JEI Journal of Electronic Imaging

RESEARCH PAPER

Location-independent adversarial patch generation for object detection

Zhiyi Ding,[a,*] **Lei Sun**,[a] **Xiuqing Mao**,[a] **Leyu Dai**,[a] **and Bayi Xu**[b]

[a]Information Engineering University, School of Cryptography Engineering, Zhengzhou, China
[b]Zhengzhou University, School of Cyber Science and Engineering, Zhengzhou, China

ABSTRACT. Object detection models are at the core of various computer vision tasks and have shown excellent performance on public datasets, but they also inherit the disadvantage of neural networks that they are vulnerable to adversarial example attacks. Adversarial patches are specific forms of adversarial examples that, as shown in previous studies, can only make specific objects (such as pedestrians and traffic signs), but not all objects, disappear. In addition, a patch must be placed on every object to deceive the detector. To solve the above problems, we propose a location-independent adversarial patch generation method that can attack objects in the range to be detected with a single patch. By attacking the confidence loss of the object detector, we creatively assign a greater weight to the foreground region, which makes its confidence decrease faster and effectively guides the convergence direction of the adversarial patch in the training process. Furthermore, we glue the patches randomly on the images to make them less sensitive to location during patch training. Experimental results indicate that the patches generated using our proposed method are not restricted to specific areas of the image and provide a minimum recall of 29.5%.

Keywords: deep learning; security threats; object detection; adversarial patch

Paper 230536G received May 4, 2023; revised Aug. 12, 2023; accepted Aug. 15, 2023; published Aug. 26, 2023.

1 Introduction

In recent years, deep neural networks have achieved excellent performance on a wide range of computer vision tasks, and in some cases, even surpassing human performance.[1] Attributed to the powerful feature extraction capability of deep neural networks, computer vision techniques, such as image classification,[2] object detection,[3] and face recognition,[4] have advanced rapidly. Emerging technologies, such as autonomous driving[5] and robot control,[6] are increasingly becoming mature.

However, this thriving landscape is overshadowed by the emergence of adversarial examples. The existence of adversarial examples was first identified by Szegedy et al.[7] when they attached some specially designed and imperceptible slight perturbations to test images and input them into a DNN-based image classification system, which yielded incorrect outputs. As more studies are conducted on adversarial examples, attack algorithms against other models or tasks have subsequently emerged, such as attacks on video security systems through identity forgery, malicious control attacks on speech, and text detection systems, and even more high-risk attacks on autonomous driving systems. Adversarial attacks on object detection models can have significant consequences. For instance, successful attacks can lead to the misclassification of objects or the detection of non-existent objects. In some cases, such consequences could result in serious

*Address all correspondence to Zhiyi Ding, shakespereding@163.com

security risks, such as autonomous vehicles misidentifying objects on the road or facial recognition software incorrectly identifying individuals. On May 7, 2016, a Tesla Model S was driving in autopilot mode on a Florida highway in the United States when it failed to slow down and crashed into a white tractor-trailer truck in front of it, resulting in a fatal crash, marking the first fatal case of a self-driving car to come to light in the world. It is widely believed that at the time of the accident, the Tesla confused the white truck body that was hit with the sky due to its strong reflection, thus not detecting the presence of the obstacle. Consequently, the vulnerability of models has become a key concern in AI security.[8]

Object detection is much more complicated than the classification task because it needs to draw bounding boxes with an appropriate size to locate the targets in addition to classifying them. In this paper, a method is proposed to perform to adversarial patch attacks against object detection models. As a special form of the adversarial example,[9] an adversarial patch is a sticker-like pattern occupying a small portion of the image, and the attack is no longer limited to imperceptible variations. The patch can be placed on the tested image and successfully trick the detector from recognizing the object properly. The paper mainly aims to generate an adversarial patch with a strong attack capability.

Thys et al.[10] from KU Leuven, Belgium, have found that pedestrian detection systems could be completely deceived with a simple print. These researchers aimed to decrease the object score and class score at the output of the detector, and they were successful in attacking the pedestrian detector based on the YOLO-V2 model[11] by back-propagation training to generate an adversarial patch. However, the researchers still followed the most common way of suppressing detection scores in their approach to adversarial patch generation. Their experimental results, even showing significant attack capabilities, can be further improved. This paper addresses some of the limitations of the Thys team's work, like its single attack category and inability to attack other targets in the image.[10] In addition, their patch needs to be placed on the tested object to attack, and the attack capability is greatly reduced for objects without adhesive patches, which are more sensitive to locations.

Moreover, with the growing use of object detection models in various applications, the impact of adversarial attacks becomes more significant. In summary, the study of adversarial attacks can explore the vulnerability of object detectors and reveal the susceptibility of deep learning models. This, in turn, can help to identify the root cause of system confusion, misjudgment, and omission of the attacked models, and investigate how to improve the robustness of deep neural networks by studying their attack principles and details.

This paper proposes a position-independent adversarial patch generation method to deceive object detectors based on the YOLO-V2 pretrained model on the COCO dataset. Unlike existing methods, this method allows the patch to be placed anywhere in the image, making it more versatile in attacking the model.

The main contributions of this paper are summarized as follows.

