remote sensing

MDPI

Article

Scale-Adaptive Adversarial Patch Attack for Remote Sensing Image Aircraft Detection

Mingming Lu [1], Qi Li [1], Li Chen [2,*] and Haifeng Li [2]

[1] School of Computer Science and Engineering, Central South University, South Lushan Road, Changsha 410083, China; mingminglu@csu.edu.cn (M.L.); dsjliqi@csu.edu.cn (Q.L.)
[2] School of Geosciences and Info-Physics, Central South University, South Lushan Road, Changsha 410083, China; lihaifeng@csu.edu.cn
* Correspondence: vchenli@csu.edu.cn; Tel.:+86-152-7313-7420

Abstract: With the adversarial attack of convolutional neural networks (CNNs), we are able to generate adversarial patches to make an aircraft undetectable by object detectors instead of covering the aircraft with large camouflage nets. However, aircraft in remote sensing images (RSIs) have the problem of large variations in scale, which can easily cause size mismatches between an adversarial patch and an aircraft. A small adversarial patch has no attack effect on large aircraft, and a large adversarial patch will completely cover small aircraft so that it is impossible to judge whether the adversarial patch has an attack effect. Therefore, we propose the adversarial attack method Patch-Noobj for the problem of large-scale variation in aircraft in RSIs. Patch-Noobj adaptively scales the width and height of the adversarial patch according to the size of the attacked aircraft and generates a universal adversarial patch that can attack aircraft of different sizes. In the experiment, we use the YOLOv3 detector to verify the effectiveness of Patch-Noobj on multiple datasets. The experimental results demonstrate that our universal adversarial patches are well adapted to aircraft of different sizes on multiple datasets and effectively reduce the Average Precision (AP) of the YOLOv3 detector on the DOTA, NWPU VHR-10, and RSOD datasets by 48.2%, 23.9%, and 20.2%, respectively. Moreover, the universal adversarial patch generated on one dataset is also effective in attacking aircraft on the remaining two datasets, while the adversarial patch generated on YOLOv3 is also effective in attacking YOLOv5 and Faster R-CNN, which demonstrates the attack transferability of the adversarial patch.

Keywords: adversarial patch; adversarial example; object detector; remote sensing image (RSI) object detection

Citation: Lu, M.; Li, Q.; Chen, L.; Li, H. Scale-Adaptive Adversarial Patch Attack for Remote Sensing Image Aircraft Detection. *Remote Sens.* **2021**, *13*, 4078. https://doi.org/10.3390/rs13204078

Academic Editor: Francesco Nex

Received: 24 August 2021
Accepted: 7 October 2021
Published: 12 October 2021

1. Introduction

Among the objects in RSIs, an aircraft is considered a typical civil and military object. It has a wide range of types and scale variations and has an important role in transportation, air surveillance, etc. In recent years, object detection algorithms based on CNNs have achieved remarkable success in tasks, such as aircraft detection [1–5]. Adversarial attacks on object detectors have also received extensive attention [6–11]. Adversarial patch attacks on object detectors (unlike traditional camouflage that evades detection of object detectors by placing camouflage nets to cover important objects) are being explored to achieve concealment of important objects, such as aircraft (simple production method and low production cost), by deceiving object detectors and guiding them to make incorrect decisions [12–14].

Currently, research on adversarial patch attacks in terms of object detectors is mainly conducted on natural images [12–15]. These attack methods usually generate a fixed-size adversarial patch to attack an object detector. Compared with natural images, objects in RSIs encounter problems, such as complex backgrounds, a wide variety of types, and large-scale variations. In RSIs, the same category of objects of different types may have different

sizes, and the same objects will have different sizes when images are captured with satellites in different orbits and drones at different altitudes above the ground. However, the current attack method for generating a fixed-size adversarial patch for natural images disregards the case of varying object sizes, and such methods are not applicable to RSIs with large variations in object scales. Therefore, to implement the adversarial patch attack on object detectors for RSIs, the difficulties posed by varying object sizes for generating adversarial patches need to be addressed.

In addition to addressing the difficulties posed by the large variation in object scales for generating adversarial patches, the attack strategy of adversarial patches needs to be considered, i.e., the adversarial patch whether makes the detector misclassify a specific object or makes the specific objects disappear from the detector's view. The attack strategy affects the setting of the objective function for optimizing the adversarial patch. Consider the vanishing attack as an example. The adversarial patch obtained by some objective function's optimization has a large visual difference from the attacked object, which can well cause the specific object to evade detection by the object detector. While the adversarial patch obtained by some objective function's optimization has a certain visual similarity with the attacked object, it is possible to detect the object of attacked category in the adversarial patch [13,15].

In this paper, we focus on attacking object detectors on RSIs, allowing the adversarial patches to replace the traditional large camouflage nets to disguise the aircraft (make the aircraft vanish). Based on large variations in object scale and attack strategy, we propose an adversarial attack method named as Patch-Noobj that generates a universal adversarial patch to make the aircraft vanish from the view of object detectors. This method randomly initializes an adversarial patch of fixed size, uses the confidence loss of the bounding box as the objective function to optimize the adversarial patch, and then reduces the confidence of the bounding box to zero as much as possible by gradient descent so that the bounding boxes containing the aircraft are filtered out. To adapt the adversarial patch to the scale change of the aircraft, we adaptively scale the width and height of the initial adversarial patch according to the size of the aircraft in the process of generating the adversarial patch.

To evaluate our attack method, we constructed a series of experiments on YOLOv3 [16], YOLOv5 [17], and Faster R-CNN [18] detectors. The results of the experiments show that our attack method can effectively prevent object detectors from detecting aircraft. Our method can reduce the Average Precision (AP) of YOLOv3 in detecting aircraft from 93.1% to 44.9% on the DOTA dataset [19], and it has a similar attack effect on the NWPU VHR-10 [20] and RSOD datasets [21]. Moreover, the attack transferability experiments show that the adversarial patches generated by our method are able to transfer among the three datasets and the three models of YOLOv3, YOLOv5, and Faster R-CNN. Additionally, we discovered the relationship between the size of the adversarial patch and the attack performance by exploring the effect of the size of the adversarial patch on the attack performance. In addition, with the help of Grad-CAM [22], which is an interpretability method for image classification tasks, we analyze why the adversarial patch can attack the object detector. The contribution of this work can be summarized in the following three points:

1. We propose an adversarial attack method for aircraft detection in RSIs, which hides the decision features of the aircraft in the object detector and reduces the confidence of the bounding box in the detector to a lower level than the threshold, thus misleading the detection results of the detector.
2. Our proposed adversarial attack method has the characteristic of adversarial patch size adaption, which can adapt to the variation of aircraft scale in RSIs and effectively attack object detectors.
3. The adversarial patches generated by our proposed attack method have attack transferability between different datasets and models.

