitive systems such as autonomous vehicles and video surveillance could cause terrible consequences. We measure the output of the object detection model through a well-designed detection score, then generate the adversarial patch through iterative optimization with other loss terms. We design three different detection scores to attack and demonstrate our adversarial patch attack on two kinds of representative object detection models: regression-based YOLOv3 model and proposal-based Faster R-CNN model. Our attack suppresses the object detection model's recognition of the target class successfully. We study the patch attack under different parameter settings and attack models with different structures and multiple datasets successfully, demonstrating the excellent performance and high transferability of our attack. Furthermore, we success attack a real-time detector running on webcam input in the physical world. (Fig. 2).

The patch generated by our attack is only related to the target class. It has a high cross-architecture and cross-dataset transferability and the ability to transfer from the digital world to the physical world, which is very important for the black-box attack in the physical world. The adversary can mislead the object detection model's recognition process through a well-designed patch without knowing any information about the target system. Unlike previous work using print patches, we use portable display devices to attack in the physical world. The adversary can use any common carrier to display the patch to attack, such as iPad, smartphone, laptop, etc. This makes our attack easy to execute and distribute in the physical world, and have higher concealment. Unlike the digital adversarial examples, the perturbation of the adversarial patch is substantial, existing defense technologies designed for small perturbations could fail.

Our key contributions in this paper are:

1. We propose a novel adversarial attack that can make specific objects with a patch escape from the detector. Our approach successfully fools several state-of-the-art object detection models and has high transferability between different architecture and datasets.

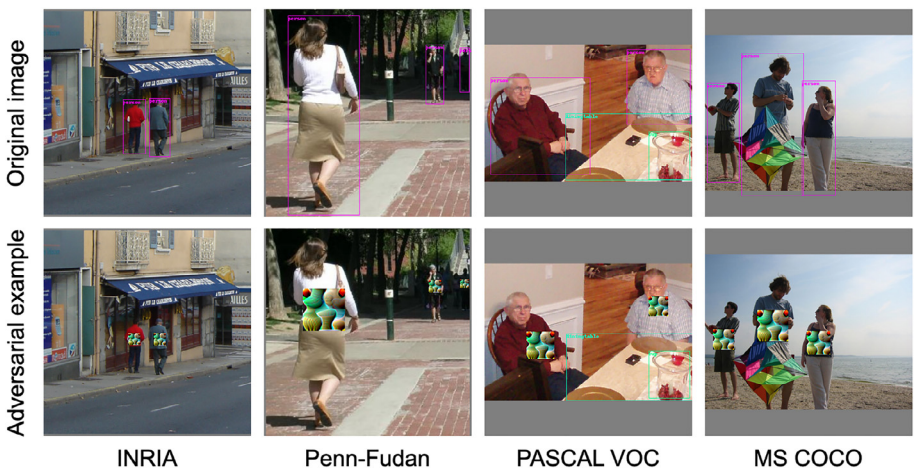

Fig. 1. Detection results of images with the adversarial patch from different datasets. The first line is the detection result of the original image. The second line is the detection result after adding the adversarial patch. The patch attacks the target class "person".

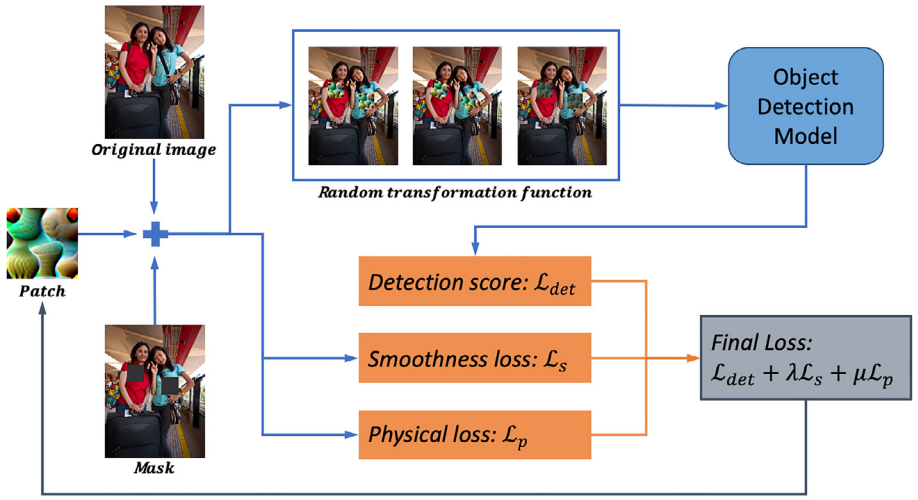

Fig. 2. The framework of the patch generation. We feed the original image to the object detection model and add a patch on the target object. Then we send the modified image to the model and continuously optimize the patch according to the output of the detection model.

2. We propose three different forms of the detection score to quantify the output of the detection model and manipulate the inference process of the detection model for specific objects. We use interpretable metrics to quantify the performance of our attack on different models and datasets and the impact of different settings. We also measure the transferability of our attack.
3. We accomplish the physical-world attack through a patch on the portable display. Our attack makes people escape from the surveillance system's detection, demonstrating the successful transfer of the adversarial patch from the digital world to the physical world.

2. Related work

2.1. Adversarial attack against deep learning model

Szegedy et al. [36] first propose adversarial attack: adding some imperceptible noise to the input sample can lead the deep learning model to a false output with high confidence. The malicious sample with perturbation is called adversarial example. The adversarial example can be generated by various adversarial attacks, such as Fast Gradient Sign Method (FGSM) [11], Projected Gradient Descent (PGD) [26], Carlini–Wagner (C&W) Attack [4], Boundary Attack [2], etc. In addition to the classification model, other kinds of deep learning models can also be affected by the adversarial attack, such as the face detection model [45], semantic segmentation model [43], and video detection model [41]. However, these attacks require the attacker to change all pixels of the input image, which are only feasible in the digital world. It is difficult to realize in the visual system of the physical world.