1. This paper proposes a method of adversarial patch attacks that does not require attaching patches to each target but rather uses a single patch to make multiple types of objects in the image disappear from the object detector. This approach contrasts with the work of Thys et al., which only demonstrated efficacy against pedestrians.[10]
2. The design of the adversarial patch in the training process ensures robustness to location by adopting a random position generation. The generated patch is not restricted to a specific location, thereby avoiding interference from patch location. This approach allows the patch to be placed in any part of the image to be attacked without necessarily overlapping the object.
3. To blind the object detector, this method uses the object confidence score of the output. The loss function for optimization has different focuses to balance the contribution of the foreground and background. For obj-confidence >0.5, the foreground region is given a greater weight to make its confidence drop faster, whereas a smaller weight is given to the background region with obj-confidence <0.5 to optimize the gradient update direction.

2 Related Work

The work related to object detection and adversarial example is mainly reviewed.

2.1 Object Detection Models

Object detection based on deep learning is a fundamental research topic in computer vision and serves as a basis for advanced tasks, such as instance segmentation, object tracking, and image description.[12] Depending on whether the candidate regions are generated first, the current mainstream detectors are mainly divided into two categories: two-stage detectors, such as Mask R-CNN[13] and Faster R-CNN,[14] and one-stage detectors, such as SSD[15] and YOLO.[16] The two-stage detector requires the CNN neural network to extract image features and generate candidate regions that may contain objects. Then it further adjusts the position coordinates and classifications of objects in the candidate regions to achieve higher accuracy. However, it operates at a lower speed. The one-stage detector, on the other hand, runs faster as it skips the candidate region generation step and uses only an end-to-end network to predict the class and location of objects.

Numerous object detection models have been developed to address various problems, but they have presented a plethora of challenges. While the current models' performance has been continuously improving, their rising complexity and increased number of parameters have made them unsuitable for industrial applications. To mitigate this issue, the knowledge distillation (KD) technique was introduced in 2015 and has been widely adopted in computer vision, particularly in image classification tasks. Over time, the application of KD has been extended to other vision tasks, including target detection. KD leverages complex teacher models to transfer knowledge learned from large-scale or multimodal data to lightweight student models, resulting in improved model compression and performance.[17] Additionally, the traditional detection performance of these methods relies solely on the discriminative capabilities of region features, which often depend on sufficient training data. Even with well-annotated data, we still face the issue of data scarcity as novel categories (e.g., rare animals) continuously emerge in practical scenarios. These aforementioned challenges have led us to investigate the detection task with an additional source of complexity, zero-shot object detection (ZSD). Yan et al.[18] developed a semantics-guided contrastive network specifically designed for ZSD. To the best of our knowledge, this is the first work that applies a contrastive learning mechanism for ZSD.

This paper aims to attack the widely used YOLO object detection models, which is highly preferred in high real-time and complex scenarios.

2.2 Adversarial Example for Image Classification

In the field of image classification, Szegedy et al.[7] were the first to discover that by making slight perturbations to interfere with the input samples, a deep neural network-based image recognition system can be deceived to output arbitrarily wrong results desired by the attacker, and the samples in this case, are called adversarial examples. Goodfellow et al.[19] proposed an algorithm called the fast gradient sign method to generate adversarial examples. This has become one of the most fundamental white-box methods for generating adversarial example for various task-oriented attack problems. Other classical attack methods in the field of image classification include PGD,[20] DeepFool,[21] universal adversarial perturbation,[22] and Carlini and Wagner attacks.[23]

2.3 Adversarial Example for Object Detection

Classical methods for object detection adversarial example generation typically iteratively optimize the loss function. The adversarial examples are continuously adjusted and updated in the gradient backpropagation process until the maximum number of iterations is satisfied or the model prediction reaches the expected value. Lu et al.[24] took the Faster-RCNN detector as an attack model to deceive the detector by minimizing the average prediction score of the "stop" flag and adding perturbations to the "stop" flag. This work is the first paper to propose adversarial example generation in the field of object detection. Xie et al.[25] proposed the Dense Adversary Generation (DAG) attack method for the object detection model and the semantic segmentation model. The method sets a non-correct label for the target and then iteratively moves toward the direction with low-class confidence, eventually making the detector misclassify all regions of interest (ROIs) of the input image. Additionally, Li et al.[26] proposed the Robust Adversarial Perturbation attack algorithm, which combines classification and regression tasks to design new loss functions that focus on destroying the region-proposal network specific to the two-stage model to attack the detector. Wei et al.[27] addressed the issues of weak transferability and high

time consumption of attack methods using a generative adversarial network (GAN) approach to learn adversarial examples for image and video object detection. This method is called Unified and Efficient Adversary, but it is more difficult to train, and its white-box attack is not improved compared to DAG.

2.4 Adversarial Patch for Object Detection

Different from the adversarial example, the adversarial patch is a local perturbation that is not limited by the perturbation paradigm and no longer aims for invisibility. So far, outstanding results have been achieved in research adversarial patches. For example, Google's Brown team first designed a universal and robust adversarial patch in the field of image classification, which can make the classifier output any target class after the patch is applied to the image. Later, the work on adversarial patching was further extended to the field of object detection, and the Ekyholt research team[28] made a series of improvements to the robust physical perturbatiaon algorithm[29] for classifiers by introducing the disappearance attack loss algorithm. They trained the algorithm to generate small, inconspicuous stickers and applied them to traffic signs, causing the object detection system to fail to recognize the stop traffic sign. Chen et al.[30] proposed the ShapeShifter method for Faster R-CNN, which uses EOT transform to generate adversarial perturbations and adds the perturbations to other regions on the traffic signs other than text, resulting the same attack results. Thys et al. generated adversarial patches by reducing the values of object confidence and class probability in the pedestrian detection box, and iteratively optimized the objective function with a backpropagation algorithm.[10] The objective function also includes a non-printability score, which ensures that the colors used in the image can be represented by the printer. If a person wears the adversarial patch, they can disappear from the detector. Lee et al.[31] generated a special adversarial patch by improving the work of DPatch.[32] The loss function of the model output was directly maximized as the optimization target. And the patch pixel values were cropped to allow printing, so that the object detector can be successfully deceived in the real physical world. The Adversarial T-shirt stealth t-shirt researched by the MIT-IBM Watson AI research institute[33] can be attached to a person to achieve the invisibility of the person to the object detection model. The above-mentioned methods' core idea is to reduce loss function score of clean samples as the optimization objective, train the patches through backpropagation, and add the generated patches to the target to deceive the detector.