2. Materials and Methods

2.1. Related Work

The adversarial example phenomenon was first discovered on the image classification task [23]. Among the current adversarial example attack methods, the most straightforward attack method is to add a small global perturbation that is imperceptible to humans to the image, thus deceiving the CNNs into making incorrect predictions [24–27]. The attack method of adding a global perturbation achieves good attack performance in the digital space; however, for the physical world, such an attack method is unlikely to be realized because we cannot add the global adversarial perturbation to the physical world image. We can only print the adversarial examples generated in the digital space and deploy them to the physical world [25]. Therefore, to flexibly deploy adversarial examples in the physical world, some researchers have explored an attack method that generates relatively small and visible local adversarial patches [28–30]. This method only changes some of the pixels in the image and enables the placement of an adversarial patch anywhere in the image.

Compared to classifiers, object detectors are more difficult to attack. The recognition process of object detectors is more complex, and attacks on object detectors require not only misleading label predictions but also misleading object presence predictions [31]. The current adversarial patch attack methods on object detection tasks are divided into two main categories according to the shape of the generated adversarial patches: (1) rectangular adversarial patches and (2) grid-shaped (star-shaped) adversarial patches.

Rectangular Adversarial Patches. This kind of attack method focuses on generating a fixed size rectangle adversarial patch through an optimization strategy. Liu et al. [12] specifically designed DPatch for object detectors. This method adds a rectangular adversarial patch to an image so that an object detector is only able to detect the adversarial patch, thus achieving the goal of preventing the object detector from detecting the real object. The adversarial patch generated by DPatch can effectively attack YOLO [32] and Faster R-CNN detectors. However, DPatch does not impose restrictions on the pixel values of the generated adversarial patches, and there is a possibility that the pixel values exceed the range of valid pixel values of the images. Lee and Kolter [33] improved DPatch and proposed a new attack method; they force the pixel values to stay within the valid range and change the update strategy of the adversarial patch. Moreover, they added some rotation, brightness, and position changes to the patches during their placement. Thys et al. [13] applied adversarial patch attacks to person detection tasks; they use the maximum value of the confidence in the bounding box of the object detector as the loss function to optimize the adversarial patch. In this method, the adversarial patch is not placed in the upper left corner but is placed directly on the object, which is different from DPatch.

Grid-Shaped (Star-Shaped) Adversarial Patches. Rectangular-shaped adversarial patches already have good attack performance; however, to improve this attack performance, we often need to increase the size of the adversarial patch and number of changed pixels. Moreover, the rectangular-shaped adversarial patch has a small receptive field and interferes with a few feature regions [34]. Therefore, to interfere with more features while reducing the number of changed pixels, some researchers have explored attack methods to generate grid-shaped (star-shaped) adversarial patches. Shudeng Wu et al. proposed an attack method named DPAttack [34] that can generate grid-shaped and star-shaped adversarial patches to effectively attack YOLOv4 [35] and Faster R-CNN detectors [18]. Yusheng Zhao et al. [36] proposed a consensus-based attack method that integrates multiple detectors, uses voting to select the locations where pixels need to be changed, and generates a grid-shaped adversarial patch.

Compared with grid-shaped and star-shaped adversarial patches, although rectangular-shaped adversarial patches change a larger number of pixels and interfere with fewer feature regions, they can train a universal adversarial patch for a dataset. The attack methods that generate grid-shaped or star-shaped adversarial patches need to generate a specific adversarial patch for each image, which greatly affects the utilization efficiency. Therefore,

in this paper, we choose to design an attack method that generates rectangular-shaped adversarial patches to attack object detectors on RSIs.

2.2. Method

2.2.1. Patch-Noobj Framework

To adapt the adversarial patch to the scale change of an aircraft and make the aircraft vanish from the view of an object detector, we propose an attack method named Patch-Noobj. The framework structure of Patch-Noobj is shown in Figure 1. Patch-Noobj consists of two parts: a patch applier and a detector. The patch applier is responsible for attaching the adversarial patches to aircraft of different sizes, while the detector utilizes a complete object detection process and is responsible for iterative updates of the adversarial patches via the loss function.

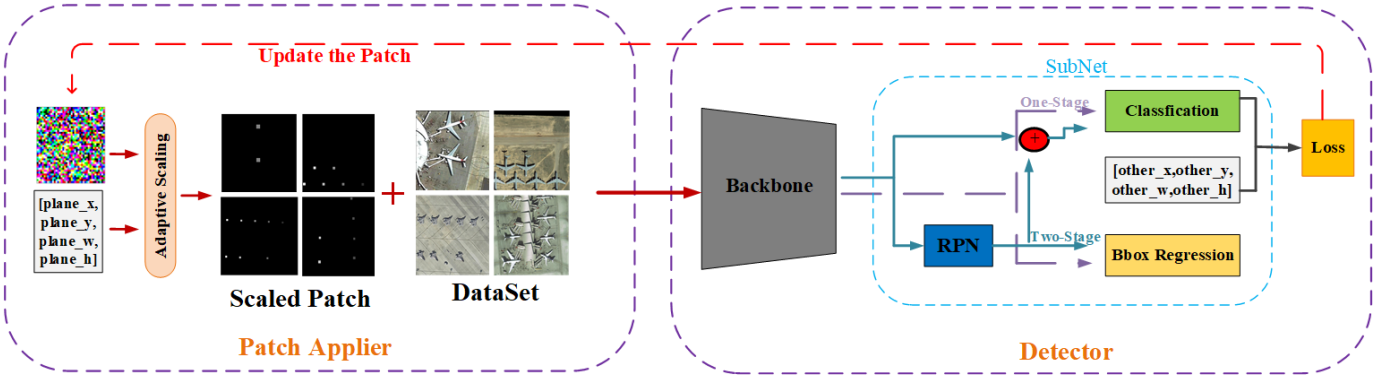


Figure 1. Overview of the framework of Patch-Noobj.

First, before an image is input to the object detector, we define the target-ground truth of the aircraft that needs to attach the adversarial patch and the untarget-ground truth of the object that does not need to attach the adversarial patch. The target-ground truth is used to calculate the scaling of the adversarial patch and construct the mask to determine where to attach the adversarial patch. The untarget-ground truth is used to calculate the loss for optimization of the adversarial patch. Both of these ground truths are similar to the ground truth of the bounding box in object detection, and they all assume the form $[x, y, w, h]$. Second, we input the image into the patch applier and randomly initialize a fixed-size adversarial patch. We calculate the scaling of the adversarial patch, construct a mask according to the target ground truth and attach the scaled adversarial patch to the aircraft in the image according to the mask. Last, we input adversarial examples with the adversarial patches into the detector, calculate the loss between the detector's output and the untarget-ground-truth based on the loss function, and iteratively update the adversarial patch by optimizing the loss.

2.2.2. Patch Applier

Patch Applier is the first component in Patch-Noobj; its task is to attach an adversarial patch on the objects that need to be attacked. In principle, the attack methods for generating a locally visible adversarial patch and a globally invisible adversarial perturbation add a perturbation to a clean image, but they differ in the way that they add the perturbation. The method of generating an adversarial patch replaces pixels in a local region of the clean image with the adversarial patch to achieve placement of the adversarial patch, while the method of generating a global perturbation directly adds pixels of the adversarial perturbation to the clean image.