2.2. Physical-world adversarial attack

Brown et al. [3] propose Adversarial Patch, which is a universal and powerful attack in the physical world. By adding a well-designed patch to the original sample, the classifier can output any target category. Sharif et al. [33] design "adversarial glasses" and successfully mislead the judgment of the face recognition system. They use a concept called Non Printability Score (NPS); NPS can ensure that the adversarial perturbation can be displayed in the physical world. Eykholt et al. [10,35] use images taken from the video as the input to attack the "stop sign" class of object detection model YOLOv2. They generate a patch and prevent stop signs from being correctly recognized with the patch. Athalye et al. [1] use the concept of Expectation over Transformation (EoT) to generate 3D printed real adversarial example objects and deceive deep neural networks. Liu et al. [22] generate spatiotemporal perturbations to form 3D adversarial examples against embodied agents, which could navigate and interact with a dynamic environment. Their perturbation has a strong attack and generalization abilities. Komkov et al. [17] implement an adversarial attack on the face detector through stickers attached to hats. Thys et al. [37] successfully fool a white-box YOLOv2 [30] detection model by printing an adversarial patch. It is the first adversarial attack against the object detection model in the physical world. Liu et al. [23] use print patches to interfere with the classification model's recognition of stop signs in the physical world. Liu et al. [24] propose a bias-based framework to generate class-agnostic universal adversarial patches with strong generalization ability, exploiting both the perceptual and semantic bias of models. Wu et al. [42] print the adversarial patch on T-shirts and posters to attack. However, the existing

adversarial patch attack still has the disadvantages that the attack mode is too fixed, the concealment is weak, and the required patch is too large.

2.3. Object detection model

At present, the object detection models are mainly divided into two major factions: the regression-based one-stage models [30,29,20] represented by YOLOv3 [31] and the two-stage models based on region proposal [6,13] represented by Faster R-CNN [32]. In comparison, the detection accuracy of two-stage models is higher, and the detection speed of one-stage models is faster.

YOLOv3 solves the problem of object detection as a regression problem. After the model processes the input image, we can obtain the label, the confidence, and position of the object in the input. YOLOv3 uses multi-scale features for detection, further improves the detection accuracy, and strengthens the ability to recognize small objects while maintaining the speed advantage. YOLOv3 change from the single-label prediction of the previous generation to multi-label prediction and use Logistic replace Softmax for class prediction. The Faster R-CNN model attaches the Region Proposal Network (RPN) and embeds the proposal inside the network, realizing the convolutional layer's feature sharing. Then, the proposal extracted by RPN is used for further classification and regression prediction. The entire model can complete the end-to-end detection task without executing a specific candidate box searching algorithm, which significantly improves efficiency.

Non-Maximum Suppression (NMS) can filter out redundant detection boxes, one of the critical steps in the object detection process [28]. NMS relies on pre-designed rules to retain only the bounding box with the highest confidence, reducing the subsequent classification costs, and it is widely used in both one-stage and two-stage object detection models.

3. Methodology

3.1. Fooling object detection model with adversarial patch

The attack goal we propose is to generate the adversarial patch of the target class. When the patch is placed on the target object, the object detection model fails to detect the object. The most important is that the generated patch must be robust. The robustness of the patch can be expressed as: the patch is not related to the image, and it can work on all objects of the target class. The patch is independent of the architecture and the dataset so that it can work on the target class in different models and datasets. The patch can work effectively, whether placed digitally or physically. In this paper, we are primarily interested in the optimization problem as defined by

$$\mathop{\mathbb{E}}_{x\sim\mathbb{X}} \min_{p\in\mathbb{P}} \mathcal{S}\{f(C(x,p,m);\theta)\} \tag{1}$$

where the transformation operator C places the adversarial patch p on the input image x with mask m, f is an object detection model with parameter θ, $\mathbb{X}$ is a distribution of input images, $\mathbb{P}$ is a distribution of patches, and $\mathcal{S}$ is a scoring function used to measure the output of the detection model for the target object.

Our attack strategy is to use images containing objects of the target class to generate the adversarial patch. The target class we select is "person" because the complex intraclass variation of "person" makes the attack more complicated. Moreover, attacking the "person" class is more valuable in the physical world. In the training process, we first freeze all the weight of the object detection model and feed the original image to the model to obtain the detection box with the probability of "person" is highest among all the class. Then we calculate the corresponding mask matrix for each detection box to determine the size and position of the patch. Finally, we place a randomly initialized patch in the mask area then use Adam optimizer [16] to continuously optimize and update the pixels of the patch until the patch is sufficiently robust.

Previous research works are carried out on YOLOv2 [37,35]; we prove that adversarial patch generated on YOLOv2 or generated using random noise cannot attack YOLOv3 successfully (Table 1), so we conduct further research. YOLOv3 uses feature maps of different scales for object detection and returns multiple vectors, each vector contains $[x, y, w, h, p_{obj}, p_{cls1}, p_{cls2}, \ldots, p_{clsn}]$, where x and y are the coordinates of the detection box, w and h are the width and height of the detection box, p_{obj} represents the confidence of whether there is an object in the detection box, p_{cls1} to p_{clsn} are the probability of each object class. YOLOv3 enhances the ability to detect small objects and can execute multi-label prediction, so we pick up the detection box where p_{cls_person} is highest from p_{cls1} to p_{clsn} and only attack these detection boxes. Our attack aims at a specific target class, filtering the detection box according to the probability of class can make our attack more accu-

Table 1
The results of attacking YOLOv3 using patches generated by different methods.

Methods	Recall	FR
Clean	100.00%	0.00%
Random noise	91.81%	0.00%
Patch_YOLOv2	89.70%	3.82%
Patch_YOLOv3(Ours)	**11.02%**	**81.00%**

rate and efficient. Non-Maximum Suppression (NMS) is one of the critical steps in the object detection process [28], also one of the reasons why the object detection model is hard to be attacked. The object detection model usually generates many priors that overlap with one object. The NMS algorithm sorts the detection box according to p_{obj}, only keep the detection box with maximum p_{obj} for one object. We notice that the generation of the adversarial patch without considering the NMS algorithm is imperfect through experiments. The reduced p_{obj} of an object causes the object to escape the detection of the object detection model, but NMS could choose another detection box to represent the object. In order to completely hide the target object, our adversarial patch must fool all the priors related to the object.

During the training process, we design a random transformation function C, which acquired the image x, patch p, and mask m to generate more diverse adversarial examples. Random transformation function includes scaling, translation, rotation, brightness, contrast, and noise. The transformation function C simulates the influence of environmental factors in the physical world. For example, we use perspective transformation to simulate angle changes, and random grayscale transformation to simulate lighting changes. This process makes the adversarial patch more robust in the physical world.

