remote sensing

MDPI

Technical Note

CamoNet: A Target Camouflage Network for Remote Sensing Images Based on Adversarial Attack

Yue Zhou, Wanghan Jiang, Xue Jiang *, Lin Chen and Xingzhao Liu

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 201100, China; sjtu_zy@sjtu.edu.cn (Y.Z.); wanghan@sjtu.edu.cn (W.J.); linchen@sjtu.edu.cn (L.C.); xzhliu@sjtu.edu.cn (X.L.)

* Correspondence: xuejiang@sjtu.edu.cn

Abstract: Object detection algorithms based on convolutional neural networks (CNNs) have achieved remarkable success in remote sensing images (RSIs), such as aircraft and ship detection, which play a vital role in military and civilian fields. However, CNNs are fragile and can be easily fooled. There have been a series of studies on adversarial attacks for image classification in RSIs. However, the existing gradient attack algorithms designed for classification cannot achieve excellent performance when directly applied to object detection, which is an essential task in RSI understanding. Although we can find some works on adversarial attacks for object detection, they are weak in concealment and easily detected by the naked eye. To handle these problems, we propose a target camouflage network for object detection in RSIs, called CamoNet, to deceive CNN-based detectors by adding imperceptible perturbation to the image. In addition, we propose a detection space initialization strategy to maximize the diversity in the detector's outputs among the generated samples. It can enhance the performance of the gradient attack algorithms in the object detection task. Moreover, a key pixel distillation module is employed, which can further reduce the modified pixels without weakening the concealment effect. Compared with several of the most advanced adversarial attacks, the proposed attack has advantages in terms of both peak signal-to-noise ratio (PSNR) and attack success rate. The transferability of the proposed target camouflage network is evaluated on three dominant detection algorithms (RetinaNet, Faster R-CNN, and RTMDet) with two commonly used remote sensing datasets (i.e., DOTA and DIOR).

Keywords: remote sensing; deep learning; object detection; adversarial attacks; CNN; target camouflage

Citation: Zhou, Y.; Jiang, W.; Jiang, X.; Chen, L.; Liu, X. CamoNet: A Target Camouflage Network for Remote Sensing Images Based on Adversarial Attack. *Remote Sens.* **2023**, *15*, 5131. https://doi.org/10.3390/rs15215131

Academic Editor: Pedro Melo-Pinto

Received: 14 August 2023
Revised: 11 October 2023
Accepted: 22 October 2023
Published: 27 October 2023

1. Introduction

With the substantial progress of remote sensing technology, the automatic interpretation of remote sensing images (RSIs) has significantly improved [1,2]. Higher resolution and large-scale data give impetus to the development of RSI interpretation systems in scene classification [3,4], object detection [5,6], and semantic segmentation [7–11]. Automated RSI interpretation with robustness and high accuracy can lead to significant economic benefits. Object detection performance, which is the cornerstone of the RSI interpretation system, is especially crucial. With the development of deep learning theory and computer hardware technology, especially the success of convolutional neural networks (CNNs), significant breakthroughs have been made in object detection [12,13]. Intuitively, many excellent detection algorithms [14,15] have been introduced to the RSI interpretation system. These RSI detection models outperform traditional detectors in terms of accuracy and efficiency, which is inseparable from CNN's excellent feature extraction capabilities.

However, many CNN-based models exhibit a lack of stability with regard to specially designed "small" perturbations to the signal input space [16]. These "small" disturbances keep the signal content perceived by humans unchanged while causing fundamental changes in the output of the CNN-based model. The CNN-based model has poor resistance

to these specially designed disturbances and is fragile. Images containing such disturbances are usually referred to as adversarial examples. The concept of adversarial examples was introduced in the field of image classification by Szegedy et al. [17]. In recent years, adversarial samples for classification tasks have been widely used in RSIs [18,19].

The object detection task is another critical problem in RSI understanding. It combines multitarget localization and multitarget classification problems simultaneously. The candidate regions generated in the object detection task are more challenging than image classification alone. Most existing research on adversarial attacks focuses on image classification problems, while little attention is given to object detectors. The first adversarial attack model for object detection was proposed by Lu et al. [20], who suggested adding perturbations to traffic signs and human faces in images to mislead detectors. Then, a series of studies were conducted on adversarial attacks for object detection.

Nevertheless, most of these adversarial attacks for object detection directly adopt gradient attacks designed for classification tasks and use random adoption for gradient initialization [21–23]. This unreasonable approach leads to compromised attack performance. In fact, the diversity of the input space cannot be directly converted into the diversity of the output space. If only the input is initialized randomly, the initialization of the output space cannot be guaranteed to be completely random. Although this has little impact on classification tasks, it is crucial for object detection tasks. For object detection, the complexity of the output space is far greater than image classification. Specifically, the classification output is a vector representing the probability of each category. In contrast, the detection output is a high-dimensional tensor containing the target's location, category, and confidence score. If the diversity of the detector's outputs is not maximized, the gradient attack may fall into the suboptimal solution. This limits the performance of adversarial attack methods in object detection tasks.

