 doi:10.20944/preprints202210.0131.v1

Article

Adversarial Patch Attack on Multi-Scale Object Detection for Remote Sensing Image

Yichuang Zhang [1], Yu Zhang [1], Jiahao Qi [1], Kangcheng Bin[1], Hao Wen [1] and Ping Zhong [1,*]

[1] National Key Laboratory of Science and Technology on Automatic Target Recognition, National University of Defense Technology, Changsha 410073, China; ycz@nudt.edu.cn (Y.Z.); Zhangyu13a@nudt.edu.cn (Y.Z.); qijiahao19@nudt.edu.cn (J.Q.); binkc21@nudt.edu.cn (K.B.); hao.wen@nudt.edu.cn (H.W.)

* Correspondence: zhongping@nudt.edu.cn

Abstract: Although deep learning has received extensive attention and achieved excellent performances in various of scenarios, it suffers from adversarial examples to some extent. Especially, physical attack poses more threats than digital attack. However, existing researches pay less attention to physical attack of object detection in remote sensing images (RSIs). In this work, we systematically analyze the universal adversarial patch attack for multi-scale objects in the remote sensing field. There are two challenges for adversarial attack in RSIs. On one hand, the number of objects in remote sensing images is more than that of natural images. Therefore, it is difficult for adversarial patch to show adversarial effect on all objects when attacking a detector of RSIs. On the other hand, the wide range of height of photography platform causes that the size of objects diverse a lot, which brings challenges for generating universal adversarial perturbation for multi-scale objects. To this end, we propose an adversarial attack method on object detection for remote sensing data. One of the key ideas of the proposed method is the novel optimization of adversarial patch. We aim to attack as many objects as possible by formulating a joint optimization problem. Besides, we raise a scale factor to generate a universal adversarial patch that adapts to multi-scale objects, which ensures the adversarial patch is valid for multi-scale objects in the real world. Extensive experiments demonstrate the superiority of our method against state-of-the-art methods on YOLO-v3 and YOLO-v5. In addition, we also validate the effectiveness of our method in real-world applications.

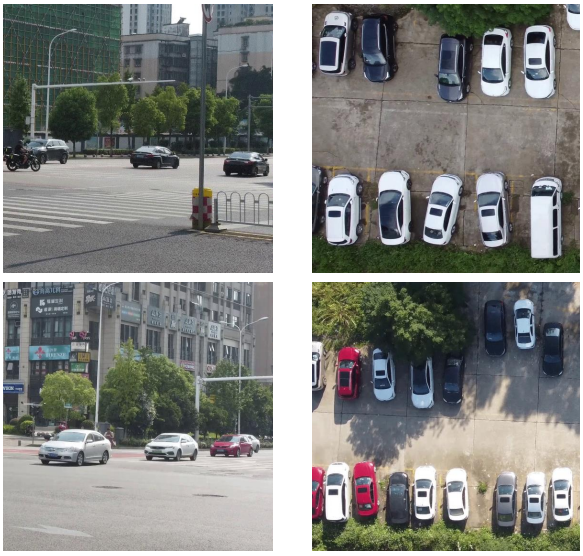
Keywords: Adversarial examples; Remote sensing images; Universal adversarial patch; Object detection; Joint optimization; Scale factor.

1. Introduction

The continuous development of aerial photography technology makes it possible for people to collect numerous high-resolution remote sensing images, which contributes a lot to many important applications in the remote sensing field [1–3]. Some typical applications include classification [4,5], image segmentation [6] and object detection [7–10]. Specifically, object detection, which tries to precisely estimate the class and locations of objects contained in each image, is one primary task [11–13]. Deep learning models, which learn a hierarchical representation of features, have been widely used and achieved a great success in many fields [14–17]. Currently, deep learning plays an significant role in most of the state-of-the-art methods in RSIs. Moreover, the constant improvement of structure of deep neural networks brings better performances.

Despite the great success that deep learning has achieved, several potential security problems should not be neglected. Recent researches found that deep models are extremely vulnerable to adversarial examples, which can be simply generated by adding carefully designed perturbation to clean examples. Szegedy et al. [18] first revealed the fragility of deep neural networks and raised the concept of adversarial examples. Since then, an increasing number of researchers have devoted to exploring the security of deep learning algorithms and corresponding methods about how to generate adversarial examples are

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1



(a) (b)

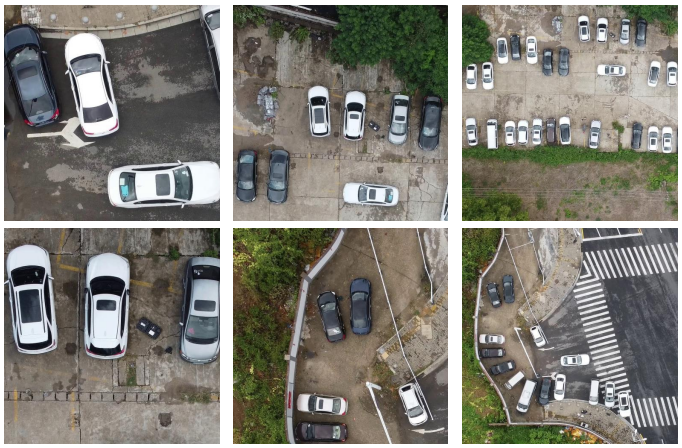
Figure 1. The illustration of the images captured in the ground and the remote sensing images. (a) Images captured in the ground. (b) Remote sensing images. The number objects of remote sensing images is larger than that of images captured in the ground.

proposed. The proposed adversarial attack methods can be divided into digital attacks and physical attacks based on the domain where the adversarial perturbations are added. Digital attacks directly modify the pixel values of the input images in the digital space. Typical digital attack methods include FGSM [19], PGD [20], DeepFool [21], UAP [22] C&W [23] and JSMA [24]. However, most of the digital attack methods can't be applied in the real world, because the perturbation may be easily filtered when performing physical attacks. As for physical attacks, the generated perturbation (e.g. adversarial glass [25], adversarial T-shirt [26], adversarial patch [27]) is always large, and it can be printed before applied in the physical world. Kurakin et al. [28] first performed experiments to verify that adversarial examples also exist in the real world. Thys et al. [29] and Hu et al. [30] generated adversarial patches to fool a person detector.

In addition to the above-mentioned adversarial attack methods, there are currently some works [31–33] about adversarial examples in the remote sensing field. Czaja et al. [34] first generated adversarial examples for classification model in RSIs. They attacked a small domain of one image to lead the model to make wrong predictions. Xu et al. [35] proposed to generate universal adversarial examples to realize black-box attack on different models. Chen et al. [36] designed experiments to attack synthetic aperture radars (SAR) images. Xu et al. [37] indicated that hyperspectral images were also affected by adversarial examples. However, it is more difficult to attack object detection than image classification because the number of bounding boxes may be very large. When the confidence of one bounding box drops, the others may still work. That is why object detectors are hard to attack. Lu et al. [38] designed a scale-adaptive patch to attack object detection in digital domain for RSIs. Du et al. [39] perform the digital and physical attack in an aerial surveillance model, which is the first work to demonstrate the physical attack in aerial scenes.

So far, adversarial attack on deep models for RSIs has not been fully explored yet. There are still several limitations for adversarial attack in RSIs. 1) Majority of the existing methods mainly focus on the digital attack, but they are constrained when applying to the physical attack of object detection 2) The number of objects in RSIs tends to be much more than images captured on the ground, which can be observed in Figure 1. Therefore, it is difficult for adversarial patch to show adversarial effect on all objects when attacking a detector of RSIs. 3) Remote sensing images are obtained by using earth-observing photography platforms and the height of the platform is always at a wide range, which causes the size

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1



(a) (b) (c)

Figure 2. The illustration of images captured at different heights. (a) Images captured at the height of 30m. (b) Images captured at the height of 60m. (c) Images captured at the height of 120m. The variation of photography height makes the objects vary a lot in size.

of the objects ranges a lot, which brings challenges for generating universal adversarial perturbation for multi-scale objects. Figure 2 shows several images captured by a UAV from the heights of 30m, 60m and 120m respectively. The size of objects is quite diversified and the number of objects in a single image is large, especially for images captured at a large height.

Based on the above analysis, this work aims to conduct adversarial attacks against object detection in digital and physical domains for RSIs. We formulate a joint optimization problem to generate a more effective adversarial patch. In order to attack as many objects as possible, a natural idea is to take the average confidence as part of the loss, namely object loss. The average confidence is the average value of the confidence of all bounding boxes in a single image. The computation of average confidence involves all objects for one image, so it is reasonable to believe that more objects will be attacked with the average confidence loss. Considering the constraint of object loss does not take the detection result into consideration, detect loss between the detection results and the ground truth is also introduced in our method. Thus, we can degrade the metrics (AP, Precision and Recall.) by minimizing detect loss. Experimental results demonstrate that the attack effect will be improved with the combination of the two losses, and relevant theoretical analysis will be shown in the ablation study of experiments. In consideration of the actual situation, we propose a novel method to make the size of adversarial patch match with the size of objects in digital attack, which ensures the adversarial patch is valid for multi-scale objects. All the images in our experiments have labels about the height of objects. When carrying out digital attacks, the adversarial patch will be scaled to the responding size with the scale factor which depends on the height label of image. Finally, the experimental results show the effectiveness of our method. The contribution of this work can be summarized in the following three points:

1) To the best of our knowledge, this is the first work to perform physical adversarial attack on multi-scale objects in the remote sensing field and the data in experiments are captured from 25m to 120m.