3.2. Loss functions

We use $\mathcal{L}_{det}$ to measure the expectation that the object detection model recognizes the target object; we call it detection score. By optimizing the detection score, we can minimize the probability of the target in the output of the detection model. As we mentioned above, we select the detection box with the highest p_{cls_target} to attack. p_{obj_target} means the confidence of the target detection box. The parameter α and β are used to measure the relative importance of each objective. We discuss different forms of detection score and the value of the two parameters in the subsequent experimental part.

$$\mathcal{L}_{det} = \alpha p_{obj\_target} + \beta p_{cls\_target} \tag{2}$$

$\mathcal{L}_s$ forces the patch p to have small total variation (TV) [27,33], and so appear more smooth and natural. The color changes in the image in the physical world are smooth and stable, and adjacent pixels with vast differences cannot be recognized accurately by the camera. Therefore, we use $\mathcal{L}_s$ to smooth the pattern in the patch. The smoother the patch, the smaller the value of $\mathcal{L}_s$. It also improves the recognizability of the adversarial patch in the physical world.

$$\mathcal{L}_s = \sum_{i,j}\left(\left(r_{i,j} - r_{i+1,j}\right)^2 + \left(r_{i,j} - r_{i,j+1}\right)^2\right)^{\frac{1}{2}} \tag{3}$$

$\mathcal{L}_p$ limits the presentable color loss of patch p in the transfer from the digital world to the physical world. As everyone knows, displays are unable to reproduce the color of the original digital image accurately. So in the physical world, the generated patch cannot work correctly in the same way as they were supposed to be. This chromatic aberration introduces difficulties when attacking the physical-world object detection models. Here we use non-printability score (NPS) [33] to reduce it, which defines the color factor contained in digital images that cannot be represented in the physical world. To make the adversarial patch work in the physical world, we hope that the colors in the patch can appear in the physical world as much as possible so that we use $\mathcal{L}_p$ as part of our optimization. i_{patch} is a single pixel in adversarial patch P, and $c_{display}$ is one of the color C that can be displayed in the physical world.

$$\mathcal{L}_p = \sum_{i_{patch} \in P} \min_{c_{display} \in C} \left| i_{patch} - c_{display} \right| \tag{4}$$

Finally, our full objective function can be expressed as:

$$\mathcal{L} = \mathcal{L}_{det} + \lambda \mathcal{L}_s + \mu \mathcal{L}_p \tag{5}$$

where λ and μ are parameters determined empirically and used to measure the relative importance of each objective. The ultimate goal of our optimization is to make the total loss $\mathcal{L}$ as small as possible.

4. Experiments

4.1. Setup

4.1.1. Dataset

INRIA Person Dataset contains a series of images of people standing or walking, collected from images and videos during the research of pedestrian detection by Dalal et al. [7]. The images in the dataset mainly come from GRAZ-01, personal photos, and google, so the clarity of the pictures is high, and it resembles the picture filmed by cameras. There are 614 positive samples in the training set and 288 positive samples in the test set. Penn–Fudan Database for Pedestrian Detection and Segmentation contains 170 images obtained from scenes around campus and city streets, and each image has at least one pedestrian [38]. MS COCO 2017 dataset is a dataset for object detection and segmentation, which includes 91 categories of objects, 328,000 images, and 2,500,000 labels [21]. PASCAL VOC 2007 dataset is a dataset for image recognition and classification,

which includes 20 categories of objects, 9963 images, and 24,640 objects [9,8]. We randomly pick 200 images containing people from MS COCO 2017 dataset and PASCAL VOC 2007 dataset for our experiment.

4.1.2. Target model

We analyze a series of mainstream object detection models and select two representative models YOLOv3 and Faster R-CNN, as our target models. YOLOv3 and Faster R-CNN are widely used as the object detector in autonomous vehicles, video surveillance, and other similar systems. We use the pre-trained PyTorch implementation of the YOLOv3 and Faster R-CNN object detection algorithm from GitHub [31,44]. We only get the output vector of the model without any modification to the model. We also purchase an NVIDIA Jetson Developer Kit with object detector built-in as our target for physical world attack. For this device, we only use a web camera to call the real-time object detection interface.

4.1.3. Evaluation metrics

We use the recall and the fooling rate (FR) to measure the adversarial patch's performance. The recall is the ratio of the number of people detected by the object detection model to the total number of people in the sample. We count the number of people detected in the clean sample as the total number of people. If the object detector alarms when it detects a person, the recall can be regarded as an alarming rate: the lower the alarming rate, the better the attack effect. The fooling rate indicates the probability of a successful attack against the object detection model (same as attack success rate). If the patch hides everyone in the picture, we count it as a successful attack. The higher the fooling rate, the better the attack effect.

4.1.4. Parameter setting

When optimizing our objective function, we use Adam optimizer [16] to train 1000 epochs, and the initial learning rate is set to 0.1. The batch size is set to 8. The size of the adversarial patch is set to 300 * 300 pixels.

4.2. Evaluation

4.2.1. The relative position of patch and detection box

We hope that our patch is in the proper position in the detection box of different scales, so we study the relative position of patch and detection box. We set the position l of the center point of the patch to $l = y - a * h$, where y and h are the center point and the height of the corresponding detection box, a is a coefficient determined empirically. YOLOv3 enhances the ability to detect small objects. If the value of a is too large, the patch exceeds the range of the detection box, and it cannot play an effective attack. We conduct experiments with different a (Fig. 3), and finally determine that 0.125 is a good value of a. To ensure that the patch can affect detection boxes of all sizes, we do not filter the detection frames in this experiment.

4.2.2. The relative size of patch and detection box

The size of the detection box output by the object detection model is diverse, so the size of the patch needs to be adjusted according to the size of the detection box. We set the adversarial patch to a square with side length $s * d$, where d is the length of the diagonal of the corresponding detection box, s is a coefficient determined empirically. Considering the hidden need to perform an attack in the physical world, we want the patch to be as small as possible. However, the reduction of the patch leads to a decrease in the fooling rate of the attack (Fig. 4). After careful consideration, we set s to 0.2.