In addition, in Patch-Noobj, the patch applier excepts attachment of the adversarial patch, the most important function of which is to realize the adaptive scaling of the adversarial patch so that the adversarial patch can adapt to aircraft of different sizes.

In this section, we focus on how to implement the adversarial patch adaptive scaling strategy in the patch applier. The placement of the adversarial patch will be described in Section 2.2.4.

Adaptive Scale. Compared with objects in natural images, the scale of aircraft in RSIs varies greatly. To adapt to the scale variation of aircraft in RSIs so that aircraft of different sizes have adversarial patches of different sizes, we adaptively scale the width and height of the initial adversarial patch according to the size of the attacked aircraft. We ensure that the scaled adversarial patch does not cover the entire aircraft when scaling the adversarial patch.

To scale the adversarial patch, first, we define a fixed size adversarial patch, e.g., 30×30 and 40×40. Second, we calculate the scaling ratio according to the size of the aircraft and size of the adversarial patch and scale the initial adversarial patch according to this ratio. The scaling ratio of the width and height of the adversarial patch is calculated as shown in Equations (1) and (2), where α is a scaling factor; w and h are the width and height, respectively, of the object; and $patch_w$ and $patch_h$ are the width and height, respectively, of the adversarial patch.

$$scale_w = \frac{\sqrt{(w * \frac{1}{4})^{\alpha}}}{patch_w}, \tag{1}$$

$$scale_h = \frac{\sqrt{(h * \frac{1}{4})^{\alpha}}}{patch_h}. \tag{2}$$

2.2.3. Detector

The detector is the second component in Patch-Noobj; its task is to perform the complete object detection process. It calculates the loss based on its own output and ground truth to update the pixel values of the adversarial patch by backpropagation.

In this section, we discuss the detection process of the currently popular Faster R-CNN and YOLOv3 detectors, and discuss how to set an optimization goal to iteratively update the pixel values of the adversarial patch according to the detection process of YOLOv3 so that the aircraft can evade detection by the object detector.

Faster R-CNN. Faster R-CNN is a two-stage detection algorithm. The first stage is to propose regions (rectangular regions) by deep fully convolutional network. The second stage is a Fast R-CNN detector that uses the proposed regions. In the first stage, the Faster R-CNN uses the deep fully convolutional network for feature extraction, and then the region proposal network (RPN) obtains a series of rectangular object proposals based on the feature map of the last convolutional layer. In the second stage, the rectangular object proposals generated by RPN are input to Fast R-CNN detector for classification and bounding box regression [18].

YOLOv3. YOLOv3 is a one-stage detection algorithm that reconstructs object detection as a single regression problem and obtains the object's bounding box coordinates and class probabilities in one step. YOLOv3 mainly divides the input image into $S * S$ grids, and object detection is performed inside these grids [37].

Each grid is responsible for predicting B bounding boxes and the object confidence of these B bounding boxes. The bounding box (bbox) is used to locate the detected object; it contains four values: x, y, w, and h. (x, y) represents the coordinates of the center point of the bbox relative to the boundary of the grid cell. w and h represent the relative width and height, respectively, of the bbox relative to the whole image. The object confidence indicates whether the bbox contains the object. If no object exists in the bbox, the object confidence should be zero; otherwise, the object confidence should be equal to the intersection over union (IOU) between the predicted bounding box and the ground truth. Each grid cell is also responsible for predicting the class probability of the category to which the object in the grid belongs. The class probability indicates the probability that the object belongs to each category given the presence of the object in the grid cell. When inferencing, YOLOv3

multiplies the object confidence and class probability as the class confidence of each object in the bounding box [37].

In summary, we discover that object confidence is in a more important position among the bounding box, object confidence, and class probability. If the object confidence is low, even if the bbox correctly locates the object and the class probability correctly classifies the object, the detected object will still be filtered.

Optimization Goal. The detection process of YOLOv3 reflects whether a bounding box contains a real target is determined by the object confidence. Therefore, to attack an object detector and disguise an aircraft, we only need to filter the bounding boxes containing the aircraft by reducing the object confidence to zero as much as possible.

In the training process of YOLOv3, whether a detector can accurately predict the object confidence of the bounding box is determined by the optimization of the confidence loss (L_{conf}). If the L_{conf} is optimized by decreasing the object confidence, then the real existing object in the bounding box will eventually disappear. L_{conf} consists of two parts: the first part is the loss of the bounding box that contains the object, and the second part is the loss of the bounding box that does not contain the object. L_{conf} is calculated as shown in Equation (3), where I_{ij}^{*} and$\hat{C}_i$ are constructed based on the ground truth; I_{ij}^{*} indicates whether the jth bounding box predictor of the ith grid is responsible for predicting the object; and $\hat{C}_i$ indicates whether the i-th grid contains the object.

In summary, to make L_{conf} optimize the adversarial patch to reduce the object confidence of the bounding box captaining the aircraft, we can intuitively consider converting I_{ij}^{obj} to I_{ij}^{noobj} so that the bounding box predictor that is originally responsible for predicting the aircraft changes its function to being not responsible for prediction. All the $\hat{C}_i$ that corresponds to the bounding box predictor that is originally responsible for predicting the aircraft are changed to zero so that the object confidence approaches zero in the optimization process. Both I_{ij}^{*} and $\hat{C}_i$ are constructed based on the ground truth. Thus, to implement this above mentioned idea, we need to process the input ground truth, filter the ground truth of the aircraft (target-ground-truth) and input only the ground truth of the nonaircraft (untarget-ground-truth) for loss calculation. To ensure that the optimizer prefers to generate adversarial patches with smooth color transitions in the optimization process, we calculate the total variation L_{tv} of the generated adversarial patches, as shown in Equation (4), where P denotes an adversarial patch. Therefore, in our attack method, the optimization goal consists of two parts: L_{conf} and L_{tv}, which are combined to form the total loss function. The total loss function is as shown in Equation (5), where β is a hyperparameter.

$$\begin{array}{c} L_{conf} = -\sum_{i=0}^{S*S}\sum_{j=0}^{B} I_{ij}^{obj}\left[\hat{C}_i \log(C_i) + \left(1-\hat{C}_i\right)\log(1-C_i)\right] - \\ \lambda_{noobj}\sum_{i=0}^{S*S}\sum_{j=0}^{B} I_{ij}^{noobj}\left[\hat{C}_i \log(C_i) + \left(1-\hat{C}_i\right)\log(1-C_i)\right] \end{array}, \tag{3}$$

$$L_{tv} = \sum_{i,j}\sqrt{\left(p_{i,j}-p_{i+1,j}\right)^2 + \left(p_{i,j}-p_{i,j+1}\right)^2}, \tag{4}$$

$$Loss = \beta\, \mathrm{L}_{tv} + \mathrm{L}_{conf}\,. \tag{5}$$

2.2.4. Attach Patch and Optimize Patch

To achieve the camouflage of the aircraft (let the adversarial patch replace the camouflage net), the two most important steps are the placement of the adversarial patch and the optimization of the adversarial patch. The placement of the adversarial patch involves the construction of a mask according to the location of the object to locate, and its optimization involves the use of a gradient descent algorithm to achieve iteratively according to the loss function.