2) We propose a novel method to attack object detection for remote sensing data. For the optimization of adversarial patch, we formulate a joint optimization problem to generate a more effective adversarial patch. Moreover, in order to make the generated patch valid for multi-scale objects in the real world, we raise a scale factor which depends on the height label of image to rescale the adversarial patch when performing digital attack.

3) To verify the superiority of our method, we carry out several comparison experiments on digital attack against Yolo-V3 and Yolo-V5. Experimental results demonstrate

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

that our method has a better performance than baseline methods. In addition, we perform experiments to test the effect of our method in the physical world.

The remainder of this paper is organized as follows. Section 2 briefly reviews related attack methods. In Section 3, the details of proposed method are demonstrated as detailed as possible. Section 4 shows the experimental results and the corresponding analyses. In Section 5, we draw a comprehensive conclusion for the whole work.

2. Related Work

2.1. Digital Attack and Physical Attack

An increasing number of researchers pay attention to the safety of deep learning since Szegedy et al. [18] proposed the conception of adversarial example. Currently, the proposed adversarial attack methods can be categorized into digital attack and physical attack based on the domain where the adversarial perturbations are added.

Digital attack. The premier researches [18–24] of adversarial attacks mainly focus on digital attack, in which tiny perturbation is added to original input images to make the target model output wrong predictions. Szegedy et al. [18] proposed L-BFGS to generate adversarial examples for the first time. Based on the gradient information, Fast Gradient Sign Method (FGSM) [19] proposed by Goodfellow et al., was aimed to quickly find an adversarial example for a given input. Madry et al. [20] proposed Projected Gradient Decent (PGD), a first-order attack. DeepFool [21] was another typical attack algorithm which estimated the distance of an input to the closest decision boundary, and it has successfully attacked lots of models. Carlini and Wagner proposed C&W [23] to find adversarial perturbations by minimizing similarity metrics: L_0, L_2, and L_∞. Chow et al. [40] presented Targeted Adversarial Objectness (TOG) to make the object detection suffer from object-vanishing, object-fabrication, and object mislabeling attacks. Liu et al. [41] proposed DPatch, adding an adversarial patch on the images to stop detectors to detect objects. Although these attack methods have achieved a great success in the digital domain, their effectiveness would fade significantly when applied in the real world.

Physical attack. Compared with digital attack, physical adversarial attack poses more threats in specific scenarios. In [28], Kurakin et al. first studied whether adversarial examples generated by digital attack remained adversarial after they were printed. Sharif et al. [25] attempted to generate adversarial glasses to fool the face recognition, and they first proposed the non-rintability score (NPS) and the total variation (TV) loss. Athalye et al. [42] proposed Expectation Over Transformation (EOT) to generate 3D adversarial object that could remain adversarial in the physical world, and the adversarial objects were robust to rotation, translation, lighting change, and viewpoint variation. To generate adversarial examples for physical objects like stop sign, Eykholt et al. [43] introduced Robust Physical Perturbations (RP2) attack method by drawing samples of experimental data and synthetic transformations with varying distances and angles. [44–47] have generated adversarial stop signs to fool object detectors(e.g. Yolo-V2 [48], Faster R-CNN [7]). Thys et al. [29], Wang et al. [49] and Hu et al. [30] generated adversarial patches to attack a person detector. Wang et al. proposed the Dual Attention Suppression (DAS) [50] which generated visually-natural physical adversarial camouflages by suppressing both model and human attention with Grad-CAM.

2.2. Adversarial Attack in the Remote Sensing Field

As deep learning becomes more popular in the field of remote sensing, relevant research about adversarial examples are introduced to the remote sensing field inevitably.

Adversarial attack on classification. Czaja et al. [34] first proposed to attack the classification model for RSI. Xu et al. focused on the black-box attack to generate universal adversarial examples that can fool different models. Chen et al. paid attention to the generation of adversarial examples about SAR images. In [37], the research of adversarial attack is extended to the hyperspectral domain.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

Adversarial attack on object detection. Du et al. [39] reported the adversarial attacks against car detectors in aerial scenes, where the patches were trained with consideration of non-printability score, total variation score, several geometric and color-space augmentations. It is worth noting that they specially designed a kind of patch that could be placed around the car, which could prompt a more realizable and convenient attack in the physical world. Lu et al. [38] proposed an attack method for aircraft detectors in RSIs which had the characteristic of adversarial patch size adaption.

Existing research on adversarial attacks in the remote sensing field mainly focuses on digital attack of classification. Although several adversarial attack methods for RSIs were proposed, there are few works to carry out physical attack on multi-scale objects and evaluate the attack effect on objects of different scales in the remote sensing field. In this paper, we focus on the generation of universal adversarial patches to attack car object detectors and test the attack effect on objects of different scales in the digital domain and the physical domain.

3. Approach

In this section, we will introduce the adversarial attack framework of this work. First, the problem formulation is presented. Next, we will demonstrate the transformations for the adversarial patch. Then, we will state the optimization of the adversarial patch. Finally, the flowchart of proposed method will be shown.

3.1. Problem Formulation

In this work, we focus on the digital attack and physical attack against two detectors, Yolo-V3 and Yolo-V5, which are widely used in object detection. Given an input image $x \subseteq R^{N\times H\times W}$ and the target object detector $f(\cdot)$. The outputs $f(x)$ are a set of candidate bounding boxes $\hat{B}(x) = \left\{\hat{b_1}, \hat{b_2}, \hat{b_3}, \ldots, \hat{b_n}\right\}$, where $\hat{b_i} = \left(\hat{x_i}, \hat{y_i}, \hat{w_i}, \hat{h_i}, \hat{C_i}, \hat{P_i}\right)$. $(\hat{x_i}, \hat{y_i})$ is the center of the i^{th} box, $\hat{w_i}$, $\hat{h_i}$ are the width and the height of the i^{th} box respectively, and $\hat{P_i}$ is the probability which decides the class of the i^{th} box. The detectors may output a great number of bounding boxes in most cases, but most of them will be suppressed through the non-maxima suppression (NMS) with a confidence threshold and intersect over union (IOU) threshold. It generally reminds us to minimize the confidence of object so that they can be filtered out to vanish under object detector. Adversarial patch attack is considered in this work, and we will add the adversarial patch on every object to hide all the objects under the two detectors. Adversarial patch is a frequently-used method for physical attack, and it is a kind of universal perturbation. In digital attack, the adversarial patch will replace a part of pixels of the objects to form an adversarial example. When it comes to the physical attack, the adversarial patch can be printed and placed on the objects. The adversarial example can be denoted as:

$$x_{adv} = A(x, P), \tag{1}$$

where x_{adv} denotes adversarial example, P is the adversarial patch and $A(\cdot)$ denotes the apply function which is aimed at attaching the adversarial patch to objects. We attempt to generate a universal adversarial patch that may attack all car objects. The optimization process can be defined as:

$$arg\min \Sigma(NMS(f(A(x, P)), conf_{th}, iou_{th})), \tag{2}$$

where $NMS(\cdot)$ denotes non-maxima suppression, $\Sigma(\cdot)$ is a count function which output the number of detected objects. $conf_{th}$ and iou_{th} are confidence threshold and IOU threshold. We hope that the detected objects are as few as possible after performing non-maxima suppression.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

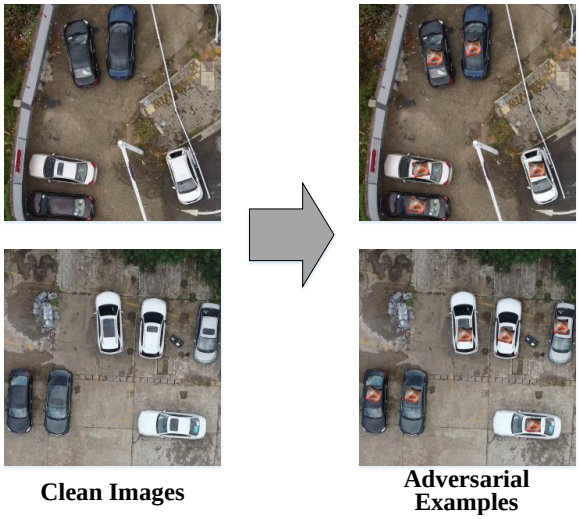


Figure 3. The illustration of clean images and adversarial examples.

3.2. Transformations of Adversarial Patch

The perturbation of most attack methods can be classified as global and local perturbation. But most global perturbation is so small that it can't work in the physical world in most cases. Adversarial patch is a kind of local and visible disturbance, which is expected to maintain the adversarial character in the physical world. It should be mentioned that apply function is a vital tool when performing adversarial patch attack, and it serves two functions. One is that we can attach the adversarial patch to the objects to generate the adversarial examples through apply function. The other is that it can make a series of transformations (e.g. patch rotation, patch rescale) for the adversarial patch. Patch rotation is to make the adversarial patch more robust in the physical world. Specifically, we make a random rotation ($\pm 20^{\circ}$) on the embedded patch. In order to generate universal adversarial patch which is valid for multi-scale objects, the adversarial patch needs to be scaled appropriately to adapt to the size of objects. To satisfy this requirement, we propose a method that the patch size adapts to the heights of the photography platform, which guarantees the patch size is consistent in the same image. To be specific, by measuring the size of the objects in images captured at the height of h, we can compute the size s_h that the patch need to be scaled. Then the scale factor can be expressed as:

$$\varepsilon_h = \frac{s_p}{s_h}, \tag{3}$$

where s_p is the original size of patch.

Finally, we can get a scale factor vector

$$\varepsilon_h = \left(\varepsilon_{h_1}, \varepsilon_{h_2}, \varepsilon_{h_3}, \ldots, \varepsilon_{h_m}\right), \tag{4}$$

where m denotes the number of flight heights.