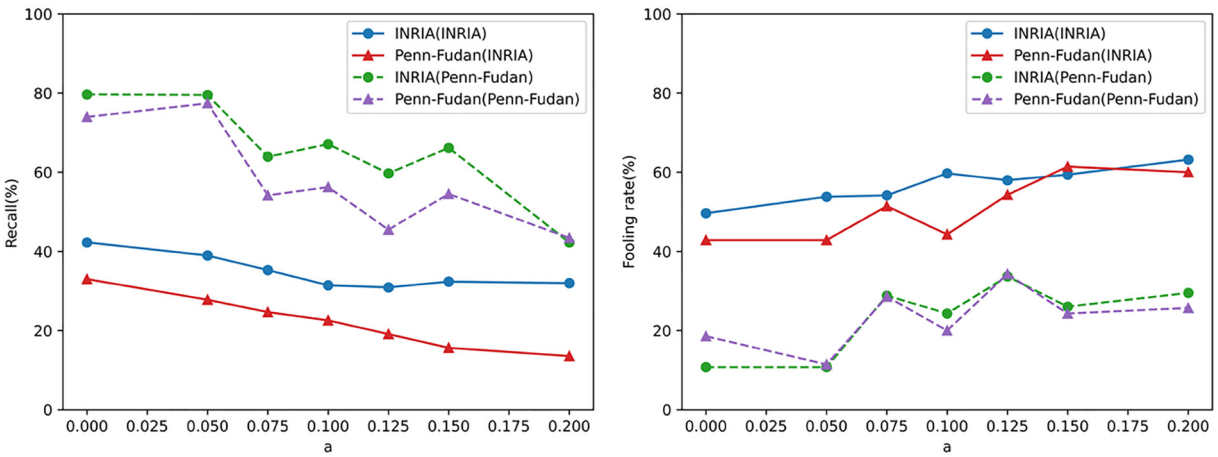

Fig. 3. Recall and fooling rate of the adversarial patch under different values of a. We generate the adversarial patch on two datasets and test their performance. "Penn–Fudan (INRIA)" represents the result of the Penn–Fudan Database for Pedestrian Detection and Segmentation, and the patch is generated on the INRIA Person Dataset.

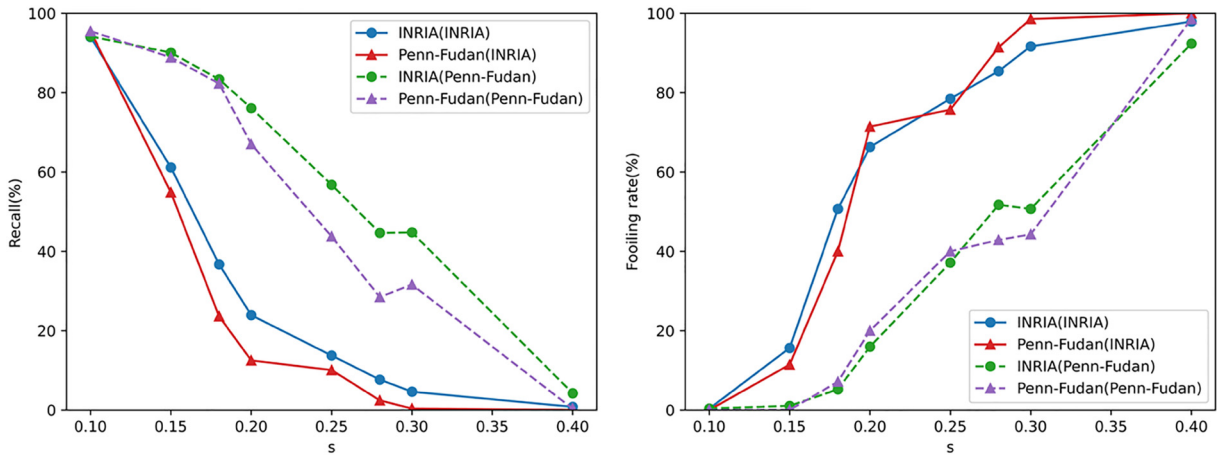

Fig. 4. Recall and fooling rate of the adversarial patch under different values of *s*. We generate the adversarial patch on two datasets and test their performance. "Penn–Fudan (INRIA)" represents the result of the Penn–Fudan Database for Pedestrian Detection and Segmentation, and the patch is generated on the INRIA Person Dataset.

4.2.3. Measuring the output of object detection model

Our goal is to generate an adversarial patch by solving the optimization problem as Eq. (1). When the patch appears on the target object, the object becomes invisible to the detection model. In other words, we hope that the patch can minimize the target object's probability in the output of the detection model, thereby hiding the target object. In the output vector of the object detection model, p_{obj} represents the confidence of whether there is an object in the detection box or not, and p_{cls} represents the probability of each class. The decrease of p_{obj} can make the model think that there is no object, and the decrease of p_{cls} can make the model unable to correctly judge the label of the object. We use a linear combination of the two vectors to measure the probability that the object detection model correctly recognizes the target object, which we called the detection score. In the attack process, we minimize the detection score to reduce the probability of the target object in the model's output so that the target object can evade detection.

In order to attack the object detection model, we design three different forms of the detection score and verify their effect. We first use p_{obj} and p_{cls} of all objects output by the model as the detection score (Table 2, 3). Although we use a dataset containing mainly target class "person" to generate the patch, the effect is not good enough. The reason is that YOLOv3 enhances the ability to detect small objects, and most of the pictures in the dataset come from the real world, resulting in many objects not belonging to our target class being detected, which interferes with the generation of the patch.

For the second form of detection score, we use $p_{obj\_target(max)}$ and p_{cls_target} for optimization (Table 2, 3), trying to optimize the attack by improving the attack candidate list of the detection box. $p_{obj\_target(max)}$ means the object has the highest confidence in the candidate detection box, but this method is not ideal. The reason is still the small objects in the image. In essence, larger objects tend to have higher p_{obj}, resulting in under-utilization of small objects of the target class during the patch generation process. Therefore, the generated patch becomes less effective against small objects; the attack effect reduces.

Table 2
Recall and fooling rate of the adversarial patch generated on INRIA Person Dataset with different detection score. Score 1: $\mathcal{L}_{\text{det}} = \alpha p_{obj} + \beta p_{cls}$. Score 2: $\mathcal{L}_{det} = \alpha p_{obj\_target(max)} + \beta p_{cls\_target}$. Score 3: $\mathcal{L}_{det} = \alpha p_{obj\_target} + \beta p_{cls\_target}$.