Assume that x denotes the original image, $f(.)$ denotes the object detector, that m denotes a constructed binary mask that is 1 at the placement position of the adversarial patch

and 0 at the remaining positions, and that p denotes the adversarial patch. The placement and optimization of the adversarial patch can be represented by Equation (6), where $\odot$ denotes the Hadamard product (element product), t denotes the target class to be attacked, and $\mathcal{L}$ denotes the loss function, which is shown in Equation (5).

$$p^* = \arg\min \mathcal{L}(f((1 - \mathrm{m}) \odot x + m \odot p; t)). \tag{6}$$

3. Results

3.1. Databases and Evaluation Metrics

In this paper, our experiments are conducted on the DOTA [19], NWPU VHR-10 [20], and RSOD datasets [21]. These three datasets have multi-resolution images, in which the variance of aircraft sizes is prominent. The DOTA dataset contains 2806 aerial images from Google Earth and some specific satellites. The size of each image is approximately 4000×4000, and each image contains objects of various scales, orientations, and shapes. The DOTA dataset contains 15 categories of common objects, such as aircraft, ships, seaports, and bridges. The NWPU VHR-10 dataset contains 800 high-resolution satellite images cropped from the Google Earth and Vaihingen datasets, which contain 10 categories of common objects, such as aircraft, ships, harbors, and bridges. The RSOD dataset is a dataset for object detection in RSIs, which has a total of 976 images containing four categories of objects: aircraft, playground, overpass and oil drum. In our experiments, because the original images in the DOTA dataset vary in size and are large, which is not suitable for training the object detector, we cut each original image into multiple images with a size of 1024×1024. For the NWPU VHR-10 and RSOD datasets, we directly apply the original size images.

To evaluate the effectiveness of our attack method, we use two evaluation methods: Average Precision (AP) and Recall. Precision and recall are calculated as shown in Equations (7) and (8). In these two equations, TP denotes the bounding box whose IOU with ground truth is greater than the threshold. FP denotes two types of bounding boxes, one is the bounding box whose IOU with ground truth is less than the threshold, and the other is the redundant bounding box whose IOU with ground truth is greater than the threshold, but the confidence is not the highest. FN denotes the objects that is not detected, and it plus TP equals to the number of ground truth. Additionally, to better illustrate the impact of the attack method on the object detector in terms of Precision and Recall, we use the PR curve for our analysis. The PR curve can better reflect the relationship between precision and recall. When the PR curve of an object detector is more convex to the upper right, the object detector is more effective. Conversely, for the attack method of the object detector, the greater the PR-curve of the object detector can be made to move to the left after being attacked, the more effective the attack method. In addition, to analyze why the attack method is effective, we also perform a visualization analysis with the interpretable method Grad-CAM [22]. Grad-CAM calculates the importance weights of each channel feature on the recognition object in the last convolutional layer and then weights and sums the feature maps of the last convolutional layer according to the calculated importance weights to obtain a heat map. This heat map is restored to the original image size by upsampling and fused with the original image. The formula for calculating the importance weight of Grad-CAM is shown in Equation (9), where Z denotes the number of pixels in the feature map, y^c denotes the score of the c-th category, and A_{ij}^k denotes the activation value at the (i, j) position in the c-th feature map.

$$Precision = \frac{TP}{TP + FP}, \tag{7}$$

$$Recall = \frac{TP}{TP + FN}, \tag{8}$$

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}. \tag{9}$$

3.2. Experiments Details

First, we trained the YOLOv3, YOLOv5, and Faster R-CNN detectors on the DOTA, NWPU VHR-10, and RSOD datasets. Second, we used Patch-Noobj, OBJ [13], and DPatch [12] to attack the YOLOv3 detector trained on the three datasets and conducted a comparative analysis of the attack effects of these three attack methods (refer to Section 3.3). To show the attack transferability of Patch-Noobj, we constructed dataset-to-dataset, model-to-model, joint dataset-to-dataset, and model-to-model attack transfer experiments (refer to Section 3.4). Third, we explored the attack performance of adversarial patches of different sizes (refer to Section 3.5). Last, we explored why the adversarial patch is able to attack the object detector with interpretable methods (refer to Section 3.6).

3.3. Patch-Noobj Attack

In this part of the experiment, we use the Patch-Noobj, OBJ [13], and DPatch [12] attack methods to attack the YOLOv3 object detectors trained on the three datasets and then compare and evaluate the attack performance of these three attack methods on the three datasets. In generating the adversarial patches, we separately select the images containing the aircraft from the training sets of the three datasets, use 1/3 of the data for training a universal adversarial patch with an initial size of 30 × 30, and compute the evaluation metrics on the remaining 2/3 of the data to evaluate the attack effectiveness. In addition, because the OBJ also places an adversarial patch on the object, we also incorporate the scaling strategy of the adversarial patch to prevent the adversarial patch from covering the object, while the DPatch places the adversarial patch on the upper left corner of the image, so no scaling of the adversarial patch is performed. The experimental results are shown in Table 1, which shows that our method has the best attack effect compared to the other two methods on the three datasets. In terms of AP, our method reduces the best AP of the YOLOv3 object detector for aircraft detection on the DOTA (93.1%), NWPU (88.1%), and RSOD (92.0%) datasets to 44.9%, 64.2%, and 71.8%, which is a reduction of 48.2%, 23.9%, and 20.2%, respectively. Compared with the other two methods, the AP reduced by our method is 20.5%, 8.5% and 6.5% more than OBJ's and is 46.5%, 20.9% and 12.1% more than DPatch's on three datasets.A similar effect was observed in terms of Recall.

Table 1. Comparison of the attack effect of our method with that of other methods. The evaluation metrics are AP and Recall.

Datasets	Method	AP		Recall	
		Clean	Patch	Clean	Patch
DOTA	OBJ	0.931	0.654	0.978	0.728
	DPatch		0.914		0.970
	Ours		**0.449**		**0.516**
NWPU	OBJ	0.881	0.727	0.882	0.745
	DPatch		0.851		0.860
	Ours		**0.642**		**0.667**
RSOD	OBJ	0.920	0.783	0.949	0.819
	DPatch		0.839		0.881
	Ours		**0.718**		**0.745**

The experimental results indicate that, for the detector with the same network structure, the attack effect is different with different datasets and the same attack method, and the attack effect is different with the same dataset and a different attack method. We show the visual and attack effects of the adversarial patches generated with different datasets and different attack methods in Figure 2. Each row in the figure represents one attack method; from Figure 2a–c, the results of the DOTA, NWPU, and RSOD datasets are displayed. As shown in Figure 2, our adversarial patches have the best visual and attack effects for all three datasets. In terms of visual effect, the adversarial patch generated by

Patch-Noobj is better, and the adversarial patches generated by the other two attack methods more resemble noise, especially the adversarial patch generated by DPatch. In terms of the attack effect, all three methods enable the aircraft in the image to evade detection by the object detector, but Patch-Noobj is more effective and makes more of the aircraft vanish from the image.