When performing digital attack, the patch can be scaled to the corresponding size via the following equation:

$$s_h = \frac{s_p}{\varepsilon_h}. \tag{5}$$

The size of adversarial patch depends on the scale factor. Besides, In consideration of the feasibility in the real world, the patch needs to be placed on the proper location, such as the roof of the car, instead of the windows. Therefore, we need to make annotations to mark the center of car roofs on the experimental data. Figure 3 shows that clean images are transferred to be adversarial examples with apply function. The size of the patch is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

matched to the size of the car roof and the angle between the patch and the car are different after random rotation of the patch.

3.3. Adversarial Patch Optimization

To obtain better attack effect, we propose the combination of detect loss and object loss to optimize adversarial patches. Besides, in order to make the physical attack more effective, we also introduce total variation (TV) loss and non-printability score (NPS) loss.

Object loss. As is analyzed in section 3.1, object confidence denotes the probability of having an object. For an image, detection models usually output a large quantity of candidate bounding boxes, and they are much more than objects, which means there are several bounding boxes for one object. In order to filter out extra boxes, Non-Maximum Suppression (NMS) is used in the detector. Firstly, it can suppress most of the candidate bounding boxes and retain the one that gains the top score through $iouf_{th}$. The final results can be got by suppressing the bounding boxes whose confidence is less than $conf_{th}$. Hence, it is natural to take the confidence of the bounding box as the loss function.

The detection result depends on the bounding box whose confidence is the largest. When attacking an object, the confidence of one bounding box is minimized so that it is less than the confidence threshold, but the other bounding boxes may work. Then, the object is still be detected. Besides, we can know that there may be many objects in a single remote sensing image from Figure 2. Specifically, because of the open view of the air, it is often the case that a great number of objects are clustered in an image captured at a large height, which means the smaller the objects are, the more the number of objects may be. The growth of the number of objects brings more threat and difficulty to minimize the confidence of all objects.

To ensure that more objects can be attacked, we propose to take the average object confidence as the object loss. The average object confidence is computed by all bounding boxes of all the objects in one image and object loss L_{obj} can be defined as:

$$L_{obj} = \frac{1}{M \times N}\left(\sum_{obj}^{M}\sum_{bbox}^{N} confidence\right), \tag{6}$$

where N denotes the number of bounding boxes of one object, M denotes the number of the objects of one image.

Detect loss. We are concentrated on attacking Yolo-V3 and Yolo-V5 in this work. Before introducing detect loss, it is necessary to describe the loss of training detection model. As we all know, for model training, every object of the training set will be labeled so that they all have ground truths. If a bounding box is responsible for an object, I^{obj} will be set to 1. Then, the set of the ground truth can be described as $GT = \left\{gt_i | I_i^{obj} = 1, 1 \le i \le N\right\}$ where $gt_i = (x_i, y_i, w_i, h_i, p_i)$, p_i decides the class of the i^{th} box. Therefore, the loss of optimizing the detection model consists of three parts, namely object loss, bounding box loss and class loss. Object loss can be calculated with the binary cross-entropy ℓ_{BCE}:

$$\ell_{BCE}\left(1, \hat{C}_i\right) = \hat{C}_i \log 1 + \left(1 - \hat{C}_i\right) \log\left(1 - \hat{C}_i\right), \tag{7}$$

$$\begin{aligned}\mathcal{L}_{obj} = &\sum_{i=0}^{N} I_i^{obj} \ell_{BCE}\left(1, \widehat{C}_i\right) - \lambda_{noobj} \\ &\sum_{i=0}^{N}\left(1 - I_i^{obj}\right)\ell_{BCE}\left(1, \widehat{C}_i\right),\end{aligned} \tag{8}$$

where λ_{noobj} is a hyperparameter which penalize the incorrect objectness scores.

Bounding box loss can be calculated with the squared error ℓ_{SE}:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

$$\ell_{SE}(x_i, \widehat{x_i}) = (x_i - \widehat{x_i})^2, \tag{9}$$

$$\ell_{SE}(y_i, \widehat{y_i}) = (y_i - \widehat{y_i})^2, \tag{10}$$

$$\ell_{SE}(w_i, \widehat{w_i}) = \left(\sqrt{w_i} - \sqrt{\widehat{w_i}}\right)^2, \tag{11}$$

$$\ell_{SE}\left(h_i, \widehat{h_i}\right) = \left(\sqrt{h_i} - \sqrt{\widehat{h_i}}\right)^2, \tag{12}$$

$$\begin{aligned} \mathcal{L}_{bbox} = \lambda_{coord} \sum_{i=0}^{N} I_i^{obj} [\ell_{SE}(x_i, \widehat{x_i}) + \ell_{SE}(x_i, \widehat{x_i})] + \\ \lambda_{coord} \sum_{i=0}^{N} I_i^{obj} \left[\ell_{SE}(w_i, \widehat{w_i}) + \ell_{SE}\left(h_i, \widehat{h_i}\right)\right], \end{aligned} \tag{13}$$

where λ_{coord} is a hyperparameter which penalizes bounding boxes

Class loss can be calculated with the binary cross-entropy ℓ_{BCE}:

$$\ell_{BCE}\left(p_i^c, \widehat{p_i^c}\right) = \widehat{p_i^c} log(p_i^c) + \left(1 - \widehat{p_i^c}\right) log\left(1 - \widehat{p_i^c}\right), \tag{14}$$

$$\mathcal{L}_{cls} = \sum_{i=0}^{N} I_i^{obj} \sum_{c \in classes} \ell_{BCE}\left(p_i^c, \widehat{p_i^c}\right). \tag{15}$$

The train loss is the sum of $\mathcal{L}_{obj}$, $\mathcal{L}_{bbox}$ and $\mathcal{L}_{cls}$:

$$Loss_{train} = \alpha \mathcal{L}_{obj} + \beta \mathcal{L}_{bbox} + \gamma \mathcal{L}_{cls}. \tag{16}$$

where α, β, γ are the parameters to balance the weights of three losses.

The detection model will be trained by minimizing the loss. On the contrary, to attack the detection model, we can consider maximizing the loss to degrade the accuracy of the model. Therefore, detect loss L_{det} can be computed by:

$$L_{det} = \frac{1}{Loss_{train}}. \tag{17}$$

Total variation loss and non-printability score loss. In consideration of the adversarial effect in the physical world, we apply the total variation (TV) loss and the non-printability score (NPS) loss in this article, which are used to reduce distortion when the patch is applied in the physical world. TV loss is introduced to make the generated patch more smooth. Without the limitation of TV loss, the perturbation will be easily filtered out so that the attack performance greatly decrease. The TV loss is given by:

$$L_{TV} = \sum_{i,j} \sqrt{(p_{i,j} - p_{i+1,j})^2 + (p_{i,j} - p_{i,j+1})^2}, \tag{18}$$

where i and j denote the pixel coordinate of P.

The digital adversarial patch will be printed by the printer, which has a color space. If the pixel of the adversarial patch is not in the color space, there will be distortion between the digital domain and the physical domain for this pixel. So the NPS loss is introduced to guarantee the pixel color is in the color space of the printer, which reduces the distortion when the adversarial patch is printed. The NPS loss is given by:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

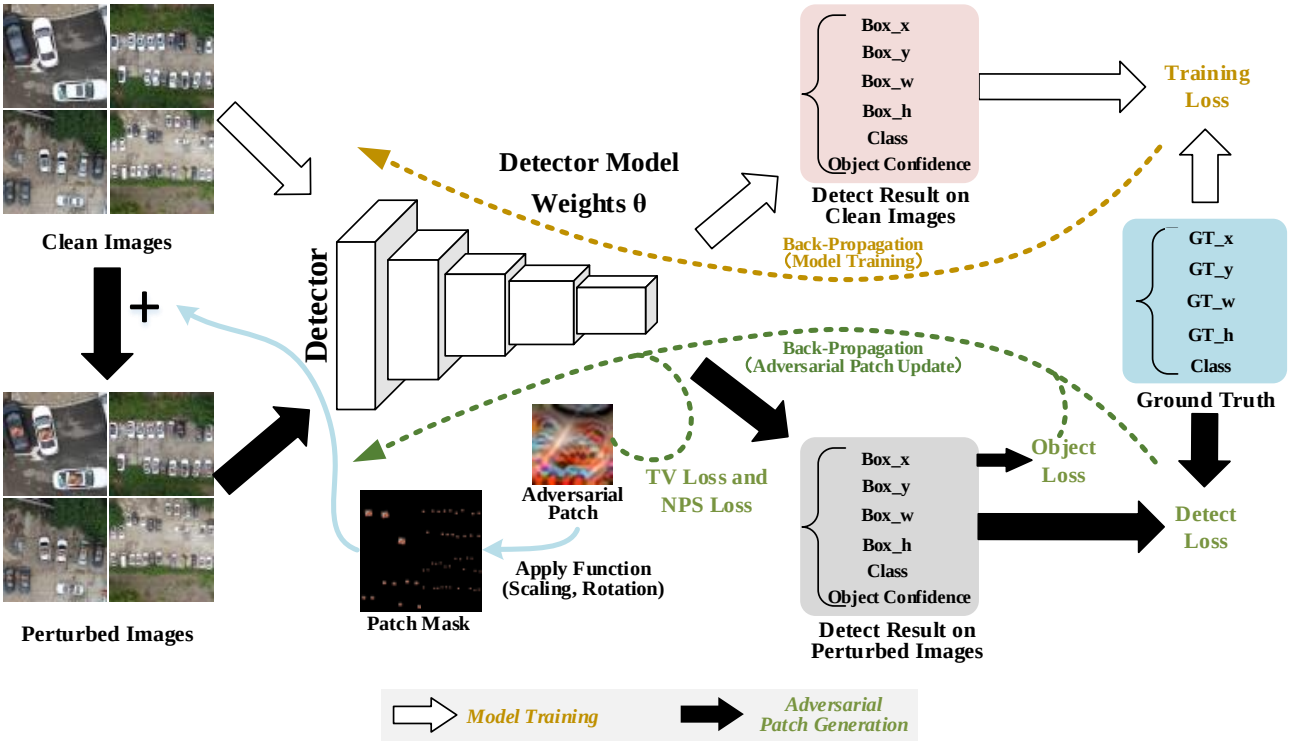


Figure 4. The flowchart of the proposed method. The whole process consists of two parts. Part 1: detection model training. It is an optimization for models' weights. Part2: generating adversarial patch. First, the optimizing adversarial patch is scaled, rotated and attached to clean images, to generate perturbed images with apply function. Second, the adversarial patch is continuously updated through the gradient ascent algorithm to minimize the loss function which is the sum of four parts (L_{obj}, L_{det}, L_{NPS} and L_{TV}).

$$L_{NPS} = \sum_{p_{patch} \in P} \min_{p_{color} \in C} \left| p_{patch} - p_{color} \right|, \tag{19}$$

where P denotes adversarial patch, and p_{patch} is a digital pixel of P. C is the color space of the printer, and p_{color} is an element of C.