		α β	0.8 0.2	0.6 0.4	0.5 0.5	0.4 0.6	0.2 0.8
Score 1	Recall on INRIA		28.93%	30.52%	28.53%	28.53%	27.08%
	Fooling rate on INRIA		64.58%	64.24%	61.46%	62.15%	60.76%
	Recall on Penn–Fudan		17.01%	16.67%	15.63%	18.75%	15.28%
	Fooling rate on Penn–Fudan		61.43%	61.43%	65.71%	60.00%	62.86%
Score 2	Recall on INRIA		28.80%	27.61%	32.23%	28.14%	31.04%
	Fooling rate on INRIA		56.60%	61.81%	58.33%	63.54%	60.76%
	Recall on Penn–Fudan		20.49%	15.28%	28.13%	17.71%	23.96%
	Fooling rate on Penn–Fudan		50.00%	60.00%	42.86%	64.29%	52.86%
Score 3	Recall on INRIA		24.57%	26.68%	**23.51%**	28.01%	28.53%
	Fooling rate on INRIA		65.63%	64.24%	**65.97%**	60.42%	52.78%
	Recall on Penn–Fudan		12.85%	13.89%	**11.46%**	15.97%	23.96%
	Fooling rate on Penn–Fudan		67.14%	62.86%	**71.43%**	57.14%	47.14%

Table 3
Recall and fooling rate of the adversarial patch generated on Penn–Fudan Database for Pedestrian Detection and Segmentation with different detection score. Score 1: $\mathcal{L}_{det} = \alpha p_{obj} + \beta p_{cls}$. Score 2: $\mathcal{L}_{det} = \alpha p_{obj\_target(max)} + \beta p_{cls\_target}$. Score 3: $\mathcal{L}_{det} = \alpha p_{obj\_target} + \beta p_{cls\_target}$.

		α	0.8	0.6	0.5	0.4	0.2
		β	0.2	0.4	0.5	0.6	0.8
Score 1	Recall on INRIA		62.75%	58.52%	59.58%	**53.24%**	91.28%
	Fooling rate on INRIA		29.86%	35.07%	31.60%	**42.01%**	0.35%
	Recall on Penn–Fudan		47.57%	46.53%	48.96%	**37.50%**	91.32%
	Fooling rate on Penn–Fudan		30.00%	24.29%	28.57%	**40.00%**	0.00%
Score 2	Recall on INRIA		78.60%	79.13%	77.54%	81.64%	57.33%
	Fooling rate on INRIA		14.24%	12.85%	15.28%	5.90%	42.71%
	Recall on Penn–Fudan		72.22%	68.06%	72.57%	79.86%	41.67%
	Fooling rate on Penn–Fudan		14.29%	21.43%	14.29%	4.29%	34.29%
Score 3	Recall on INRIA		77.54%	76.09%	79.26%	78.60%	91.41%
	Fooling rate on INRIA		16.67%	14.58%	9.72%	14.24%	0.00%
	Recall on Penn–Fudan		68.40%	69.79%	73.61%	69.10%	91.67%
	Fooling rate on Penn–Fudan		17.14%	12.86%	17.14%	14.29%	0.00%

Finally, we choose p_{obj_target} and p_{cls_target} of the target class as the detection score (Tables 2, 3). It avoids the influence of other objects and successfully achieves a minimum recall of 11.46% and a maximum fooling rate of 71.43%. We conduct experiments with different coefficients and found that the attack effect is best when the values of a and b are both 0.5. At the same time, we notice that the patch generated on Penn–Fudan Database for Pedestrian Detection and Segmentation is not effective enough. We think this is due to the lack of training data (the dataset contains only 170 images). If we want to generate a sufficiently robust patch, we need an appropriate amount of training data.

The goal of the adversarial patch attack is to make specific objects invisible to the object detection model by adding the adversarial patch. We achieve the attack through an adequate design detection score and loss function. In order to measure the performance of our attack more accurately, we use four datasets to conduct experiments on three different detection models and calculate the average of recall and fooling rate. Specifically, we use INRIA Person Dataset to generate the patch on different models to attack the four datasets. The result shows that our attack performs well, regardless of the training set and architecture of the target model (Table 4). Simultaneously, the patch we generate on one model or dataset can successfully attack another model or dataset, proving that our attack has transferability across models and datasets. This will be described in detail in the following sections.

As we mention in the previous, our adversarial patch can attack any kind of object. We use the image of other classes from MS COCO 2017 dataset to attack. We select 1500 images of 5 classes, 300 images of each, 200 images for training, and 100 images for testing. We calculate each class's accuracy under different IoU thresholds(the value of clean samples in parentheses), the recall, and the fooling rate. It can be seen that our attack can efficiently attack any kind of object (Fig. 5), which only requires a small number of samples to generate a patch(about 0.61% of the entire dataset). Under different threshold settings, an adversarial patch can significantly reduce the accuracy of the object detection model (Table 5). Our attack achieves an average recall of 21.33% and an average fooling rate of 77.60%. At the same time, we notice that under different IoU thresholds, the patch attack causes a similar degradation in accuracy of the detection model, indicating that the attack's effect is not related to the target class.

4.2.4. Transferability across models and datasets

We demonstrate the transferability of the adversarial patch between different models (Table 6). We observe that the transfer efficiency of the adversarial patch between different architecture is ideal, with an average 32.90% and the highest transfer rate reaching 41.01%. At the same time, we also notice that the adversarial patch performs not so good when transfer to YOLOv3 pre-trained on the PASCAL VOC 2007 dataset. We think this is because of the different decision boundaries of the model caused by different training sets.

We also demonstrate the transferability of the adversarial patch across datasets (Fig. 6). We use the patch generated on the INRIA Person Dataset to attack other datasets (Fig. 1). It can be seen that the patch generated on INRIA Person Dataset performs a good attack effect on other datasets. The patch generated on the INRIA Person Dataset can also attack the Penn–Fudan Database for Pedestrian Detection and Segmentation dataset, the PASCAL VOC 2007 dataset, and the MS COCO 2017 dataset. The experiments we conducted on different models all got a very low recall and high fooling rate. The cross-dataset

Table 4
Performance of adversarial patch attack against different models. The value of recall and fooling rate is the average on four different datasets.

Model	Recall	FR
YOLOv3(VOC)	**15.47%**	**75.58%**
YOLOv3(COCO)	21.28%	63.35%
Faster R-CNN	24.16%	48.51%

Fig. 5. Detection results of images of different classes with the adversarial patch. The first line is the detection result of the original image. The second line is the detection result after adding the adversarial patch.