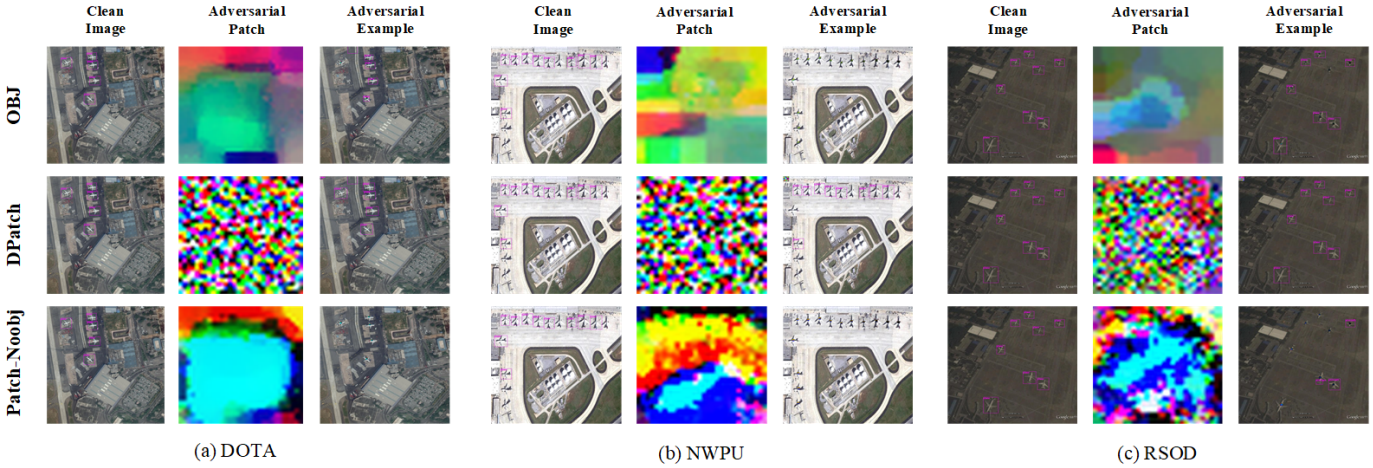


Figure 2. Visual and attack effects of adversarial patches on different datasets. In each subfigure, each row represents the visual and attack effect of the adversarial patch generated by one attack method.

Moreover, adding an adversarial patch to each image changes some of the pixels in the image; these altered pixels achieve an attack on the object detector, making the aircraft disappear from the view of the detector. However, the effects of these altered pixels on the detection results of other categories of objects while the attacking aircraft is unknown. Therefore, we further explored the effect of the attack method on the detection results of other categories of objects.

In the experiments, because images in the RSOD dataset that contain aircraft do not contain other categories of objects, we only use the DOTA and NWPU datasets in our analysis. For the DOTA dataset, among the selected images containing aircraft, the number of objects in some categories is very small, and the objects in these categories have low AP values in the object detector. Thus, we filtered other categories, except aircraft, again and selected only categories whose AP was at least 20% on clean images to explore the effect of the adversarial patch on them. The experimental results are shown in Table 2, which reveal that DPatch has the least impact on the detection results of other categories but has a poor attack effect on aircraft because it places the adversarial patch in the upper left corner of the image and does not cover the objects in the image. Our method places the adversarial patch on the aircraft but has little impact on the detection results of other categories. Compared to clean images, the AP of the object detector for detecting other categories of objects on the DOTA dataset and NWPU dataset was reduced by only 1.4% and 3.3%, respectively.

Table 2. Effect of the different attack methods on the detection results of other categories of objects.

Datasets	Method	Plane (AP)		Other Classes (AP)	
		Clean	Patch	Clean	Patch
DOTA	OBJ	0.931	0.654	0.570	0.562
	DPatch		0.914		0.565
	Ours		0.449		0.556
NWPU	OBJ	0.881	0.727	0.918	0.883
	DPatch		0.851		0.896
	Ours		0.642		0.885

3.4. Attack Transferability

To evaluate our attack method more comprehensively, in this experiment, we set up three scenarios to evaluate the attack transferability: dataset-to-dataset (Scenario A), model-to-model (Scenario B), and joint dataset-to-dataset and model-to-model (Scenario C). For scenario A, we choose YOLOv3 as the attacked detector and use the adversarial patch trained on one dataset to attack the object detectors trained on the other two datasets. For scenario B, we mainly train the YOLOv3, YOLOv5, and Faster R-CNN detectors on the same dataset and then use the adversarial patches trained on YOLOv3 to attack the YOLOv5 and Faster R-CNN detectors. For scenario C, we use the adversarial patch generated on YOLOv3 detector trained on one dataset to attack the YOLOv5 and Faster R-CNN detectors trained on the other two datasets.

In the experiments, we use AP as an evaluation metric. The experimental results for the three scenarios are shown in Tables 3–5. In Table 3, we divide the experimental results into three groups according to the target dataset. The cases in which the source dataset and target dataset are the same in each group indicate the attack effect of the adversarial patch obtained on the target dataset, and the remaining cases indicate that the attack effect of the adversarial patch obtained on the source dataset transfers to the target dataset. As shown in Table 3, the adversarial patches generated by our method have strong attack transferability among the three datasets. When the adversarial patches trained on the source dataset are transferred to the target dataset, all the transferable adversarial patches are able to maintain comparable attack effectiveness with the adversarial patches trained on the target dataset. When the adversarial patches obtained on the NWPU and RSOD datasets are transferred to the DOTA dataset, the transferable adversarial patches reduce the AP of the object detectors to detect the aircraft by 30.0% and 34.0%, which is only 18.2% and 14.2% lower than the native adversarial patches obtained on the DOTA dataset. When the adversarial patches on the DOTA and RSOD datasets are transferred to the NWPU dataset, the transferable adversarial patches reduce the AP of the object detectors to detect the aircraft by 14.5% and 23.9%, which is only 9.4% lower and 12.0% lower than the native adversarial patches obtained on the NWPU dataset. When the adversarial patches on the DOTA and NWPU datasets are transferred to the RSOD dataset, the transferable adversarial patches reduce the AP of the object detectors to detect the aircraft by 13.6% and 15.5%, which is only 6.6% lower and 4.7% lower than the native adversarial patches obtained on the RSOD dataset. In addition, Table 3 shows that the adversarial patches obtained on the same source dataset have different attack transferability on different target datasets. The adversarial patches obtained on the DOTA and NWPU datasets have more prominent attack transferability on the RSOD dataset, and the adversarial patch obtained on the RSOD dataset has more prominent attack transferability on the NWPU dataset. The results of dataset-to-dataset attack transfer experiment illustrate that with the same model and the same attack object, the adversarial patches trained on one dataset can have good attack capability on another dataset, and, according to the difference between the target and source datasets, the adversarial patches trained on the source dataset also have different attack capability on the target datasets, but the variability in attack capability is small.

Table 3. Scenario A: Attack transferability of dataset-to-dataset.