The loss function is made up of four parts: object Loss, detect Loss, TV loss and NPS loss. Object Loss contributes to attacking more objects in one image. Detect loss is aimed at reducing the accuracy of the detection model. In this work, we propose the combination of detect loss and object loss which is in favor of better attack effect. The ablation experiment in section 4.4 will verify the effectiveness of this method. The last two losses, TV loss and NPS loss are used to reduce the distortion when the adversarial patch is applied in the physical world. We can get the total loss function as below:

$$L = L_{obj} + \lambda L_{det} + L_{TV} + L_{NPS}. \tag{20}$$

where λ is a hyperparameter to balance the weight of L_{det}.

3.4. The Flowchart of the Proposed Method

Figure 4 shows the flowchart of the proposed method. It can be divided into two parts. First, we need to train a detector with high accuracy. The model training depends on the clean data with object labels. The target models in this paper include Yolo-V3 and Yolo-V5. Owning the trained models, we consider generating adversarial patches on the two detectors. The adversarial patches are added to the clean images with the apply function to create the perturbed images which will be fed into the model. Apply function is devoted to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

attaching the adversarial patch to all the objects. First, a series of transformations (patch rotation, patch rescale) are done on the adversarial patch to generate the patch mask. Then, in order to attach the patch to all the objects, we will replace the pixel of clean image with the pixel in patch mask if the pixel of the patch mask is zero. The update of adversarial patches rely on the gradient ascent algorithm to minimize the loss function which consists of L_{obj}, L_{det}, L_{TV} and L_{NPS}.

4. Experiments

All of our experiments are designed on Yolo-V3 and Yolo-V5, and the remote sensing images we attacked were photographed by a UAV with a wide range of heights from 25m to 120m. First, the experimental setup is presented in Section 4.1. Then, the experimental results of digital attacks are presented to measure the effectiveness of our method in Section 4.2. Third, in Section 4.3, we carry out experiments to show the performance in the physical world of our method. Finally, several ablation studies are done to analyze the influencing factors on the attack effect.

4.1. Experimental Setup

(1) **Data collection.** For the purpose of realizing adversarial attack on multi-scale objects, some data contain multi-scale objects are needed. Naturally, we consider collect data by a UAV in a vertical angle from different heights. In order to collect the data we required, we designed a reasonable scheme for data capturing. First, we choose two scenes as our experimental site, including the street side and car park, where many kinds of cars are often seen. Afterwards, we take DJI Mini 2 as the capturing tool. In our scheme, the flight height ranges from 25 m to 120 m with a height interval of 5m, so there are totally 20 flight heights. The resolution of raw images is 3840×2160. However, the size is too large, which is not suitable for our experiment. Therefore, we need to do some processing on the raw images. They are tailored into a smaller size of 960×960 firstly, and then are resized to 640×640 when being attacked. Ultimately there are 253 training images with 1680 car objects and 186 testing images with 748 car objects. For physical attack, the generated adversarial patches in digital domain are printed and put on the proof of the cars. In the same way, the images in physical experiments are captured at different heights.

(2) **Detectors.** Yolo models are the typical object detectors. With the continuous improvement and renewal, they have got better performance and higher speed for real-time reasoning. Especially for Yolo-V3 and Yolo-V5, they are the popular algorithms which are widely used in object detection. Therefore, we choose Yolo-V3 and Yolo-V5 based on ultralytics as our target models. As for model training, we select visdrone2019 dataset as our training dataset for two reasons, one is that it is a remote sensing dataset, the other is for its high resolution. To satisfy the requirements of our experiments, we have done some adjustments on the dataset, only the car objects were retained, and the others were removed, and then we deleted those images that don't contain car object. Ultimately, there are 6132 training images and 515 testing images left for model training. The input size of two models are fixed in 640×640 when training.

(3) **Baseline methods.** In order to evaluate the effectiveness of our method, several experiments are designed to compare our method with the other methods, including OBJ [29], Dpatch [41] and Patch-Noobj [38], which are all patch attack methods.

(4) **Metrics and implementation details.** Currently, most methods regard the average precision (AP) as the evaluation metrics. However, we think it is not a proper criterion, because high false alarm may also degrade the value of AP, but the true objects can still be detected when the false alarm increase. To develop a better evaluation, we adopt the attack success rate (ASR) metrics as our evaluation metrics. The ASR can be expressed as follows:

$$ASR_\tau = \frac{S(confidence < \tau)}{S_{all}} \tag{21}$$

 doi:10.20944/preprints202210.0131.v1

where $S(confidence < \tau)$ denotes the sum of objects whose confidence is less than the threshold τ. S_{all} denotes the sum of all objects in the test data. We define the object is attacked successfully if there is no detection box on the object. Particularly, Six confidence threshold (τ=0.1, 0.2, 0.3, 0.4, 0.5, 0.6) are selected to show the effectiveness of our method.

For digital attack evaluation, the test data are divided into 5 groups (Group1: 25m-40m, Group2: 45m-60m, Group3: 65m-80m, Group4: 85m-100m, Group5: 105m-120m) in this work and the ASR will be evaluated for the 5 groups. Besides, we also evaluate the total ASR for all the test data. We take the same way to evaluate the physical attack effect.

For implementation details, we set the batch size as 3, the maximum epochs as 200. All the experiments are realized in PyTorch, and our desktop is equipped with an intel core i9-10900X CPU and an Nvidia RTX-3090 GPU. For the sake of fairness, all experiments were carried out in the same conditions.

4.2. Digital Attack

In this section, we perform digital experiments to test ASR for our methods, OBJ [29], DPatch [41] and Patch-Noobj [38] on Yolo-V3 and Yolo-V5. For the sake of fairness, all the experiments for the four methods are carried out at the same condition. For examples, in Dpatch, there is one adversarial patch for one image originally, but we will place the adversarial patch on every object in our experiments, which guarantees the conditions are the same as the other three methods. All of our test data were collected from 25m to 120m in a vertical angle. To prove that our method has a better attack effect for different-scale objects, we divide the test data into 5 groups. In particular, we evaluate the ASR for different scales and compute the total ASR for all test data, and compare our method with the other three methods. Finally, a comprehensive analysis of the evaluation results were conducted.

The experimental results on Yolo-V3 are shown in Table 1. First, the attack effect of every group is evaluated. It is obvious that our method outstands the other methods in most cases. We can see that Patch-Noobj and DPatch show better performance in small-scale objects than large-scale objects from the data of Table 1. Moreover, OBJ performs better on large-scale objects than small-scale objects. Specifically, our method show good performance for all groups. Take the confidence 0.5 as an example, The ASRs of Patch-Noobj and DPatch are 39.33%, 45.27% and 1.33%, 0.68% for the first two groups, which are much less than 96.00% and 93.92% for our method. The loss of OBJ is The overall attack effect of OBJ is better than DPatch and Patch-Noobj, but is still inferior to our methods in the majority of the cases.

When we focus on the different confidence, it can be seen that our method is overwhelming for all confidence. The ASR of our method in confidence 0.1 for all groups is more than 70% for Yolo-V3. As confidence increase, our method keeps on top in most cases. The ASRs of our method wins the first for all the groups in all confidence except Group1 in confidence 0.5. The total ASRs of our method are 79.81%, 85.56%, 88.37%, 90.37%, 92.25%, and 94.12% for 6 different confidence thresholds, which are all the best in the four methods and are head and shoulders above the second best.

Table 2 shows the attack results for Yolo-V5. It can be found that the attack performance for Yolo-V5 is not as good as Yolo-V3 for all the four methods. That may be because the architecture of Yolo-V5 is more robust than Yolo-V3. Nevertheless, our method has a significant advantage compared with the other methods for ASR of different groups and the total ASR under different confidences. It can be seen that DPatch and Patch-Noobj have little effect in small-scale objects. The overall attack effect of OBJ is better than DPatch and Patch-Noobj, but poorer than our method.