Table 5
Performance of adversarial patch attack against different classes.

Class	Recall	FR	AP@(0.5:0.95)	AP@0.5	AP@0.75
Bus	29.53%	70.00%	26.2%(63.6%)	50.5%(91.6%)	23.3%(71.2%)
Boat	30.46%	69.00%	10.5%(24.4%)	32.7%(58.1%)	**2.7%(17.5%)**
Cat	21.62%	76.00%	23.0%(66.4%)	49.5%(98.9%)	13.4%(74.9%)
Dog	16.52%	84.00%	**10.3%(67.6%)**	**23.0%(98.1%)**	5.2%(79.8%)
Bear	**8.51%**	**89.00%**	19.2%(73.3%)	35.1%(98.1%)	19.5%(86.9%)
Average	21.33%	77.60%	17.8%(59.1%)	38.2%(89.0%)	12.8%(66.1%)

Table 6
Transferability of the adversarial patch between different models.

		Target models		
		YOLOv3(COCO)	YOLOv3(VOC)	Faster R-CNN
Source models	YOLOv3(COCO)	–	12.41%	**41.01%**
	YOLOv3(VOC)	28.74%	–	32.28%
	Faster R-CNN	39.04%	19.26%	–

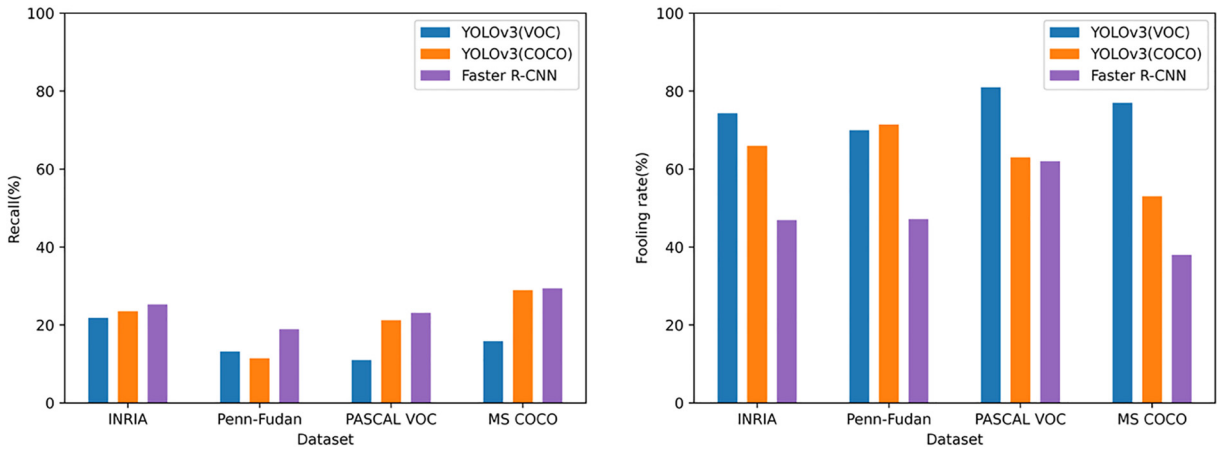

Fig. 6. Recall and fooling rate of adversarial patch attack on different datasets.

attack achieves a minimum recall of 11.02% and a maximum fooling rate of 81.00%. Even the attack performance on the PASCAL VOC 2007 dataset is better than the attack performance on the INRIA Person Dataset, which is used to generate the patch.

These results mean that the patch generated by our attack is independent of the model's architecture, the target image, or the data used to generate the patch. High transferability indicates the patch can interfere with the object detection model's recognition of the target's essential features. So the adversary can operate a complete black-box attack with the adversarial patch. Once the patch is generated, it can be used to attack any model or any image. This is very beneficial for us to carry out attacks in the physical world.

4.2.5. Adversarial patch in physical world

The ultimate goal of the adversarial patch is to achieve attacks in the physical world. We simulate the monitoring system in the real world and successfully carried out a black-box attack. We purchase an NVIDIA Jetson with a built-in object detector and call the detection interface through a webcam. When the detector discovers a person, it alarms. The adversary uses a portable display to show the patch, trying to evade detection. The original patch is a square with 300 pixels on each side, and we enlarge it to about 20 cm. It can be seen that the attacker successfully evades the detection by placing the patch on his chest, and another object behind him is not affected (Fig. 7). This proves that our attack is valuable in the physical world, and only interferes with the detector's recognition of objects with the patch.

Unlike detection models in the digital world, the relative position between the detector and the object in the physical world is constantly changing due to the relative motion of the two objects, such as pedestrians walking in front of the camera of the surveillance system. This displacement causes dynamic changes in the angle between the detector and the object, as well as changes in the size and background of the object. In the digital world, we have proved the robustness of the adversarial patch against objects of different sizes and backgrounds in several datasets, so our experiments mainly focus on the change of angle. We pick five representative angles ranging from -45° to $+45^{\circ}$ and use a patch to attack the surveillance system. We notice that the patch's attack ability does not decrease due to the change of angle; it can successfully attack at all angles. Also, the detector can successfully detect people at all angles when the patch is not used (Fig. 8). Therefore, we can declare that the adversarial patch can successfully attack surveillance systems in real scenarios. These results show that it is entirely feasible to make an object with the adversarial patch invisible to an object detector for attacking detectors in the physical world, which poses a massive threat to all detector-based security-sensitive systems.

4.2.6. Robustness of adversarial patch

In our threat model, the adversarial patch is used by the adversary to blind the surveillance camera. As we all know, whether a person walks from left to right or from right to left does not affect the recognition of the surveillance system to it. Based on these two considerations, we horizontally flip the adversarial example after adding the patch, and we notice that the attack performance of the flipped samples decreases (Table 7). Recall increases by an average of about 10%, and the fooling rate decreases by more than 10%. This shows that the adversarial patch attack is not robust enough to horizontal flip, and we think it is a potential way of defending against such attacks.

4.2.7. Ablation study

In order to verify the effectiveness of the adversarial patch attack we propose, we perform ablation study. We examine an attack using only the detection score, two attacks using only a part of the detection score, and a random noise attack. We calculate the average recall and fooling rate on four different datasets. There is a big gap between the results of other experiments and our method. Experimental results show that the attack performance drops significantly after removing some components, proving the effectiveness of our attack framework. The specific results are listed in (Table 8).