3 Methodology

3.1 Overall Structure

The paper aims to create a position-independent, universal adversarial patch that can deceive the object detector when placed anywhere in the image. As mentioned previously, Ekyholt et al. and Chen et al. showed that it is possible to perform an adversarial patch attack on the object detector. However, these previous works targeted single types of objects, such as stop signs and pedestrians. In contrast, the approach presented in this paper focuses on all targets in the image and aims to create more challenging adversarial patches. In this paper, patches are trained using the Inria dataset, which is dominated by pedestrians and transportation. Moreover, the patch is no longer limited to covering the patch only on the target to be detected. Instead, the patch can be placed in a random position in the image, causing the target to disappear from the detector.

This section explains how to address these challenges. In this paper, an iterative update is performed at the pixel level to train a patch that can effectively reduce the recall of the object detector, and the overall framework of the algorithm is shown in Fig. 1. At any position in the image, the algorithm places the current version of the patch on the image after applying different transformations. The resulting image is then fed to the detector, and the algorithm extracts the presence confidence scores of the targets that are still detected. There scores are used to compute the loss function, and the objective function continuously optimized through backpropagation over the entire network to obtain the final generated patch.

Next, this paper explains more in detail the process of generating these adversarial patches. Brown et al. of Google[26] generated patches by maximizing the loss of the CNN classifier when applying the patches to the input image. To make the patches effective under all inputs and potential transformations, the patches are transformed randomly before the inputs. Inspired by the

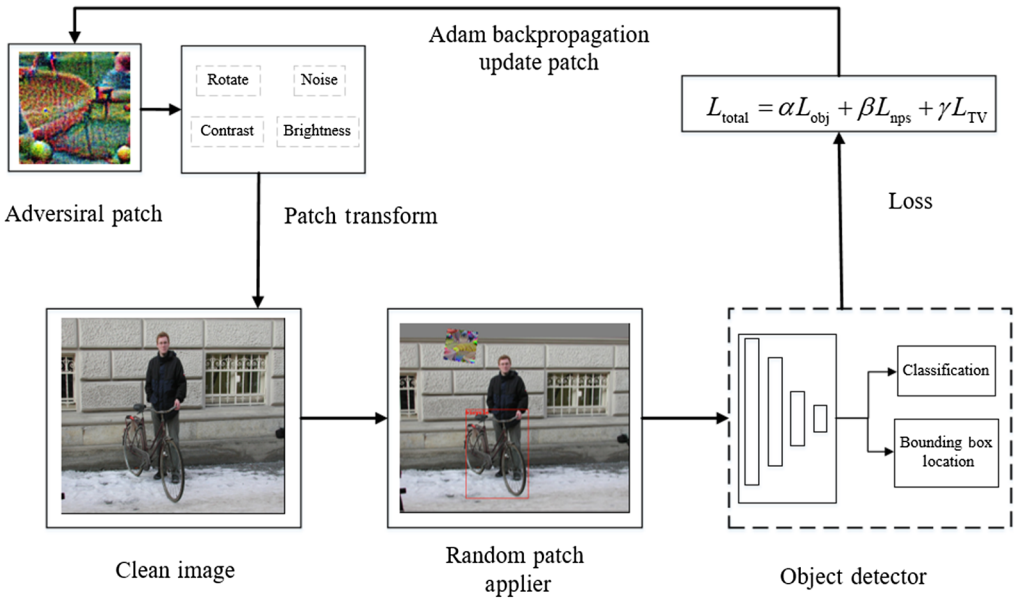

Fig. 1 Overall architecture of the patch generation method.

above work, this paper first overlays the patch on the original image and then calculates the object score loss of the object detector on the image. For all the objects involved in the training, the lower the value of the existence object score is, the better the optimization process is. When the object confidence falls below the threshold, the detector will identify the target as background, and at this time the object will disappear from the detector. This can be expressed as

$$\arg \min_{\delta} \mathbb{E}_{(x,y)\sim D, t\sim T}(J(h_\theta(A(\delta, x, t)), y)), \tag{1}$$

where D represents the distribution over the samples in the dataset, T denotes the patch-transformed distribution. The label value y is also included in this equation. Additionally, A is the patch application function, which indicates that the transformation t is applied to the patch before it is added to the original image x. That is, a random rotation and random position determination are performed on the patch, and then the patch is added to the image. This approach differs from the Thys team's method, which only targeted the "person" category in the image.[10] Instead, this function $J(\cdot)$ seeks to extract the object confidence score of all items in the image, with the aim of causing all detectable objects to disappear from the detector.

3.2 Objective Function

Specifically, the optimization of the objective function in this paper consists of three parts.