Source Dataset	Target Dataset	Clean	Adversarial Patch	Decrease (↓)
DOTA	DOTA	0.931	**0.449**	**0.482** ↓
NWPU			0.631	0.300 ↓
RSOD			0.591	0.340 ↓
DOTA	NWPU	0.881	0.736	0.145 ↓
NWPU			**0.642**	**0.239** ↓
RSOD			0.762	0.119 ↓
DOTA	RSOD	0.920	0.784	0.136 ↓
NWPU			0.765	0.155 ↓
RSOD			**0.718**	**0.202** ↓

As shown in Table 4, which represents the model-to-model scenario, for the same dataset, the adversarial patch trained on YOLOv3 showed that it can effectively attack the black-box models YOLOv5 and Faster R-CNN. For the DOTA dataset, the transferable adversarial patch reduces the AP of these two black-box models for detecting aircraft by 23.5% and 20.5%. For the NWPU dataset, the transferable adversarial patch reduces the AP of these two black-box models for detecting aircraft by 27.4% and 15.2%. For the RSOD dataset, the transferable adversarial patch reduces the AP of these two black-box models for detecting aircraft by 20.4% and 21.4%. In addition, combining the AP reductions for the three datasets, the adversarial patch obtained on YOLOv3 achieves the largest reduction in the AP of YOLOv5 for detecting aircraft, which reduces by an average of 23.8% for the three datasets. This approach reduces the AP of the Faster R-CNN the least, which reduces by an average of 19.0% for the three datasets. This finding suggests that the more similar the model structures of the object detectors are, the stronger the adversarial patches generated based on these models are in terms of the attack transferability between them.

Table 4. Scenario B: Attack transferability of model-to-model.

Datasets	Model	Clean	Adversarial Patch	Decrease (↓)
DOTA	YOLOv5	0.972	0.737	0.235 ↓
DOTA	Faster R-CNN	0.815	0.610	0.205 ↓
NWPU	YOLOv5	0.906	0.632	0.274 ↓
NWPU	Faster R-CNN	0.709	0.557	0.152 ↓
RSOD	YOLOv5	0.928	0.724	0.204 ↓
RSOD	Faster R-CNN	0.720	0.506	0.214 ↓

Table 5 represents the scenarios of joint dataset-to-dataset and model-to-model. The adversarial patches generated by our method have good attack transferability between different models trained on different datasets. For the situation in which DOTA is the source dataset, when the adversarial patch trained on YOLOv3 is used to attack the YOLOv5 and Faster R-CNN detectors trained on the NWPU dataset, it is able to reduce the AP of these two detectors for detecting aircraft by 22.7% and 14.3%, respectively. For the RSOD dataset, the adversarial patch trained on YOLOv3 is able to reduce the AP of YOLOv5 and Faster R-CNN for detecting aircraft by 15.5% and 21.2%. For the situation in which NWPU is the source dataset, the adversarial patch trained on YOLOV3 is able to reduce the AP of YOLOv5 and Faster R-CNN for aircraft detection with the DOTA dataset by 12.7% and 13.9%, respectively, and to reduce the AP by 21.7% and 19.9%, respectively, with the RSOD datasets. For the situation in which RSOD is the source dataset, the adversarial patch trained on YOLOV3 is able to reduce the AP of YOLOv5 and Faster R-CNN for aircraft detection on the DOTA dataset by 16.0% and 20.6%, respectively, and to reduce the AP by 24.3% and 15.4%, respectively, with the NWPU datasets. Dividing the experimental results according to the model structure, the AP reduction also indicates that the more similar the

model structures of the object detectors are, the stronger the adversarial patches generated based on these models are in terms of the attack transferability between them.

Table 5. Scenario C: Attack transferability of joint dataset-to-dataset and model-to-model.

Source Dataset	Target Dataset	Model	Clean	Adversarial Patch	Decrease (↓)
DOTA	NWPU	YOLOv5	0.906	0.679	0.227 ↓
		Faster R-CNN	0.709	0.566	0.143 ↓
	RSOD	YOLOv5	0.920	0.765	0.155 ↓
		Faster R-CNN	0.720	0.508	0.212 ↓
NWPU	DOTA	YOLOv5	0.972	0.845	0.127 ↓
		Faster R-CNN	0.815	0.676	0.139 ↓
	RSOD	YOLOv5	0.920	0.703	0.217 ↓
		Faster R-CNN	0.720	0.521	0.199 ↓
RSOD	DOTA	YOLOv5	0.972	0.812	0.160 ↓
		Faster R-CNN	0.815	0.609	0.206 ↓
	NWPU	YOLOv5	0.906	0.663	0.243 ↓
		Faster R-CNN	0.709	0.555	0.154 ↓

3.5. Attack Performance for Different Size Patches

The size of the adversarial patch affects the attack performance. For a fixed-size adversarial patch, the larger the size is, the better the attack performance is because it covers a larger area of the object and interferes more with the features of the object. However, in this paper, we add an adaptive scaling strategy for the adversarial patch to adapt to objects of different sizes and cannot determine what size of the initial adversarial patch has the best attack performance. Therefore, in this part of the experiment, we explore the impact of different sizes of adversarial patches on the attack performance of our method.

We explored the attack performance of a total of five different sizes of adversarial patches on the DOTA dataset; the experimental results are shown in Figure 3 and Table 6. The experimental results show that the overall trend of poor attack performance when the size of the adversarial patch is small and strong attach performance when the size of the adversarial patch is large. However, according to the experimental results, the adversarial patch has the best attack performance when its size reaches 30×30; the attack performance is gradually enhanced with an increase in size when the size of the adversarial patch is smaller than 30×30; and the attack performance gradually weakens with an increase in size increases when the size of the adversarial patch exceeds 30×30. These results show that the case that the larger the size of the adversarial patch is, the better the attack performance is does not hold for our method. The attack performance will start to weaken when the size limit of the adversarial patch is exceeded.

Table 6. Attack performance of our adversarial patches of different sizes. The evaluation metrics are AP and Recall.

Size	AP	Recall
10×10	0.623	0.690
20×20	0.548	0.609
30×30	**0.449**	**0.516**
40×40	0.477	0.542
50×50	0.490	0.546