According to the range of modification to the detected target pixels, adversarial attacks for object detection can generally be classified into global attacks [20,24–27] and patch-based attacks [28–30]. A global attack modifies the entire image's pixels when adversarial examples are generated. Unlike the global attack, the patch-based attack only adds a perturbation in a specific area of the original image so that the perturbation in this area can affect the whole image and deceive the detector. Note that the targets in RSIs usually have different sizes and uneven distributions. Hence, it is difficult for a local perturbation attack to exert the same effect on all targets. However, the gradient attack algorithms adopted by these models are all designed for classified tasks and are unsuitable for object detection tasks.

However, both global attacks and patch-based attacks are easily detected by the naked eye [31,32]. Global attacks are unsuitable for the RSI's object detection because the target only occupies a small area in the RSI [33]. This means the perturbation generated in the background area usually does not improve the attack effect but may add the perturbed pixels and weaken the invisibility. For patch-based attacks, humans can easily perceive attacks due to the large contrast between patch pixels and image pixels, which is not conducive to the camouflage of targets [34]. These patches usually have a large average gradient, so some defense measures can even detect the location of the target by detecting patch-based attacks [35].

To overcome these problems, we propose a target camouflage network based on the adversarial attack for RSIs, named CamoNet. In the natural world, camouflage is one of the essential anti-predator defenses that prevent prey from being recognized by predators [36]. Inspired by this, we limit the perturbation to the target itself with a mask and propose a detection space initialization (DSI) strategy. Moreover, we integrate a key pixel distillation (KPD) module into CamoNet. It can further reduce the pixels of disturbance while ensuring the attack effect, making the attack more concealed and imperceptible to the naked eye.

Next we summarize the main contributions of this work as follows:

(i) This paper proposes a target camouflage network based on the adversarial attack for RSIs, named CamoNet, which provides a new view instead of traditional target cam-

ouflage. We propose a new initialization strategy, to maximize diversity in the detector's outputs among the generated samples. It can boost the performance of the gradient attack algorithms designed for classification in the object detection task.

(ii) Inspired by the anti-predator defense mechanism in the natural world, the KPD module is employed for the target camouflage network. It can make the attack more concealed and imperceptible to the naked eye without weakening the attack effect.

(iii) The experimental results of the DOTA and DIOR datasets validate the effectiveness of CamoNet. The ablation experiment demonstrates each component's effectiveness. At the same time, we further explore the network's generality and transferability.

The rest of the paper is organized as follows. Related work is introduced in Section 2. In Section 3, CamoNet is proposed, the validation experiment results of which are shown in Section 4. Finally, the conclusions are presented in Section 5.

Throughout the paper, matrices, vectors, and scalars are represented by bold uppercase letters $\boldsymbol{X}$, bold lowercase letters $\boldsymbol{x}$, and regular letters x, respectively. Superscripts $(\cdot)^T$ and $(\cdot)^{-1}$ represent the transpose and inverse, respectively. We use $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_p$ to denote the ℓ_1, ℓ_2, and ℓ_p norms, respectively.

2. Related Work

In recent years, a batch of excellent work has emerged in the field of remote sensing object detection. Lu et al. [37] proposed adaptively placing counter patches of different sizes according to the size of the aircraft, which can effectively improve the success rate of patch-based attacks on RSIs. Zhang et al. [38] raised a scale factor to generate adversarial patches that adapt to multiscale objects. Zhang et al. [32] designed an IoU-based objective function specific for patch attacks, which can push the detected boxes far from the initial ones. Tang et al. [39] proposed a novel natural weather-style patch attack, which is more natural and stealthy. Due to the threat posed by adversarial attacks to remote sensing detectors, Li et al. [40] began researching remote sensing target detectors that are robust to adversarial attacks.

In summary, their modifications to the algorithm mainly focus on the design of gradient attack vectors and objective functions. In the following parts of this section, we first introduce several classical gradient attack methods in the field of adversarial attacks. Then, we introduce two adversarial attack objective functions designed for object detection tasks. Finally, we introduce two remote sensing image object detection datasets used in the experimental part of this article.

2.1. Gradient Attacks

Gradient attacks are input samples formed by intentionally adding subtle pixels to the data. The disturbed input makes the model give an incorrect output with high confidence. In this subsection, several representative gradient-based attacks used in the experiments of this article are briefly shown.