3.2.1 *Print loss*

To accommodate adversarial attacks in the physical world, Thys' team introduced non-printable loss and smoothing loss.[10] However, print loss and smoothing loss were primarily designed to enhance the visual fitness of the adversarial patch and are not directly linked to its attack capability.

Since the color gamut of the printer is limited, some colors of the digital display may not be printed, and in this case, so the printer fails to reproduce the colors of the digital display exactly, with the loss shown in the following equation:

$$L_{\text{NPS}} = \sum_{p_{\text{patch}} \in p} \min_{c_{\text{print}} \in c} \|p_{\text{patch}} - c_{\text{print}}\|_1, \tag{2}$$

where p_{patch} represents a pixel in a patch, and c_{print} is one of a set of printable colors. The print loss enables adversarial patches to be printed out with minimal color distortion by printers with limited color range.

3.2.2 *Smoothing loss*

The smoothing loss is to make the color of the patch smoother during optimization as well as to prevent noisy images, as shown in the following equation:

$$L_{\text{tv}} = \sum_{i,j} \sqrt{(p_{i,j} - p_{i+1,j})^2 + (p_{i,j} - p_{i,j+1})^2}. \tag{3}$$

The function of the smoothing loss is to make the color of the patch smoother, with the goal of making the neighboring pixels have similar colors. Specifically, the loss score is lower if the neighboring pixels are similar and higher if they are different, thereby minimizing the color difference between neighboring pixels to produce a smoother patch.

3.2.3 *Object confidence loss*

YOLO produces three outputs when an object is detected: the location of the bounding box, the confidence of presence, and the category probability. When the object confidence score is smaller than a threshold of 0.5, it is identified by the detector as a background region. The algorithm proposed in this paper aims to make all objects in the image disappear from the detector. To achieve this, the algorithm is trained to minimize the object confidence score of the detector output. Since the algorithm proposed in this paper restricts its attack to the presence confidence score of objects only, the whole training process become highly focused, and all the information on the adversarial patch is concentrated on attacking the presence confidence of objects. If the algorithm also attacked the category loss or location loss, it would have to attempt to find a feature domain that deviates from its original category or location during training process. However, in a high-dimensional, complex feature space, the feature vectors are allowed to deviate in any direction. Since there are many objects and different backgrounds in the dataset, these complex factors cause the feature vectors to fail to generate a uniform pointing and affect the convergence of the patch. Therefore, this paper chooses the confidence score of the object as the loss of the objective function, as shown in the following equation:

$$L_{\text{obj}} = \frac{1}{m} J(f(x_i)), \tag{4}$$

where x_i represents the first image of the current batch, and m images will be selected for training in each batch. $f(x_i)$ represents the output of the sample after inputting it to the detector f, including the bounding box location, presence confidence, and classification probability. The function $J(\cdot)$ is used to extract the confidence of object presence, and $J(f(x_i))$ represents the object confidence score of all detected objects extracted in the output:

$$J(f(x_i)) += \begin{cases} -0.5 \times \text{conf}, & \text{if } 0.2 \le \text{conf} < 0.3, \\ -1 \times \text{conf}, & \text{if } 0.3 \le \text{conf} < 0.5, \\ -2 \times \text{conf}, & \text{if } 0.5 \le \text{conf} < 0.7, \\ -4 \times \text{conf}, & \text{if conf} \ge 0.7. \end{cases} \tag{5}$$

The loss function for object detection is commonly comprised of three parts: classification, regression, and confidence losses. The classification loss measures the accuracy of the detected target's class while the bounding box position loss assesses the degree of difference between the predicted and real object box. The confidence loss indicates whether the predicted box contains the target, with higher confidence values suggesting that the bounding box is more likely to contain the target. Consequently, fixed threshold values (typically set to 0.5) are often used to identify foreground and background samples based on prediction box confidence. In IoU-based object detection frameworks, prediction boxes with obj-confidence values greater than the threshold are classified as foreground, while those below the threshold are considered background. During training, it is necessary to minimize multiple loss functions concurrently with the ultimate objective of obtaining the best detection outcome. In the context of an adversarial attack, the model is unable to predict the bounding box, which requires that all obj-confidence scores be <0.5. Consequently, our loss optimization becomes different, and the update direction for the loss becomes more biased toward the foreground class than the background class.

For obj-confidence scores >0.5, it is necessary to assign higher weights to reduce the confidence level faster, while obj-confidence scores below 0.5 require lower weights.

This paper exhibits creativity by assigning different weights for different confidence scores to optimize the direction of the gradient update, balance the contribution of the foreground and background to the loss function, and improve the efficiency of the attack. The weight assignment is shown in Eq. (5).

Thus the overall optimization objective can be expressed as

$$L_{\text{total}} = \alpha L_{\text{obj}} + \beta L_{\text{nps}} + \gamma L_{\text{TV}}. \tag{6}$$

The overall objective of the algorithm presented in this paper is to minimize the total loss. During the optimization process, the algorithm freezes all weights in the network and updates only the pixel values in the patches.

3.3 Training

Initially, the patch is a noisy pattern. To enhance the robustness of the patch during the optimization process, this paper adopts the expectation of transformation strategy by performing a set of transformations on the patch before applying it to the image. These transformations include random rotation, angle deflection, noise addition, brightness, contrast adjustment, and other operations.