Similar to the attack effect of Yolo-V3, As confidence increase, the ASRs of every method are increasing and our method keeps on top in most cases. The total ASRs of our method are 18.32%, 32.35%, 41.84%, 52.27%, 59.49%, and 70.05% for 6 different confidence thresholds, which are all the best in the four methods and are head and shoulders above the second best.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

Table 1. Experimental Results of Different Scales (25m-120m) and Different Confidence (0.1-0.6) for Yolo-V3, Specifically, Red and Blue Indicate Sota and the Second Best.

confidence	method	ASR(%)					
		Group1 (25m-40m)	Group2 (45m-60m)	Group3 (65m-80m)	Group4 (85m-100m)	Group5 (105m-120m)	total
0.1	Raw	0.00	0.00	0.69	0.68	0.00	0.27
	Random Patch	0.00	0.00	1.38	0.68	0.00	0.40
	Dpatch [41]	0.00	0.00	7.59	56.76	47.13	22.59
	OBJ [29]	74.00	55.41	71.72	65.54	41.40	61.36
	Patch-Noobj [38]	17.33	19.59	59.31	75.00	69.43	48.26
	Ours	82.00	**74.32**	**82.76**	**82.43**	**77.71**	**79.81**
0.2	Raw	0.00	0.00	0.69	0.68	0.00	0.27
	Random Patch	0.00	0.00	1.38	1.35	0.64	0.67
	Dpatch	0.00	0.00	9.66	62.16	56.05	25.94
	OBJ	87.33	72.30	77.93	72.30	49.68	72.33
	Patch-Noobj	26.67	27.70	68.28	81.08	76.43	56.15
	Ours	91.33	**82.43**	**88.28**	**85.81**	**80.25**	**85.56**
0.3	Raw	0.00	0.00	1.38	0.68	0.00	0.40
	Random Patch	0.00	0.00	1.38	1.35	0.64	0.67
	Dpatch	0.00	0.00	13.79	68.24	63.06	29.41
	OBJ	88.67	84.46	73.79	70.27	56.69	74.60
	Patch-Noobj	31.33	35.81	77.24	84.46	78.34	61.50
	Ours	91.33	**88.51**	**90.34**	**87.84**	**84.08**	**88.37**
0.4	Raw	0.67	0.00	2.07	0.68	0.00	0.67
	Random Patch	0.67	0.00	2.07	1.35	1.27	1.07
	Dpatch	0.67	0.68	15.86	75.00	70.70	33.02
	OBJ	92.67	83.78	88.28	75.68	60.51	79.95
	Patch-Noobj	34.00	39.86	81.38	84.46	80.89	64.17
	Ours	94.67	**92.57**	**91.72**	**89.19**	**84.08**	**90.37**
0.5	Raw	0.67	0.00	2.07	1.35	0.64	0.94
	Random Patch	0.67	0.68	2.76	2.03	1.91	1.60
	Dpatch	1.33	0.68	19.31	78.38	76.43	35.70
	OBJ	96.67	93.24	88.28	75.68	62.42	83.02
	Patch-Noobj	39.33	45.27	83.45	85.81	82.17	67.25
	Ours	96.00	**93.92**	**94.48**	**91.89**	**85.35**	**92.25**
0.6	Raw	1.33	1.35	2.07	2.03	0.64	1.47
	Random Patch	1.33	1.35	3.45	2.70	1.91	2.14
	Dpatch	2.00	2.70	27.59	82.43	80.25	39.44
	OBJ	96.67	93.24	91.72	81.08	65.61	85.43
	Patch-Noobj	53.33	50.68	85.52	88.51	84.71	72.59
	Ours	97.33	**97.30**	**95.86**	**92.57**	**87.90**	**94.12**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

Table 2. Experimental Results of Different Scales (25m-120m) and Different Confidence (0.1-0.6) for Yolo-V5, Specifically, Red and Blue Indicate Sota and the Second Best.

confidence	method	ASR(%)					
		Group1 (25m-40m)	Group2 (45m-60m)	Group3 (65m-80m)	Group4 (85m-100m)	Group5 (105m-120m)	total
0.1	Raw	0.00	0.00	0.69	0.68	0.00	0.27
	Random Patch	0.00	0.00	1.38	0.68	0.00	0.40
	Dpatch [41]	0.00	0.00	0.00	0.00	2.55	0.53
	OBJ [29]	**10.00**	2.70	4.14	4.05	18.47	8.02
	Patch-Noobj [38]	0.00	0.00	0.69	3.38	2.55	1.34
	Ours	9.33	**9.46**	**14.48**	**27.03**	**30.57**	**18.32**
0.2	Raw	0.00	0.00	1.38	1.35	0.00	0.53
	Random Patch	0.00	0.00	2.07	2.03	0.64	0.94
	Dpatch	0.00	0.00	0.00	2.03	5.73	1.60
	OBJ	19.33	8.11	13.10	12.16	31.85	17.11
	Patch-Noobj	0.00	0.68	1.38	6.08	10.83	3.88
	Ours	**22.67**	**20.95**	**26.90**	**36.49**	**53.50**	**32.35**
0.3	Raw	0.00	0.00	1.38	2.03	0.00	0.67
	Random Patch	0.00	0.00	2.07	2.03	1.27	1.07
	Dpatch	0.00	0.00	0.00	2.70	10.83	2.81
	OBJ	25.33	16.22	17.93	22.30	43.31	25.27
	Patch-Noobj	0.00	0.68	2.76	10.81	25.48	8.16
	Ours	**34.00**	**30.41**	**32.41**	**44.59**	**66.24**	**41.84**
0.4	Raw	0.00	0.00	2.07	2.03	0.64	0.94
	Random Patch	0.00	0.00	2.07	2.70	1.91	1.34
	Dpatch	0.00	0.00	0.00	4.73	14.65	4.01
	OBJ	38.67	22.30	24.83	27.03	56.69	34.22
	Patch-Noobj	0.00	0.68	2.76	14.19	32.48	10.29
	Ours	**41.33**	**40.54**	**48.97**	**53.38**	**75.80**	**52.27**
0.5	Raw	0.00	0.00	2.07	2.70	1.27	1.20
	Random Patch	0.00	0.00	3.45	2.70	1.27	1.47
	Dpatch	0.00	0.00	2.07	10.14	19.11	6.42
	OBJ	**48.67**	33.11	30.34	39.86	65.61	43.85
	Patch-Noobj	0.00	2.03	6.90	24.32	45.22	16.04
	Ours	48.00	**49.32**	**57.93**	**58.78**	**82.17**	**59.49**
0.6	Raw	0.00	0.68	2.07	2.70	1.27	1.34
	Random Patch	0.00	0.68	2.76	3.38	1.27	1.60
	Dpatch	0.00	0.00	6.21	19.59	29.94	11.36
	OBJ	60.67	47.30	44.83	52.70	79.62	57.35
	Patch-Noobj	0.00	5.41	13.10	37.84	56.69	22.99
	Ours	**61.33**	**64.19**	**65.52**	**71.62**	**86.62**	**70.05**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

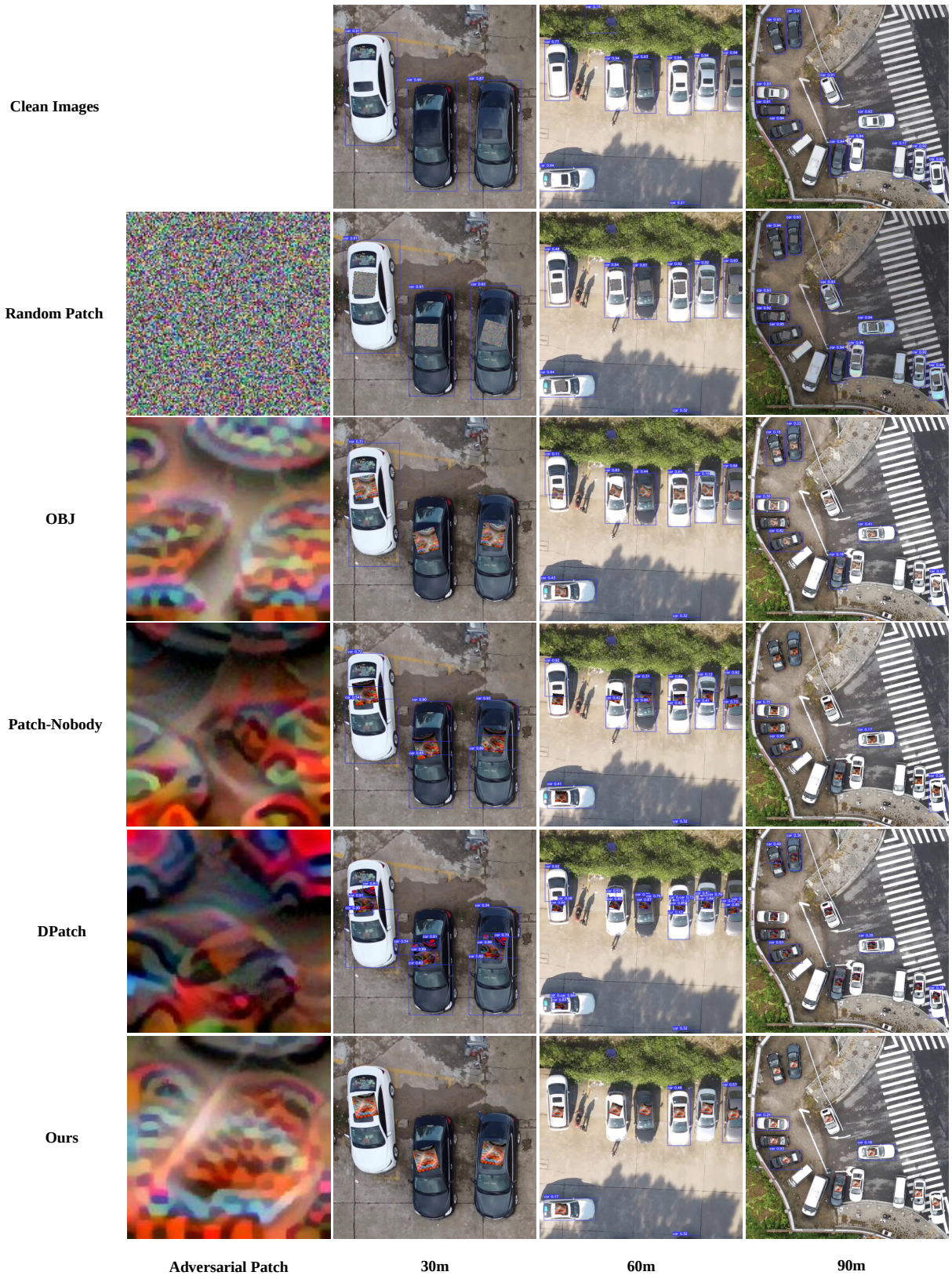


Figure 5. Comparison of the attack effects for our methods, Dpatch, OBJ, Patch-Noobj and random patch for Yolo-V3. Images shown in this figure are captured at the height of 30m, 60m and 90m.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