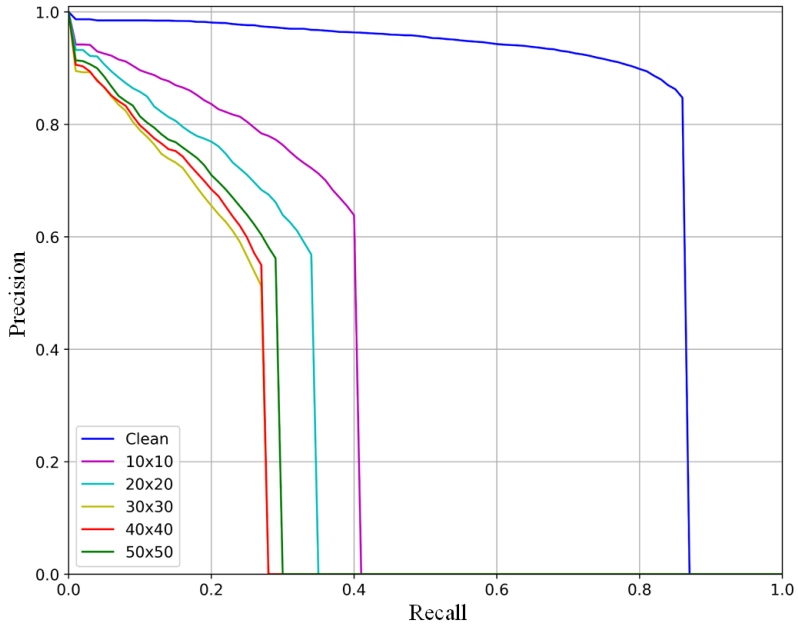
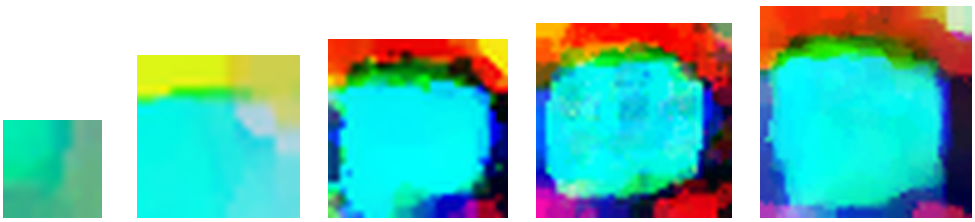


Figure 3. PR curves of our adversarial patches of different sizes compared with clean images (Clean).

In addition, we also visualize adversarial patches of different sizes, as shown in Figure 4. The visual effect of the adversarial patch gradually improves as the size of the adversarial patch increases and eventually generates a visually relatively consistent adversarial patch. The small adversarial patch is visually single in color, which may explain its poor attack performance. The large adversarial patch is rich in color and can more effectively interfere with the features of the attacked object, so the attack performance is better. Combined with this analysis, when the size of the adversarial patch reaches 30×30, it has the best attack performance, and, when the size is 40×40 and 50×50, the attack performance starts to weaken, which is not much different from the attack performance of the adversarial patch with size 30×30. These results may be attributed to the notion that the adversarial patch has converged to a fixed shape, but, as the size increases, the optimization process produces some noise-like pixels in the large adversarial patch, which causes a decrease in the attack performance. Because the shape is basically the same as the adversarial patch with size 30×30, the attack performance does not produce a large difference.



(**a**) 10×10 (**b**) 20×20 (**c**) 30×30 (**d**) 40×40 (**e**) 50×50

Figure 4. Visual effect of adversarial patches of different sizes. From left to right, the visualization results of the adversarial patches of size 10×10, $20 \times 20 \ldots 50 \times 50$.

3.6. Why Patch-Noobj Works

The experiments in this study suggest that the adversarial patch generated by our method is effective in attacking the YOLOv3 detector by reducing the object confidence of aircraft. However, since we do not know how the adversarial patch interferes with the YOLOv3 detector and reduces the object confidence of aircraft, we conduct further investigation.

We perform a visualization analysis with Grad-CAM, an interpretable algorithm for image classification tasks. The application scenario of the original Grad CAM visualization method is an image classification task, while our current task is an object detection task, so we have improved Grad-CAM to apply it to the object detection task.

In this paper, the main idea of improving Grad CAM is that, for each category of objects in the input image, we obtain a bounding box with the highest class probability for this category, backpropagate the class probability of this bounding box and calculate its gradient to the feature map of the previous convolution layer.We calculate the importance weight of each feature map and multiply and sum the importance weights with the feature map to obtain the heat map. We fuse the heat map with the input image to obtain the Grad CAM visualization image.

We used Grad-CAM to visualize and analyze the feature extraction of the YOLOv3 detector on the normal examples and the adversarial examples with the adversarial patches. The experimental results, which are shown in Figure 5, indicate that, for the clean images, the detector focuses on the features on the aircraft during detection with a high degree of attention. When the adversarial patch is attached to the aircraft, for aircraft that are successfully attacked, the features on which the detector focuses are either shifted from the aircraft to other locations or remain on the aircraft, and the degree of attention is significantly reduced. The experimental results indicate that the adversarial patch is able to attack the detector because it affects the capture of aircraft features by the detector, thus causing the detector to lose the aircraft's contextual information and eventually be deceived.

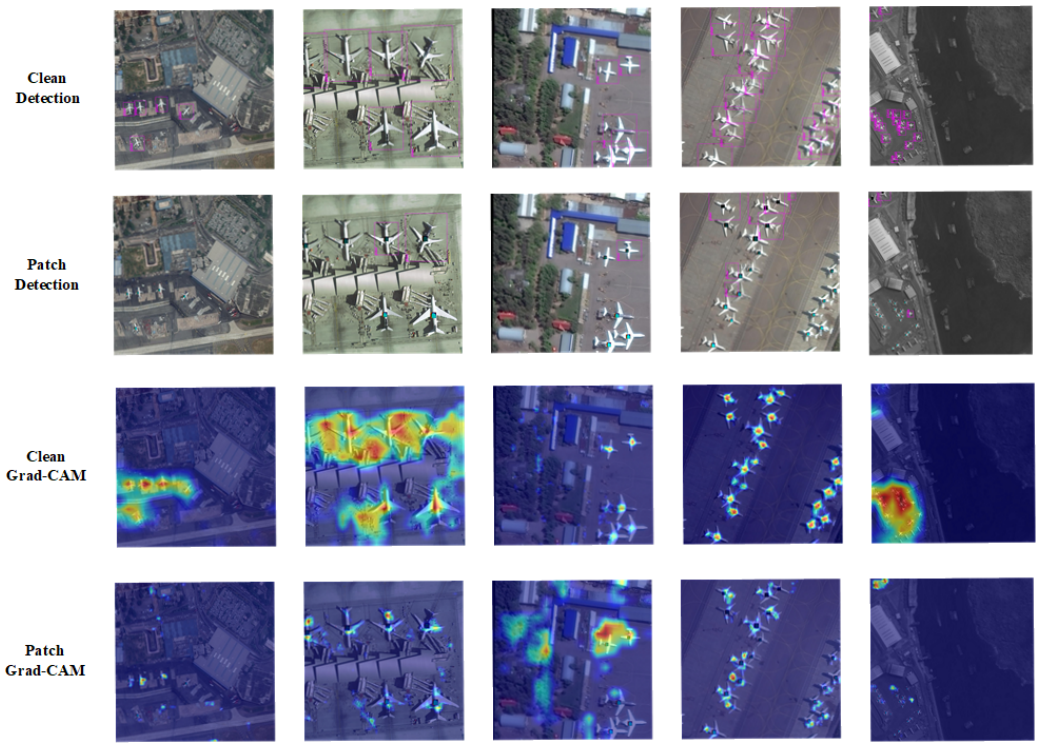


Figure 5. Grad-CAM visualization results for normal and adversarial examples. Each column in the figure represents the detection results and Grad-CAM visualization results of a normal example and its corresponding adversarial example.