In particular, to reduce the sensitivity of patches to position, the patches are positioned at random locations in the image during each iteration, and object confidence score is extracted from the output to construct the loss function. The process of training adversarial patches, in which the location of the patch is randomized rather than fixed in a certain region, is intended to enhance its ability to generalize across different scenarios and locations. If the patch is fixed to a certain location, it would only be effective in attacking targets in that specific location and would be incapable of effectively attacking in other regions. By randomizing the location of the patch during training, the patch is exposed to different locations and scenarios, which better prepares it to adapt to attacks from different directions and locations. This randomization also

Algorithm 1 IndependentPatchGen.

Input: A clean picture x, object detection model to be attacked f, decay factor: β_1, β_2

The maximum number of iterations: $T = 5000$

Confidence extraction function

Output: Adversarial patch δ

1 While $t < T$:

1	**1**	$x_{\text{attack}} = x + \text{transform}(\delta)$ Transform the patch and randomly place it in the image
2	**2**	$L_{\text{obj}} = \frac{1}{m}\sum_{i=1}^{m} J_{\text{obj}}(f(x_{\text{attack}}))$ Extract obj loss after the patch input detector
3	**3**	L_{NPS}, L_{TV} Calculate the non-printable loss and smooth loss
4	**4**	$L_{\text{total}} = \alpha L_{obj} + \beta L_{nps} + \gamma L_{TV}$ Calculate the total loss
5	**5**	$g_t = \frac{\partial L_{\text{total}}}{\partial \delta}$ Calculate the gradient of the total loss
6	**6**	$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$ Calculate the first-order moment of the gradient
7	**7**	$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ Calculate the second-order moment of the gradient
8	**8**	$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$ Perform bias correction
9	**9**	$\delta_{t+1} = clip_{(0,1)}\left\{\delta_t + \frac{\eta}{\sqrt{\hat{v}_t}+\varepsilon}\hat{m}_t\right\}$ update the patch using Adam
10	**10**	$t = t + 1$
11	end while	

Return δ

reduces its sensitivity to specific regions, making it more versatile and flexible for use in a variety of situations. In summary, randomizing patch locations during the training process enhances its versatility in different environments, making it more effective and adaptable for use in various scenarios.

To update the pixel values of the patch pattern, we utilize Adam's algorithm for backpropagation. This algorithm applies independent adaptive learning rates to different parameters, making it efficient in multi-dimensional optimization problems with smaller gradients, where it can speed up the descent of the loss function, jump out of local minimum values, and promote better convergence.

4 Analysis of Data and Experimental Results

4.1 Dataset and Experiment Details

The CPU of this experimental environment is a 4× 10-core Intel® Xeon® E5-2650 processor. The GPU is Nvidia RTX 3080Ti graphics card with 11 GB of video memory, the GPU driver environment is CUDA10.0, the development language is based on Python3.6. Moreover, this paper utilizes Pytorch as the primary deep learning framework and supplement it with Numpy, Opencv, and other necessary third-party libraries.

The experiments in this paper are based on the YOLO-V2 model trained from the PASCAL COCO dataset. The patch training dataset is the Inria dataset, INRIA Person is a multi-environment pedestrian dataset, which is one of the most popular and most used static pedestrian detection datasets at present, published by INRIA (the National Institute of Information and Automation, France). Recall calculation is based on an IOU of 0.5. Other parameter settings are shown in Table 1.

4.2 Evaluation Indicators

4.2.1 *Recall*

There are different evaluation criteria for measuring the performance of object detection in deep learning. First, this paper introduces the concept of positive and negative cases in detection results. To be specific, true positive (TP) refers to the number of samples that are originally positive and the detection results are positive. False positive indicates the number of samples that are originally negative and the detection results are positive. False negative (FN) indicates the number of samples that are originally positive and the detection result is negative. True negative indicates the number of samples that are originally negative and the detection results are negative. Recall is a performance metric that measures the percentage of all true targets detected by our model, and the calculation formula is defined as follows:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{7}$$

4.3 Experimental Results

As mentioned earlier, the main goal of this paper is to train a patch for the Inria dataset that deceives the detector. This paper achieves this by placing the patch on different images to create

Table 1 Hyperparameter settings for the algorithm.

Parameter	Value
Maximum number of iterations	$T = 5000$
Objective function hyperparameters	$\alpha = 1$, $\beta = 0.01$, and $\gamma = 2.5$
Adam decay rate	$\beta_1 = 0.9$ and $\beta_2 = 0.999$
Batch size	$m = 16$
Initial learning rate	$\eta = 0.001$

a generalized adversarial patch. Once the patch is attached to the input image, the detector does not extract a valid ROI and misclassifies all targets as background regions due to the low confidence score. This results in the disappearance of all targets from the detector.

4.3.1 *Recall*

In this paper, patches are trained with pretrained YOLO-V2, YOLO-V5, and YOLO-V6 models in the COCO dataset and applied to the Inria test set to evaluate the patch attack effectiveness. These models achieve a recall rate of ∼100% at an IOU threshold of 0.5, indicating that the models can detect almost all targets. Since the recall rate is heavily affected by the threshold, this paper also evaluates it at an IOU threshold of 0.5 during the validation period.

The main purpose of applying this patch is to decrease the recall of all categories in the dataset to lower values, and the more the recall is reduced, the more successful the attack becomes. As depicted in Table 2, our approach successfully deceives nearly all categories within the Inria dataset after ∼1500 training iterations. The recall rate diminishes from 100% to 41.3% on YOLO-V2, from 97.7% to 31.2% on YOLO-V5, and from 98.2% to 29.5% on YOLO-V6. Moreover, the patch in our study can suppress the detection of all objects in the image, not just pedestrians, as shown in Fig. 2, in comparison to the Thys team's work.