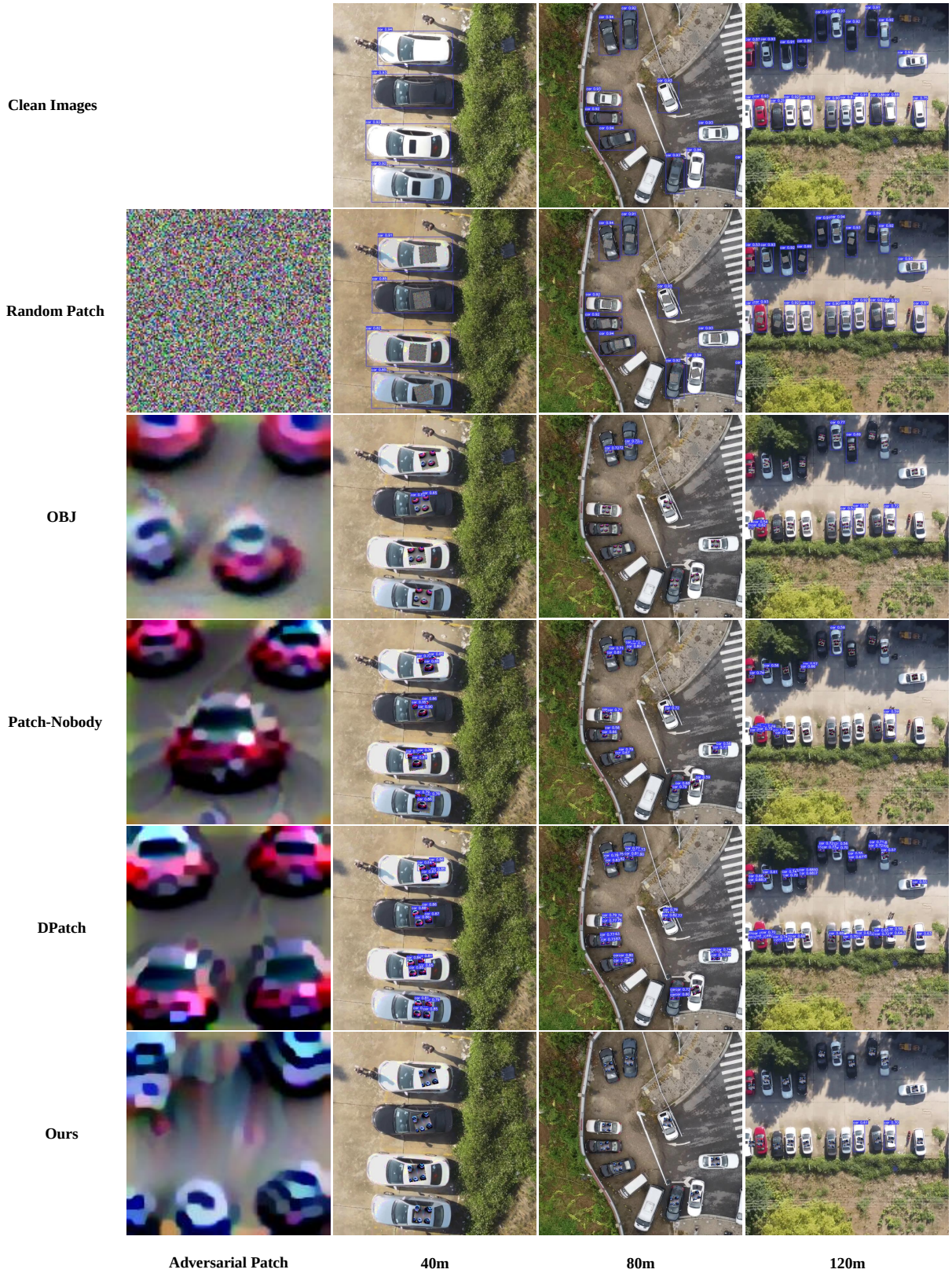


Figure 6. Comparison of the attack effects for our methods, Dpatch, OBJ, Patch-Noobj and random patch for Yolo-V5. Images shown in this figure are captured at the height of 30m, 60m and 90m.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

The conclusion can be drawn from Table 1 and Table 2 that the performance of our method are the best in majority of the cases and the second best in the other cases, which demonstrate the superiority of our method for multi-scale objects attack.

Besides, we provide visualizations of the adversarial patch and attack results of different methods in Figure 5 and Figure 6. Figure 5 shows the attack results for Yolo-V3 and Figure 6 shows the attack results for Yolo-V5. For Yolo-V3, images captured from 30m, 90m and 120m are selected and for Yolo-V5, images captured from 40m, 80m and 120m are selected to show the attack effect. It can be observed that our adversarial patch attack more objects than the other three methods. DPatch and Patch-Noobj show better performance in small objects than large objects. Observing the texture of the adversarial patch, we may find that the patch texture of DPatch and Patch-Noobj seem to be cars. Therefore, if the scale of patch is large, the texture will be easily detected as cars, which lead that the object can't vanish thoroughly. For our method, there is no texture about car in the adversarial patch, and owning to the joint of detect loss and object loss in our loss function, the attack effect is better than DPatch, Patch-Noobj and OBJ. We may find the texture between Yolo-V3 and Yolo-V5 are different, which proves the texture also depends on the architecture of the detection model. In spite of this, there is a similar conclusion about the attack effect for Yolo-V3 and Yolo-V5.

4.3. Physical Attack

The physical attack results of our method on Yolo-V3 and Yolo-V5 will be shown in this section. In view of the actual size of the car, the generated adversarial patches are printed with a size of 1.1m$\times$1.1m. When carrying out the physical attack, the adversarial patches are placed on the roofs of cars. We take the DJI Mini2 to capture videos that contain cars covered with adversarial patches from 20m to 120m. Similar to the strategy of digital attack, we divide these data into five groups (Group1: 20m-40m, Group2: 40m-60m, Group3: 60m-80m, Group4: 80m-100m, Group5: 100m-120m).

Table 3. The numbers of objects with adversarial patch for Yolo-V3 and Yolo-V5.

Model	ASR(%)					
	Group1 (25m-40m)	Group2 (45m-60m)	Group3 (65m-80m)	Group4 (85m-100m)	Group5 (105m-120m)	total
Yolo-V3	1780	1770	1840	1600	1750	8740
Yolo-V5	1540	1470	1500	1500	1530	7540

Table 4. Physical attack results on Yolo-V3 and Yolo-V5.

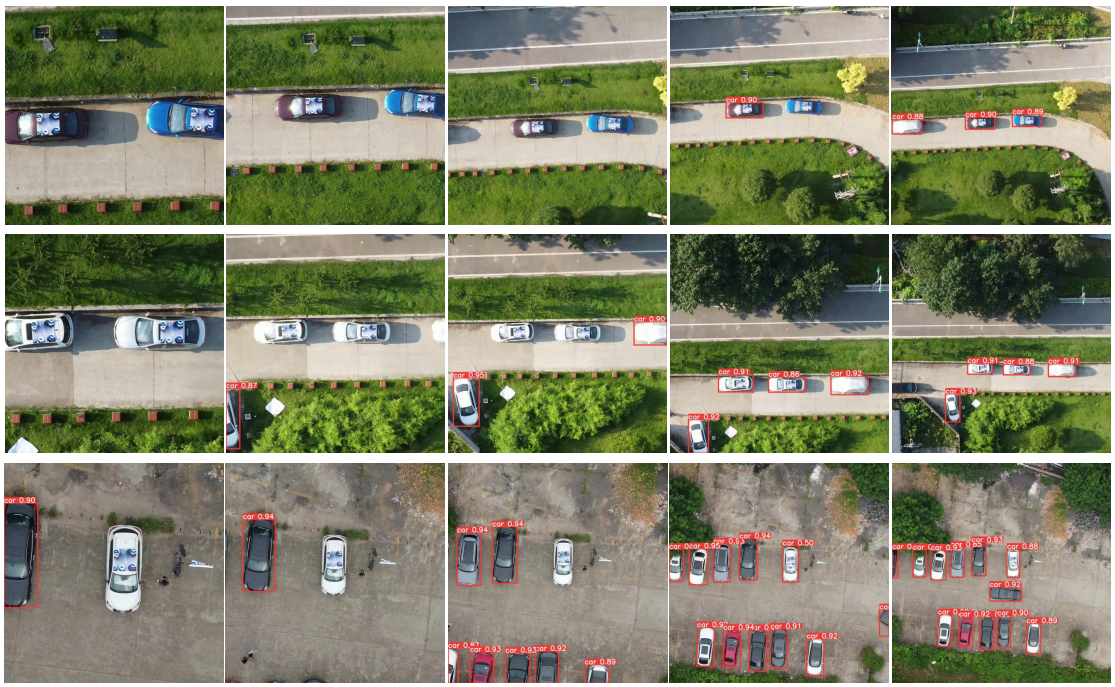
Model	ASR(%)					
	Group1 (25m-40m)	Group2 (45m-60m)	Group3 (65m-80m)	Group4 (85m-100m)	Group5 (105m-120m)	total
Yolo-V3	0.45	0.73	0.60	0.24	0.12	0.43
Yolo-V5	0.44	0.50	0.50	0.09	0.00	0.30

We further preprocess the captured data. Firstly, the videos should be cut according to the flight height. Then, the videos that have been cut will be sent to the detection model to output the detection result. In order to better evaluate the physical attack effect, we process the videos that have been detected as frame-by-frame images. For physical attack, we take ASR as the evaluation metrics and the confidence threshold is set to 0.5. In our experiments,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

Group1 Group2 Group3 Group4 Group5

Figure 7. Selected examples of Physical attack on Yolo-V3. These examples are from three scenes and the images in the same row are form the same scene. For every scene, we select five images from five groups.