4. Discussion and Conclusions

In this paper, we propose an adversarial attack method named Patch-Noobj for object detectors, which can hide the decision features of an aircraft in object detectors and effectively make the aircraft disappear from the view of the object detectors. Patch-Noobj adds an adaptive scaling strategy for the adversarial patch to adapt to different sizes of aircraft. The adversarial patches are scaled according to the size of the aircraft so that the

adversarial patches can attack the object detectors without completely covering the aircraft. In the experiments, Patch-Noobj can effectively attack object detectors and outperform other methods with multiple datasets. It reduces the AP of the detector in detecting aircraft on the DOTA, NWPU VHR-10, and RSOD datasets by 48.2%, 23.9%, and 20.2%, respectively. There was a little effect on the detection results of other categories when attacking the aircraft. Moreover, the experimental results show that the adversarial patches generated by Patch-Noobj have good attack transferability. Through visual analysis, we show why the adversarial patches are able to attack the object detectors. In addition, our method currently has some limitations. It only trains the adversarial patch based on a single model, and the adversarial patch attached by way of constructing a mask does not blend in perfectly with the object, especially some edge parts.

In future work, we will consider three directions of research based on this paper. First, to enhance the robustness of the attack method, we will incorporate the idea of ensemble learning in the generation process of adversarial patches. Second, we will explore generating adversarial patches of different shapes, such as circles and polygons, instead of just generating rectangular-shaped adversarial patches. Last, we will explore different ways of attaching the adversarial patches, such as the perspective transformation method, instead of simply attaching the adversarial patches by mask.

Author Contributions: Conceptualization, Q.L. and L.C.; methodology, Q.L.; software, Q.L.; validation, Q.L., M.L., L.C., and H.L.; formal analysis, Q.L. and L.C.; investigation, Q.L.; resources, L.C.; data curation, Q.L.; writing—original draft preparation, Q.L.; writing—review and editing, L.C., H.L., and M.L.; visualization, Q.L.; supervision, M.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (41871364, 41871276, 41871302, and 41861048).

Institutional Review Board Statement: Not applicable for studies not involving humans.

Informed Consent Statement: Not applicable for studies not involving humans or animals.

Data Availability Statement: The data that support the findings of this study are available from the author upon reasonable request. The source code can be visited at https://github.com/GeoX-Lab/XAI4RS.

Acknowledgments: This work was carried out in part using computing resources at the High Performance Computing Center of Central South University.

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

References

1. Ji, F.; Ming, D.; Zeng, B.; Yu, J.; Qing, Y.; Du, T.; Zhang, X. Aircraft Detection in High Spatial Resolution Remote Sensing Images Combining Multi-Angle Features Driven and Majority Voting CNN. *Remote Sens.* **2021**, *13*, 2207. [CrossRef]
2. Wang, J.; Xiao, H.; Chen, L.; Xing, J.; Pan, Z.; Luo, R.; Cai, X. Integrating Weighted Feature Fusion and the Spatial Attention Module with Convolutional Neural Networks for Automatic Aircraft Detection from SAR Images. *Remote Sens.* **2021**, *13*, 910. [CrossRef]
3. Han, X.; Zhong, Y.; Zhang, L. An efficient and robust integrated geospatial object detection framework for high spatial resolution remote sensing imagery. *Remote Sens.* **2017**, *9*, 666. [CrossRef]
4. Cai, B.; Jiang, Z.; Zhang, H.; Zhao, D.; Yao, Y. Airport detection using end-to-end convolutional neural network with hard example mining. *Remote Sens.* **2017**, *9*, 1198. [CrossRef]
5. Wang, Y.; Li, H.; Jia, P.; Zhang, G.; Wang, T.; Hao, X. Multi-scale densenets-based aircraft detection from remote sensing images. *Sensors* **2019**, *19*, 5270. [CrossRef] [PubMed]
6. Mohamad Nezami, O.; Chaturvedi, A.; Dras, M.; Garain, U. Pick-Object-Attack: Type-specific adversarial attack for object detection. *Comput. Vis. Image Underst.* **2021**, *211*, 103257. [CrossRef]
7. Zhang, Y.; Wang, F.; Ruan, W. Fooling Object Detectors: Adversarial Attacks by Half-Neighbor Masks. *arXiv* **2021**, arXiv:2101.00989.
8. Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; Yuille, A. Adversarial examples for semantic segmentation and object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1369–1378. [CrossRef]

9. Chen, S.T.; Cornelius, C.; Martin, J.; Chau, D.H. Robust Physical Adversarial Attack on Faster R-CNN Object Detector. *arXiv* **2018**, arXiv:1804.05810.
10. Li, Y.; Tian, D.; Chang, M.C.; Bian, X.; Lyu, S. Robust adversarial perturbation on deep proposal-based models. *arXiv* **2018**, arXiv:1809.05962.
11. Bose, A.J.; Aarabi, P. Adversarial attacks on face detectors using neural net based constrained optimization. In Proceedings of the 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP), Vancouver, BC, Canada, 29–31 August 2018; pp. 1–6. [CrossRef]
12. 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.
13. 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, Long Beach, CA, USA, 16–17 June 2019. [CrossRef]
14. Zhao, Y.; Zhu, H.; Liang, R.; Shen, Q.; Zhang, S.; Chen, K. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 1989–2004. [CrossRef]
15. Yang, X.; Wei, F.; Zhang, H.; Zhu, J. Design and interpretation of universal adversarial patches in face detection. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28. 2020, Proceedings, Part XVII 16*; Springer: Cham, Switzerland, 2020; pp. 174–191. [CrossRef]
16. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
17. Jocher, G.; Stoken, A.; Borovec, J.; NanoCode012; Christopher, S.; Liu, C.; Laughing; Hogan, A.; Lorenzo, M.; Tkianai; et al. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 11 August 2021).
18. 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]
19. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983. [CrossRef]
20. Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [CrossRef]
21. Xiao, Z.; Liu, Q.; Tang, G.; Zhai, X. Elliptic Fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images. *Int. J. Remote Sens.* **2015**, *36*, 618–644. [CrossRef]
22. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626. [CrossRef]
23. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
24. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
25. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2016**, arXiv:1607.02533.
26. 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. [CrossRef]
27. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582. [CrossRef]
28. Liu, A.; Liu, X.; Fan, J.; Ma, Y.; Zhang, A.; Xie, H.; Tao, D. Perceptual-sensitive gan for generating adversarial patches. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 1028–1035. [CrossRef]
29. Karmon, D.; Zoran, D.; Goldberg, Y. Lavan: Localized and visible adversarial noise. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 2507–2515.
30. Brown, T.B.; Mané, D.; Roy, A.; Abadi, M.; Gilmer, J. Adversarial patch. *arXiv* **2017**, arXiv:1712.09665.
31. 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 18), Baltimore, MD, USA, 13–14 August 2018.
32. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271. [CrossRef]
33. Lee, M.; Kolter, Z. On physical adversarial patches for object detection. *arXiv* **2019**, arXiv:1906.11897.
34. Wu, S.; Dai, T.; Xia, S.T. Dpattack: Diffused patch attacks against universal object detection. *arXiv* **2020**, arXiv:2010.11679.
35. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
36. Zhao, Y.; Yan, H.; Wei, X. Object hider: Adversarial patch attack against object detectors. *arXiv* **2020**, arXiv:2010.14974.
37. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]