4.3.2 *Quantitative analysis*

Analysis of the convergence time and complexity of the algorithm. The term "epoch" refers to each instance where the algorithm uses all available samples. In the backpropagation process, this paper utilized the Adam optimization algorithm, set the initial learning rate to 0.001, specified the maximum number of iterations to 5000, and set the batch size to 16.

As illustrated in Fig. 3, the rate of loss decline is most notable at the beginning of the training period (when training iterations are <500 epochs). As the patch receives more training, the effect of its attack becomes less potent. Hence, we concluded that a saturation point exists for the training patches. Increasing the number of training iterations beyond this point will no longer improve the attack's effectiveness.

Table 2 Recall of adversarial patch attacks against different object detection models.

Model	Clean (%)	Recall (%)
YOLO-V2 trained PATCH	100	41.3
YOLO-V5 trained PATCH	97.7	31.2
YOLO-V6 trained PATCH	98.2	29.5

(a) (b)

Fig. 2 Example of our patch on the Inria test set, which is not only effective for pedestrians but also inhibits the detection of all objects in the image. (a) Original image and (b) attack.

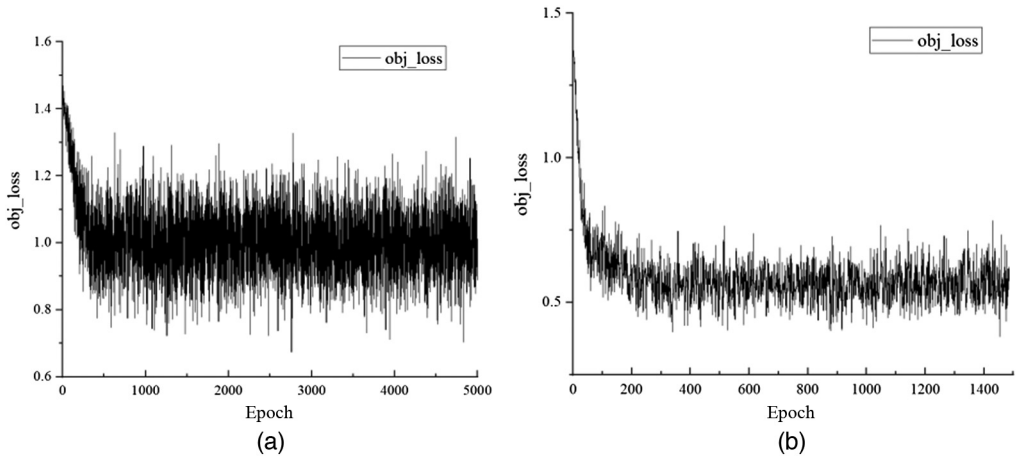

Fig. 3 Comparison of convergence times of the two methods: (a) the approach of Thys' team and (b) ours.

In addition, to analyze the convergence and complexity of the algorithm, this paper conducts ablation experiments on object confidence score loss to verify the effect of adding and not adding weighting on patch convergence. As shown in Fig. 3, the object confidence score decreases faster when the foreground region is given more weight. In this paper, we obtained a training level of 5000 epochs for the baseline method with ∼1500 epochs of training, significantly reducing the training time.

Randomness of patch location. The location-independent property of the patches implies that the same patch can appear anywhere in the image. The primary purpose of this approach is to assess the attack's effectiveness at different positions where patches are placed. If the attack efficiency does not vary for different positions, there is no need to design a specific attack area. This means that the attacker can place patch in any region of the image. Figure 4 depicts the detector being attacked by a randomly located patch. The first row presents the detection results for a clean sample, and the second row illustrates the detection results after adding the patch. Notably, the location of the patch does not impact the attack results. Therefore, detection suppression can be achieved for objects in the image regardless of the patch's location.

During the detection process, all identified objects are misclassified as background, regardless of the patch's position. As a result, we can place the patch randomly in the image without designing its specific location. This enhances the attack's feasibility.

Comparison of the number of patches. To investigate the impact of patch quantity on attack effectiveness, this paper compares the patches generated by Thys' team with those in this study. Figure 5(a) depicts the result when only one patch generated using the Thys team's method

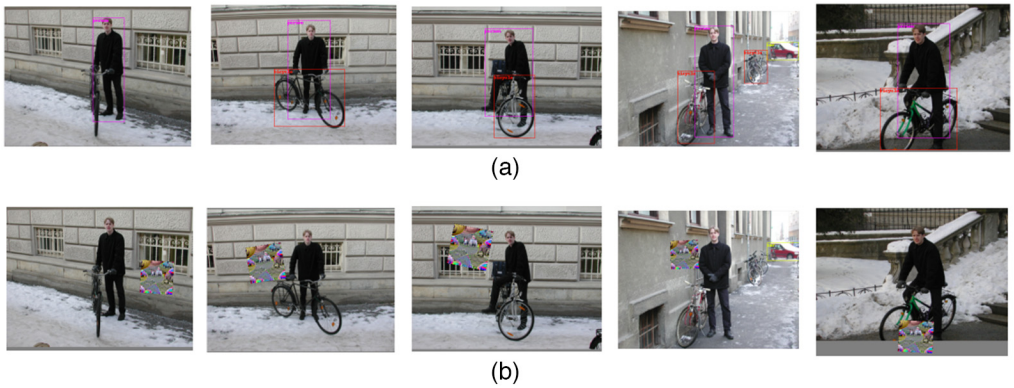

Fig. 4 Examples of our patch with random location: (a) original detection and (b) adversarial attack.