Group1 Group2 Group3 Group4 Group5

Figure 8. Selected examples of Physical attack on Yolo-V5. These examples are from three scenes and the images in the same row are form the same scene. For every scene, we select five images from five groups.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

only the objects covered with adversarial patches are valid for computing the ASR. Table 3 shows the number of objects in every group for Yolo-V3 and Yolo-V5.

The physical attack results are shown in Table 4. It can be observed that the ASRs of Group1, Group2 and Group3 are much higher than that of Group4 and Group5 for Yolo-V3 and Yolo-V5. This results may attribute to image degradation. In real scenarios, the low-resolution images will suffer from some complex process of degradations, which usually come from complicated combinations of degradation processes, such as imaging system of cameras, image editing, and Internet transmission. Therefore, when we capture an image, there will be some distortion in this image and the object in this image will appear to be different from the real object. Similarly, when carrying out the experiments of physical attack, as the resolution of adversarial patch increasing, the adversarial patch also suffer from the degradation of the real world, which will lead to bad performance for physical attack.

Some selected examples of physical attacks on Yolo-V3 and Yolo-V5 are presented in Figure 7 and Figure 8. Figure 7 shows several images of three scenes for Yolo-V3 and Figure 8 shows several images of three scenes for Yolo-V5. The confidence threshold is set 0.5 in this experiment. It can be observed that the attack results for Group1, Group2 and Group3 are better than that of Group4 and Group5. Moreover, we find that the adversarial patches suffer from serious degradation as the size of object decreasing, which is the reason for the poor attack result.

4.4. Ablation Studies

In this section, the influence of detect loss and object loss on attack effect will be studied by carrying out several experiments for ablation studies.

Table 5. Experimental results about different loss items for Yolo-V3, bold fonts indicate the best effect.

Confidence	Loss	ASR(%)					
		Group1 (25m-40m)	Group2 (45m-60m)	Group3 (65m-80m)	Group4 (85m-100m)	Group5 (105m-120m)	total
0.1	**Object Loss**	74.00	65.54	74.48	80.41	71.97	73.26
	Detect Loss	30.67	56.08	69.66	75.68	63.06	58.96
	Object Loss +0.001×Detect Loss	68.67	62.16	68.97	77.70	73.89	70.32
	Object Loss +0.005×Detect Loss	80.00	65.54	80.00	80.41	67.52	74.60
	Object Loss +0.01×Detect Loss	77.33	63.51	73.10	75.00	67.52	71.26
	Object Loss +0.02×Detect Loss	**82.00**	**74.32**	**82.76**	**82.43**	**77.71**	**79.81**
	Object Loss +0.04×Detect Loss	70.67	58.78	74.48	77.70	72.61	70.86
0.3	**Object Loss**	90.00	83.11	84.83	88.51	79.62	85.16
	Detect Loss	51.33	74.32	84.83	83.78	75.80	73.93
	Object Loss +0.001×Detect Loss	84.00	75.00	82.76	84.46	80.89	81.42
	Object Loss +0.005×Detect Loss	**93.33**	83.78	87.59	83.78	77.71	85.16
	Object Loss +0.01×Detect Loss	90.00	82.43	84.14	83.11	80.89	84.09
	Object Loss +0.02×Detect Loss	91.33	**88.51**	**90.34**	**87.84**	**84.08**	**88.37**
	Object Loss +0.04×Detect Loss	83.33	77.70	88.28	83.78	73.25	81.15
0.5	**Object Loss**	92.67	92.57	89.66	91.22	82.17	89.57
	Detect Loss	74.00	85.14	87.59	88.51	83.44	83.69
	Object Loss +0.001×Detect Loss	93.33	89.86	91.03	87.16	85.99	89.44
	Object Loss +0.005×Detect Loss	95.33	93.24	88.28	88.51	83.44	89.71
	Object Loss +0.01×Detect Loss	95.33	91.22	91.03	85.81	84.71	89.57
	Object Loss +0.02×Detect Loss	**96.00**	**93.92**	**94.48**	**91.89**	**85.35**	**92.25**
	Object Loss +0.04×Detect Loss	90.00	87.84	92.41	89.86	86.62	89.30

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2022 doi:10.20944/preprints202210.0131.v1

Table 6. Experimental results about different loss items for Yolo-V5, bold fonts indicate the best effect.

Confidence	Loss	ASR(%)					
		Group1 (25m-40m)	Group2 (45m-60m)	Group3 (65m-80m)	Group4 (85m-100m)	Group5 (105m-120m)	total
0.1	Object Loss	5.33	8.78	6.90	10.14	19.11	10.16
	Detect Loss	0.00	0.00	0.00	0.00	11.46	2.41
	Object Loss +0.001×Detect Loss	11.33	8.78	13.79	15.54	23.57	14.71
	Object Loss +0.005×Detect Loss	9.33	**9.46**	**14.48**	**27.03**	**30.57**	**18.32**
	Object Loss +0.01×Detect Loss	**13.33**	4.73	6.90	8.78	14.01	9.63
	Object Loss +0.02×Detect Loss	8.00	5.41	8.28	12.84	14.65	9.89
	Object Loss +0.04×Detect Loss	9.33	3.38	2.76	6.08	10.83	6.55
0.3	Object Loss	19.33	25.68	28.97	35.81	49.04	31.95
	Detect Loss	0.00	0.00	0.69	6.08	25.48	6.68
	Object Loss +0.001×Detect Loss	30.67	27.03	33.10	35.14	56.69	36.76
	Object Loss +0.005×Detect Loss	34.00	**30.41**	**32.41**	**44.59**	**66.24**	**41.84**
	Object Loss +0.01×Detect Loss	**40.00**	21.62	24.83	29.05	34.39	30.08
	Object Loss +0.02×Detect Loss	27.33	24.32	26.21	33.78	36.31	29.68
	Object Loss +0.04×Detect Loss	37.33	16.22	13.79	22.30	29.94	24.06
0.5	Object Loss	38.00	37.84	44.14	50.68	70.06	48.40
	Detect Loss	0.00	1.35	1.38	20.95	42.04	13.50
	Object Loss +0.001×Detect Loss	56.00	42.57	52.41	56.76	79.62	57.75
	Object Loss +0.005×Detect Loss	48.00	**49.32**	**57.93**	**58.78**	**82.17**	**59.49**
	Object Loss +0.01×Detect Loss	**60.67**	43.92	42.76	50.00	63.06	52.27
	Object Loss +0.02×Detect Loss	53.33	44.59	37.93	43.24	57.32	47.46
	Object Loss +0.04×Detect Loss	59.33	34.46	33.79	43.24	49.04	44.12

Table 7. the comparison of attack effect on AP, Precision and Recall, bold fonts indicate the best effect.

Model	Loss	AP(%)	Precision	Recall
Yolo-V3	Raw	99.0	99.0	97.3
	Object Loss	65.7	68.0	61.9
	Detect Loss	**29.6**	**31.3**	**39.8**
	Detect Loss+Object Loss	59.7	58.2	59.3
Yolo-V5	Raw	99.1	99.4	96.6
	Object Loss	46.1	42.0	44.2
	Detect Loss	**2.83**	**4.77**	**30.7**
	Detect Loss+Object Loss	36.0	32.9	46.1

Table 5 and Table 6 show the attack results with different combinations of detect loss and object loss for Yolo-V3 and Yolo-V5. It can be observed that the combination of detect loss and object loss can obtain a better attack effect. From the perspective of experimental results, we may find that the object loss plays a major role in the ASR. However, with the help of detect loss, higher ASR may be gained for different scales. Moreover, we explored the influence of different values of λ for detect loss. It is found that the best λ is 0.02 for Yolo-V3 and the best λ is 0.005 for Yolo-V5.

The aim of object loss is to decrease the confidence of all the objects in one image, and detect loss contributes more to degrading the accuracy of detection model. Table 7 shows the attack effect on AP, precision and recall, which are evaluation metrics for detection

model. It is easy to find that detect loss has the lowest scores for the three parameters whether for Yolo-V3 or Yolo-V5, which proves the view that detect loss play an important role in decrease the accuracy of model. It can be concluded that the object loss may work better with the degradation of model's accuracy.