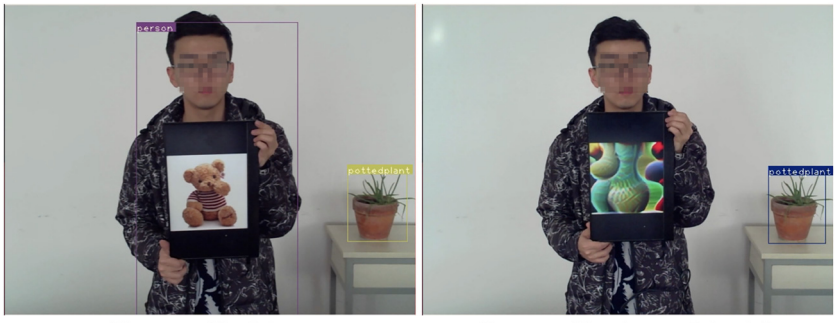

Fig. 7. Attacking surveillance system in the physical world with adversarial patch.

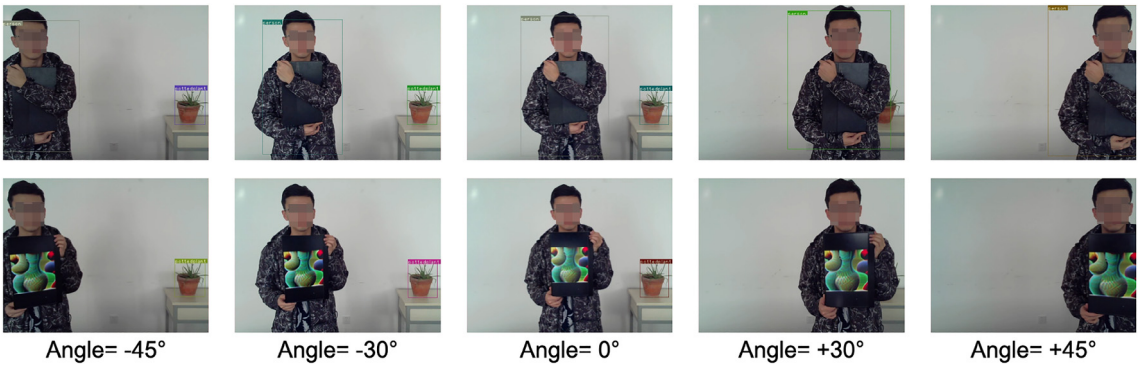

Fig. 8. Attacking surveillance system in the physical world with adversarial patch at different angles.

Table 7
Performance of adversarial patch attack with or without flip.

		Without flip		With flip		Degradation	
Model	Dataset	Recall	Fooling rate	Recall	Fooling rate	ΔRecall	ΔFooling rate
YOLOv3(a)	INRIA	23.51%	65.97%	38.44%	47.22%	**14.93%**	**18.75%**
	Penn–Fudan	11.46%	71.43%	18.40%	57.14%	6.94%	14.29%
	PASCAL VOC	21.22%	63.00%	30.94%	51.00%	9.72%	12.00%
	MS COCO	28.91%	53.00%	41.00%	35.00%	12.09%	18.00%
YOLOv3(b)	INRIA	21.84%	74.31%	34.81%	58.68%	12.97%	15.63%
	Penn–Fudan	13.17%	70.00%	20.16%	60.00%	6.99%	10.00%
	PASCAL VOC	11.02%	81.00%	17.55%	67.00%	6.53%	14.00%
	MS COCO	15.85%	77.00%	29.81%	53.00%	**13.96%**	**24.00%**

Table 8
Ablation study

	Average recall	Average FR
$\mathcal{L} = \mathcal{L}_{det} + \mathcal{L}_s + \mathcal{L}_p$(Ours)	**21.28%**	**63.35%**
$\mathcal{L} = \mathcal{L}_{det}$	49.21%	28.31%
$\mathcal{L} = p_{obj\_target} + \mathcal{L}_s + \mathcal{L}_p$	30.66%	47.60%
$\mathcal{L} = p_{cls\_target} + \mathcal{L}_s + \mathcal{L}_p$	84.76%	1.10%
Random noise	90.04%	0.50%

5. Conclusion

In this work, we successfully attack YOLOv3 and Faster R-CNN, two of the most advanced object detection models. Our adversarial attack is performed by adding an adversarial patch to the original image. We propose an adversarial attack framework that can generate an adversarial patch for the specific class of objects. We suppress the detection model's inference of the target object by minimizing the well-designed detection score. Our attack can successfully hide the target object class "person" or any class in the object detection model. In the experiment, we compare the adversarial patch's performance with different detection scores and parameters, verify the effectiveness of the attack against different detection models and datasets. We demonstrate the high transferability of adversarial patch between different architecture and datasets. Besides, we study the robustness of the adversarial patch attack and potential defense. We also demonstrate that our adversarial patch can fool real-time object detectors in the physical world, which shows the feasibility of transferring digital adversarial patch to the physical world. Our work also proves the vulnerability of object detectors based on deep neural networks against the adversarial patch attack.

6. Future work

The constant movement of objects in the physical world can cause continuous changes in environmental factors such as position, lighting, and occlusion, which affect our attack. In future work, we want to develop more robust adversarial attacks in a complex environment.

CRediT authorship contribution statement

Yajie Wang: Methodology, Writing - review & editing. **Haoran Lv:** Software, Validation. **Xiaohui Kuang:** Formal analysis, Conceptualization. **Yu-an Tan:** Investigation. **Quanxin Zhang:** Funding acquisition, Data curation. **Jingjing Hu:** Resources, Project administration, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 61876019 and No. 61772070).