Fig. 5 Examples of (a) one patch attack, (b) the original patch attack of the Thys team's approaches, and (c) our patch attack.

was stuck on the entire image, whereas Fig. 5(b) depicts the original method in which a patch was applied to each target. Figure 5(c) shows the result when only one patch generated using the approach taken in this paper was stuck on the entire image.

The method in this paper can suppress all the objects in the image, whereas the Thys team's method needs to apply patches to each target individually and has no attack capability for other objects without sticky patches. Moreover, the recall was computed for the three patch-sticking methods discussed above, as presented in Table 3. The results reveal that the Thys team approach is more susceptible to the number of patches, leading to an increase in recall up to 60.5%. In summary, the use of adversarial patches leads to varying degrees of error in the detector, but the method presented in this paper can achieve stronger attacks using fewer patches.

Table 3 Comparison of one patch, the original patch of the Thys team's approaches, and our patch in terms of recall.

Approach	Recall (%)
CLEAN	100
OBJ with one patch	60.5
OBJ	48.6
Ours	41.3

4.3.3 *Comparison experiments*

In this section, we conduct an experimental evaluation of our proposed method along with other approaches, such as random noise and the Thys team's methods. These comparative experiments are intended to assess the performance, effectiveness, and practicality of our proposed method across various models. By conducting these comparison experiments, we observe that our method described in this paper attains the highest attack success rate on several models.

As seen in Tables 4–6, the impact of the random noise approach on recall is minimal, making it ineffective in terms of attack capability. However, when considering the YOLO-V2 and YOLO-V5 models, our proposed method outperforms the other two baseline methods significantly, resulting in a minimum recall rate of *%. Furthermore, we conducted comparative experiments on the two-stage model, Faster-RCNN, using our proposed method described in this paper. The results demonstrate that our method achieves the highest attack success rate. These findings indicate that our method holds greater potential for launching attacks on different models.

In conclusion, this section presents an evaluation through comparative experiments with other methods. The results demonstrate that the method proposed in this paper achieves a high success rate on several models, thereby establishing the superiority and feasibility of our approach. Consequently, it offers enhanced capabilities for addressing various attack scenarios.

4.3.4 *Ablation experiments*

In this paper, the ablation experiments are conducted to examine the influence of various loss functions on the patch attack capability and assess their respective attack success rates. Specifically, we employ different loss function terms on the YOLO-V5 model to establish strategies that effectively evaluate each term's impact. Three strategies are selected for comparison, with recall serving as the evaluation metric. The experimental results are shown in Table 7.

Table 4 Recall (%) of adversarial patch attacks against YOLO-V2.

YOLO-V2	Clean (%)	Recall (%)
Random noise	100	91.8
Thys' method	100	48.6
Ours	100	41.3

Table 5 Recall (%) of adversarial patch attacks against YOLO-V5.

YOLO-V5	Clean (%)	Recall (%)
Random noise	97.7	96.7
Thys' method	97.7	42.5
Ours	97.7	31.2

Table 6 Recall (%) of adversarial patch attacks against YOLO-V6.

YOLO-V6	Clean (%)	Recall (%)
Random noise	98.2	96.3
Thys' method	98.2	38.4
Ours	98.2	29.5

Table 7 Trade-off experiments of different loss functions on YOLO-V5.

Print loss	Smoothing loss	Obj loss	Recall (%)
✓	✗	✗	97.5
✗	✓	✗	96.3
✗	✗	✓	30.7
✓	✓	✓	31.2

Table 8 Trade-off experiments of different loss functions on YOLO-V2.

Approach	Recall (%)
CLEAN	100
OBJ-CLS	59.5
OBJ	48.6
Ours	41.3

The experimental results demonstrate that different loss functions significantly affect the efficacy of patch attacks. Among them, the print loss and smoothing loss methods exhibit minimal impact on the attack success rate. As observed in Table 8, employing only print loss and smoothing loss hardly influences the recall rate, indicating a lack of attack capability. Conversely, utilizing object confidence loss yields a more pronounced attack capability, resulting in a greater decrease in the recall rate. Thus it can be concluded that print loss and smoothing loss primarily contribute to enhancing the visual appearance of the adversarial patch and do not directly influence its attack capability.

Furthermore, object detection can be regarded as a unified framework for both regression tasks (bounding box location) and classification tasks (target category), as it requires precise target localization and accurate target classification. Consequently, multiple loss functions are necessary for effective training. For our patch-based training approach, we employed a different combination of position loss, confidence loss, and category loss functions in the YOLO-V5 output. The results presented in Table 8 demonstrate the varying performance of these loss functions in the attack task. Our method incorporates a weighted confidence loss, assigning greater weight to the foreground region, resulting in improved attack outcomes.

To summarize, our ablation experimental results demonstrate the significant influence of various loss function choices on the effectiveness of adversarial patch. These findings serve as a reference and guide for future enhancements in object detection adversarial patch generation methods. Furthermore, they contribute to a better understanding of the impact that different loss functions have on the adversarial patch attack task.