5. Conclusion

In this study, we analyze the physical attack on multi-scale objects in remote sensing data for the first time. First, we formulate a joint optimization problem, in which object loss and detect loss are introduced, to generate a universal adversarial patch. Object loss which denotes the average confidence of all bounding boxes in one image, contributes to attack as many objects as possible. Detect loss between the detection results and the ground truth, aims at degrading the accuracy of a detector. Experimental results demonstrate the effectiveness of the combination of the two losses. Besides, we raise a scale factor to make the scale of the adversarial patch adapts to the size of object in digital attack, which ensures the adversarial patch is valid for multi-scale objects in the real world. Those images for adversarial attack are captured from 25m to 120m. In experiments, we divide the test data into five groups based on the height label of images. Several digital attack experiments for Yolo-V3 and Yolo-V5 are carried out, and we compute the ASR in six confidence (0.1, 0.2, 0.3, 0.4, 0.5, and 0.6). For ASR of different scales, our methods outperforms state-of-the-art methods in most cases. For the total ASR, our method also outstands the other methods for every confidence. Besides, the generated adversarial patch of our method is printed, and we perform physical attack experiments to verify the attack effect for different photography heights.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z. and P.Z.; software, Y.Z.; validation, Y.Z.; formal analysis, Y.Z.; investigation, Y.Z.; resources, P.Z.; writing–original draft preparation, Y.Z.; writing—review and editing, Y.Z., Y.Z., J.Q., K.B., H.W. and P.Z.; visualization, Y.Z.; supervision, P.Z.; project administration, P.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of China under Grant 61971428.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, R.; Kuffer, M.; Persello, C. The Temporal Dynamics of Slums Employing a CNN-Based Change Detection Approach. *Remote Sens.* **2019**, *11*, 2844. [CrossRef]
2. Peng, B.; Meng, Z.; Huang, Q.; Wang, C. Patch Similarity Convolutional Neural Network for Urban Flood Extent Mapping Using Bi-Temporal Satellite Multispectral Imagery. *Remote Sens.* **2019**, *11*, 2492. [CrossRef]
3. Zhang, X.; Han, L.; Dong, Y.; Shi, Y.; Huang, W.; Han, L.; González-Moreno, P.; Ma, H.; Ye, H.; Sobeih, T. A Deep Learning-Based Approach for Automated Yellow Rust Disease Detection from High-Resolution Hyperspectral UAV Images. *Remote Sens.* **2019**, *11*, 1554. [CrossRef]
4. Liu, H.; Li, J.; He, L.; Wang, Y. Superpixel-Guided Layer-Wise Embedding CNN for Remote Sensing Image Classification. *Remote Sens.* **2019**, *11*, 174. [CrossRef]
5. Matos-Carvalho, J.P.; Moutinho, F.; Salvado, A.B.; Carrasqueira, T.; Campos-Rebelo, R.; Pedro, D.; Campos, L.M.; Fonseca, J.M.; Mora, A. Static and Dynamic Algorithms for Terrain Classification in UAV Aerial Imagery. *Remote Sens.* **2019**, *11*, 2501. [CrossRef]
6. Guan, Z.; Miao, X.; Mu, Y.; Sun, Q.; Ye, Q.; Gao, D. Forest Fire Segmentation from Aerial Imagery Data Using an Improved Instance Segmentation Model. *Remote Sens.* **2022**, *14*, 3159. [CrossRef]
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.

9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 779–788.
10. Redmon, J.; Farhadi, A. Yolov3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
11. Dong, X.; Qin, Y.; Gao, Y.; Fu, R.; Liu, S.; Ye, Y. Attention-Based Multi-Level Feature Fusion for Object Detection in Remote Sensing Images. *Remote Sens.* **2022**, *14*, 3735. [CrossRef]
12. Zhao, Y.; Li, J.; Li, W.; Shan, P.; Wang, X.; Li, L.; Fu, Q. MS-IAF: Multi-Scale Information Augmentation Framework for Aircraft Detection. *Remote Sens.* **2022**, *14*, 3696. [CrossRef]
13. Zheng, Y.; Sun, P.; Zhou, Z.; Xu, W.; Ren, Q. ADT-Det: Adaptive Dynamic Refined Single-Stage Transformer Detector for Arbitrary-Oriented Object Detection in Satellite Optical Imagery. *Remote Sens.* **2021**, *13*, 2623. [CrossRef]
14. Mohamed, A.r.; Dahl, G.E.; Hinton, G. Acoustic Modeling Using Deep Belief Networks. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *20*, 14–22. [CrossRef]
15. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification With Deep Convolutional Neural Networks. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NeurIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
17. Qin, H.; Cai, Z.; Zhang, M.; Ding, Y.; Zhao, H.; Yi, S.; Liu, X.; Su, H. Bipointnet: Binary Neural Network for Point Clouds. *arXiv* **2020**, arXiv:2010.05501.
18. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing Properties of Neural Networks. *arXiv* **2013**, arXiv:1312.6199.
19. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572.
20. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv* **2017**, arXiv:1706.06083.
21. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
22. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal Adversarial Perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1765–1773.
23. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 39–57.
24. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrücken, Germany, 21–24 March 2016; pp. 372–387.
25. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-art Face Recognition. In Proceedings of the 2016 ACM Sigsac Conference on Computer And Communications Security (CCS), Vienna, Austria, 24-28 October 2016; pp. 1528–1540.
26. Xu, K.; Zhang, G.; Liu, S.; Fan, Q.; Sun, M.; Chen, H.; Chen, P.Y.; Wang, Y.; Lin, X. Adversarial T-Shirt! Evading Person Detectors in a Physical World. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020, pp. 665–681.
27. Brown, T.B.; Mané, D.; Roy, A.; Abadi, M.; Gilmer, J. Adversarial Patch. *arXiv* **2017**, arXiv:1712.09665.
28. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the Physical World. In Proceedings of the Workshop of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
29. Thys, S.; Van Ranst, W.; Goedemé, T. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 0–0.
30. Hu, Z.; Huang, S.; Zhu, X.; Sun, F.; Zhang, B.; Hu, X. Adversarial Texture for Fooling Person Detectors in the Physical World. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19-24 June 2022; pp. 13307–13316.
31. Xu, Y.; Du, B.; Zhang, L. Assessing the Threat of Adversarial Examples on Deep Neural Networks for Remote Sensing Scene Classification: Attacks and Defenses. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 1604–1617. [CrossRef]
32. Chan-Hon-Tong, A.; Lenczner, G.; Plyer, A. Demotivate Adversarial Defense in Remote Sensing. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). Brussels, Belgium, 11-16, July, 2021; pp. 3448–3451.
33. Chen, L.; Zhu, G.; Li, Q.; Li, H. Adversarial Example in Remote Sensing Image Recognition. *arXiv* **2019**, arXiv:1910.13222.
34. Czaja, W.; Fendley, N.; Pekala, M.; Ratto, C.; Wang, I.J. Adversarial Examples in Remote Sensing. In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2018), Seattle, WA, USA, 6–9 November 2018; pp. 408–411.
35. Xu, Y.; Ghamisi, P. Universal Adversarial Examples in Remote Sensing: Methodology and Benchmark. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]

36. Chen, L.; Xu, Z.; Li, Q.; Peng, J.; Wang, S.; Li, H. An Empirical Study of Adversarial Examples on Remote Sensing Image Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7419–7433. [CrossRef]
37. Xu, Y.; Du, B.; Zhang, L. Self-Attention Context Network: Addressing the Threat of Adversarial Attacks for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2021**, *30*, 8671–8685. [CrossRef] [PubMed]
38. Lu, M.; Li, Q.; Chen, L.; Li, H. Scale-Adaptive Adversarial Patch Attack for Remote Sensing Image Aircraft Detection. *Remote Sens.* **2021**, *13*, 4078. [CrossRef]
39. Du, A.; Chen, B.; Chin, T.J.; Law, Y.W.; Sasdelli, M.; Rajasegaran, R.; Campbell, D. Physical Adversarial Attacks on an Aerial Imagery Object Detector. In Proceedings of the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, 4-8 January 2022; pp. 1796–1806.
40. Chow, K.H.; Liu, L.; Gursoy, M.E.; Truex, S.; Wei, W.; Wu, Y. TOG: Targeted Adversarial Objectness Gradient Attacks on Real-Time Object Detection Systems. *arXiv* **2020**, arXiv:2004.04320.
41. Liu, X.; Yang, H.; Liu, Z.; Song, L.; Li, H.; Chen, Y. Dpatch: An Adversarial Patch Attack on Object Detectors. *arXiv* **2018**, arXiv:1806.02299.
42. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing Robust Adversarial Examples. In Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 284–293.
43. Evtimov, I.; Eykholt, K.; Fernandes, E.; Kohno, T.; Li, B.; Prakash, A.; Rahmati, A.; Song, D. Robust Physical-World Attacks on Deep Learning Models. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
44. Chen, S.T.; Cornelius, C.; Martin, J.; Chau, D.H.P. Shapeshifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), Dublin, Ireland, 10–14 September 2018; pp. 52–68.
45. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Song, D.; Kohno, T.; Rahmati, A.; Prakash, A.; Tramer, F. Note on Attacking Object Detectors with Adversarial Stickers. *arXiv* **2017**, arXiv:1712.08062.
46. Lu, J.; Sibai, H.; Fabry, E. Adversarial Examples That Fool Detectors. *arXiv* **2017**, arXiv:1712.02494.
47. Song, D.; Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Tramer, F.; Prakash, A.; Kohno, T. Physical Adversarial Examples for Object Detectors. In Proceedings of the 12th USENIX Workshop on Offensive Technologies (WOOT 2018), co-located with USENIX Security 2018, Baltimore, MD, USA, 13–14 August 2018.
48. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
49. Wang, Y.; Lv, H.; Kuang, X.; Zhao, G.; Tan, Y.a.; Zhang, Q.; Hu, J. Towards a Physical-World Adversarial Patch for Blinding Object Detection Models. *Inf. Sci.* **2021**, *556*, 459–471. [CrossRef]
50. Wang, J.; Liu, A.; Yin, Z.; Liu, S.; Tang, S.; Liu, X. Dual Attention Suppression Attack: Generate Adversarial Camouflage in Physical World. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8565–8574.