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, Kevin Kwok. Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397, 2017..
[2] Wieland Brendel, Jonas Rauber, Matthias Bethge, Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint arXiv:1712.04248, 2017..
[3] T.B. Brown, D. Mané, A. Roy, M. Abadi, J. Gilmer, Adversarial Patch (2017), arXiv preprint arXiv:1712.09665.
[4] Nicholas Carlini, David Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 39–57.
[5] Xiao Chen, Chaoran Li, Derui Wang, Sheng Wen, Jun Zhang, Surya Nepal, Yang Xiang, Kui Ren, Android hiv: A study of repackaging malware for evading machine-learning detection, IEEE Transactions on Information Forensics and Security 15 (2019) 987–1001.
[6] Jifeng Dai, Yi Li, Kaiming He, Jian Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Advances in Neural Information Processing Systems, 2016, pp. 379–387..
[7] Navneet Dalal, Bill Triggs, Histograms of oriented gradients for human detection, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, IEEE, 2005, pp. 886–893.
[8] S.M. Mark Everingham, Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, Andrew Zisserman, The pascal visual object classes challenge: A retrospective, International Journal of Computer Vision 111 (1) (2015) 98–136.
[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, Andrew Zisserman, The pascal visual object classes (voc) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338.
[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, Dawn Song, Robust physical-world attacks on deep learning visual classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1625–1634..
[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014..
[12] Zhitao Guan, Xueyan Liu, Wu. Longfei, Wu. Jun, Xu. Ruzhi, Jinhu Zhang, Yuanzhang Li, Cross-lingual multi-keyword rank search with semantic extension over encrypted data, Information Sciences 514 (2020) 523–540.
[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[14] Yanyu Huang, Bo Li, Zheli Liu, Jin Li, Siu-Ming Yiu, Thar Baker, Brij B. Gupta, Thinoram: Towards practical oblivious data access in fog computing environment, IEEE Transactions on Services Computing, 2019..
[15] Danny Karmon, Daniel Zoran, Yoav Goldberg, Lavan: Localized and visible adversarial noise. arXiv preprint arXiv:1801.02608, 2018..
[16] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014..
[17] Stepan Komkov, Aleksandr Petiushko, Advhat: Real-world adversarial attack on arcface face id system. arXiv preprint arXiv:1908.08705, 2019..
[18] Jin Li, Yanyu Huang, Yu Wei, Siyi Lv, Zheli Liu, Changyu Dong, Wenjing Lou, Searchable symmetric encryption with forward search privacy, IEEE Transactions on Dependable and Secure Computing, 2019..
[19] Yuanzhang Li, Yaxiao Wang, Ye Wang, Lishan Ke, Yu-an Tan, A feature-vector generative adversarial network for evading pdf malware classifiers, Information Sciences, 2020..
[20] Zuoxin Li, Fuqiang Zhou, Fssd: feature fusion single shot multibox detector, arXiv preprint arXiv:1712.00960, 2017..
[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, C. Lawrence Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755..
[22] Aishan Liu, Tairan Huang, Xianglong Liu, Yitao Xu, Yuqing Ma, Xinyun Chen, Stephen J Maybank, Dacheng Tao, Adversarial attacks for embodied agents. arXiv preprint arXiv:2005.09161, 2020..
[23] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, Dacheng Tao, Perceptual-sensitive gan for generating adversarial patches, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 1028–1035..
[24] Aishan Liu, Jiakai Wang, Xianglong Liu, Chongzhi Zhang, Bowen Cao, Hang Yu, Patch attack for automatic check-out. arXiv preprint arXiv:2005.09257, 2020..
[25] Zheli Liu, Bo Li, Yanyu Huang, Jin Li, Yang Xiang, Witold Pedrycz, Newmcos: Towards a practical multi-cloud oblivious storage scheme, IEEE Transactions on Knowledge and Data Engineering, 2019..
[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu, Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017..
[27] Aravindh Mahendran, Andrea Vedaldi, Understanding deep image representations by inverting them, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5188–5196.
[28] Alexander Neubeck, Luc Van Gool, Efficient non-maximum suppression, in: 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, IEEE, 2006, pp. 850–855..
[29] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[30] Joseph Redmon, Ali Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
[31] Joseph Redmon, Ali Farhadi, Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018..
[32] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, 2015, pp. 91–99..

[33] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter, Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 1528–1540.
[34] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter, Adversarial generative nets: Neural network attacks on state-of-the-art face recognition. arXiv preprint arXiv:1801.00349, 2(3), 2017..
[35] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, Physical adversarial examples for object detectors. In 12th {USENIX} Workshop on Offensive Technologies {WOOT} 18), 2018..
[36] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus, Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013..
[37] Simen Thys, Wiebe Van Ranst, Toon Goedemé, Fooling automated surveillance cameras: adversarial patches to attack person detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.
[38] Liming Wang, Jianbo Shi, Gang Song, I-Fan Shen, Object detection combining recognition and segmentation, in: Asian Conference on Computer Vision, Springer, 2007, pp. 189–199.
[39] Wenjie Wang, Donghai Tian, Weizhi Meng, Xiaoqi Jia, Runze Zhao, Rui Ma, Msym: A multichannel communication system for android devices, Computer Networks 168 (2020) 107024.
[40] Yajie Wang, Yu-an Tan, Wenjiao Zhang, Yuhang Zhao, Xiaohui Kuang, An adversarial attack on dnn-based black-box object detectors, Journal of Network and Computer Applications, 2020, p. 102634..
[41] Xingxing Wei, Siyuan Liang, Ning Chen, Xiaochun Cao, Transferable adversarial attacks for image and video object detection. arXiv preprint arXiv:1811.12641, 2018..
[42] Zuxuan Wu, Ser-Nam Lim, Larry Davis, Tom Goldstein, Making an invisibility cloak: Real world adversarial attacks on object detectors. arXiv preprint arXiv:1910.14667, 2019..
[43] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Linxi Xie, Alan Yulie, Adversarial examples for semantic segmentation and object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1369–1378.
[44] Jianwei Yang, Jiasen Lu, Dhruv Batra, Devi Parikh, A faster pytorch implementation of faster r-cnn. https://github.com/jwyang/faster-rcnn.pytorch, 2017..
[45] Siqi Yang, Arnold Wiliem, Shaokang Chen, Brian C. Lovell, Using lip to gloss over faces in single-stage face detection networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 640–656.
[46] Yali Yuan, Liuwei Huo, Zhixiao Wang, Dieter Hogrefe, Secure apit localization scheme against sybil attacks in distributed wireless sensor networks, IEEE Access 6 (2018) 27629–27636.
[47] Quanxin Zhang, Yuhang Zhao, Yajie Wang, Thar Baker, Jian Zhang, Jingjing Hu, Towards cross-task universal perturbation against black-box object detectors in autonomous driving, Computer Networks, 2020, p. 107388..