5 Conclusions

In this paper, we demonstrate the attack capability of the method on the pedestrian detection dataset by minimizing the confidence score of the detector output to generate patches. Compared to previous work, the patch in this paper is more robust and general because: (1) the method proposed in this paper only requires attaching a patch to the entire image to perform an effective attack on the detector. Furthermore, it does not attack a single type of target but rather suppresses the detection of all objects in the image. (2) The attack of the method successfully suppresses the detection without the need to overlap the patch with the target object. Additionally, it is less sensitive to the patch's location. The successful implementation of the work in this paper also highlights the inherent vulnerability of deep learning-based detectors to patch-based adversarial attacks. This finding is of great significance when studying the robustness of deep neural networks and adversarial defense.

Data Availability

The data used in this study are available upon reasonable request. To access the data, interested researchers can contact the corresponding author (provide email address or any other contact information) and submit a formal request outlining the purpose of data usage and the intended analyses. Access to certain sensitive or confidential information may require additional permissions and data use agreements.

Acknowledgment

We would like to express our gratitude to Xiuqing Mao, Leyu Dai, and Bayi Xu. We would also like to extend our appreciation to Prof. Lei Sun. Their valuable support and help have made this research possible. Finally, we would also like to thank the anonymous reviewers for their comments and suggestions that have helped to improve the quality of this paper.

References

1. N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: a survey," *IEEE Access* **6**, 14410–14430 (2018).
2. H. Li et al., "Adversarial examples for CNN-based SAR image classification: an experience study," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **14**, 1333–1347 (2021).
3. S. Mane and S. Mangale, "Moving object detection and tracking using convolutional neural networks," in *Second Int. Conf. Intell. Comput. and Control Syst. (ICICCS)*, IEEE, pp. 1809–1813 (2018).
4. M. Shen et al., "Effective and robust physical-world attacks on deep learning face recognition systems," *IEEE Trans. Inf. Forensics Security* **16**, 4063–4077 (2021).
5. Z. Xiong et al., "Multi-source adversarial sample attack on autonomous vehicles," *IEEE Trans. Veh. Technol.* **70**(3), 2822–2835 (2021).
6. S. K. A. Fahad and A. E. Yahya, "Inflectional review of deep learning on natural language processing," in *Int. Conf. Smart Comput. and Electron. Enterprise (ICSCEE)*, IEEE, pp. 1–4 (2018).
7. C. Szegedy et al., "Intriguing properties of neural networks," Computer Science (2013).
8. N. Akhtar et al., "Threat of adversarial attacks on deep learning in computer vision: survey II," arXiv.2108.00401 (2021).
9. T. B. Brown et al., "Adversarial patch," in *Comput. Vis. and Pattern Recognit.* (2017).
10. S. Thys et al., "Fooling automated surveillance cameras: adversarial patches to attack person detection," in *Proc. IEEE/CVF Conf. Comput. Vis. and Pattern Recognit. Workshops* (2019).
11. J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 7263–7271 (2017).
12. Z. Zou et al., "Object detection in 20 years: a survey," arXiv.1905.05055 (2019).
13. R. Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conf. Comput. Vis. and Pattern Recognit.*, IEEE Computer Society (2013).
14. S. Ren et al., "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2015).
15. W. Liu et al., "SSD: single shot multibox detector," *Lect. Notes Comput. Sci.* **9905**, 21–37 (2016).
16. J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," in *Comput. Vis. and Pattern Recognit.* (2018).
17. Z. Li et al., "When object detection meets knowledge distillation: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(8), 10555–10579 (2023).
18. C. Yan et al., "Semantics-guided contrastive network for zero-shot object detection," *IEEE Trans. Pattern Anal. Mach. Intell.* (2022).
19. I. J. Goodfellow et al., "Explaining and harnessing adversarial examples," arXiv:1412.6572 (2014).
20. A. Madry, et al., "Towards deep learning models resistant to adversarial attacks," Machine Learning (2017).
21. S. M. Moosavi-Dezfooli et al., "DeepFool: a simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 2574–2582 (2016).
22. S. M. Moosavi-Dezfooli et al., "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 1765–1773 (2017).
23. N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symp. Security and Privacy (SP)*, IEEE, pp. 39–57 (2017).
24. J. Lu et al., "Adversarial examples that fool detectors," (2017).
25. C. Xie et al., "Adversarial examples for semantic segmentation and object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1369–1378 (2017).
26. Y. Li et al., "Robust adversarial perturbation on deep proposal-based models," in *BMVC* (2018).
27. X. Wei et al., "Transferable adversarial attacks for image and video object detection," in *IJCAI* (2018).

28. K. Eykholt et al., "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 1625–1634 (2018).
29. D. Song et al., "Physical adversarial examples for object detectors," in 12th USENIX Workshop on Offensive Technologies (WOOT 18) (2018).
30. S. T. Chen et al., "ShapeShifter: robust physical adversarial attack on faster R-CNN object detector: recognizing outstanding," PhD thesis (2019).
31. M. Lee and Z. Kolter, "On physical adversarial patches for object detection," in *Comput. Vis. and Pattern Recognit.* (2019).
32. X. Liu et al., "DPATCH: an adversarial patch attack on object detectors," in *Natl. Conf. Artif. Intell.* (2019).
33. K. Xu et al., "Adversarial T-shirt! Evading person detectors in a physical world," in *Comput. Vis. and Pattern Recognit.* (2019).

Zhiyi Ding is a PhD student with extensive knowledge and experience in the field of artificial intelligence and security. Specifically, he is currently researching the impact of adversarial examples in the field of object detection. With a focus on developing defense strategies for object detection models, he aims to contribute innovative solutions to protect the increasingly important role of AI in our society.

Biographies of the other authors are not available.