The fast Gradient Notation Method (FGSM) [21] is a gradient-based attack. At the same time, it is also a non-iterative attack because it computes the gradient only once for each image. However, FGSM does not match the reality by assuming that DCNN is linear. The activation functions in DCNN make them not strictly linear.

Given an image $\boldsymbol{I}$ and its true label $\boldsymbol{y}$, the adversarial example $\boldsymbol{I}_{\text{adv}}$ can be calculated as

$$\boldsymbol{I}_{\text{adv}} = \text{clip}(\boldsymbol{I} + \epsilon \, \text{sign}(\nabla_{\boldsymbol{I}} J(\theta, \boldsymbol{I}, \boldsymbol{y}))) \tag{1}$$

where $\nabla_{\boldsymbol{I}} J(\theta, \boldsymbol{I}, \boldsymbol{y})$ calculates the gradients of the loss function $J(\cdot)$ with respect to the input sample $\boldsymbol{I}$, $\text{sign}(\cdot)$ denotes the sign function, and $\text{clip}(\cdot)$ clips the pixel values in the image. In addition, with different values of ϵ, we can obtain adversarial examples with different attack strengths.

For the gradient-based attack, project gradient descent (PGD) [22] can be considered an iterative version of FGSM. At each iteration, the adversarial example can be updated as follows:

$$\boldsymbol{I}_{\text{adv}}{}^{t+1} = \text{clip}\left(\boldsymbol{I}_{\text{adv}}{}^{t} + \alpha \ \text{sign}\left(\nabla_{\boldsymbol{I}_{\text{adv}}} J\left(\theta, \boldsymbol{I}_{\text{adv}}{}^{t}, \boldsymbol{y}\right)\right)\right) \tag{2}$$

where α is the step size. We can obtain adversarial examples with different attack strengths by setting the parameters α and the iterations of PGD.

MI-FGSM [23] is also an iterative version of FGSM, which applies the idea of momentum to generate adversarial examples. At each iteration, the adversarial example can be updated as follows:

$$\boldsymbol{G_{t+1}} = \mu \cdot \boldsymbol{G_t} + \frac{\nabla_{\boldsymbol{I}_{\text{adv}}{}^{t}} J\left(\theta, \boldsymbol{I}_{\text{adv}}{}^{t}, \boldsymbol{y}\right)}{\|\nabla_{\boldsymbol{I}_{\text{adv}}{}^{t}} J\left(\theta, \boldsymbol{I}_{\text{adv}}{}^{t}, \boldsymbol{y}\right)\|_1} \tag{3}$$

$$\boldsymbol{I}_{\text{adv}}{}^{t+1} = \text{clip}\left(\boldsymbol{I}_{\text{adv}}{}^{t} + \alpha \ \text{sign}(\boldsymbol{G_{t+1}})\right) \tag{4}$$

where $\boldsymbol{G}$ is the accumulated gradient, which is updated by accumulating the velocity vector in the gradient direction.

2.2. Objective Functions

Object detection's objective function is more complex than the classification task's objective function. Given an RSI $\boldsymbol{I}$, the object detector first detects a large number of S proposals $\hat{\boldsymbol{\mathcal{B}}}(\boldsymbol{I}) = \{\hat{\boldsymbol{o}}_1, \hat{\boldsymbol{o}}_2, \ldots, \hat{\boldsymbol{o}}_S\}$, where $\hat{\boldsymbol{o}}_i = (\hat{b}_i^x, \hat{b}_i^y, \hat{b}_i^W, \hat{b}_i^H, \hat{C}_i, \hat{\boldsymbol{p}}_i)$ is a proposal centered at $(\hat{b}_i^x, \hat{b}_i^y)$ having a dimension $(\hat{b}_i^W, \hat{b}_i^H)$ with a confidence score of $\hat{C}_i \in [0, 1]$, and K-class probabilities $\hat{\boldsymbol{p}}_i = (\hat{p}_i^1, \hat{p}_i^2, \ldots, \hat{p}_i^K)$. Because in actual application scenarios it is often difficult to obtain the annotations of the attacked image, this paper only studies two objective functions that do not need the true label $\boldsymbol{y}$.

DAG (dense adversary generation) [24] was the first model to propose the object-mislabeling loss in object detection. Its objective function is as follows

$$J_{mislabeling}(\theta, \boldsymbol{I}) = \sum_{i=1}^{S} (\ell_{CE}(\hat{\boldsymbol{p}}_{\boldsymbol{i}}, \hat{l}_i) - \ell_{CE}(\hat{\boldsymbol{p}}_{\boldsymbol{i}}, \hat{l}_i')) \tag{5}$$

where $\hat{l}_i = \arg\max(\hat{\boldsymbol{p}}_{\boldsymbol{i}})$, which is the predicted label of the ith bounding box. S is the total number of bounding boxes predicted by the model. $\ell_{CE}(\cdot)$ is the cross-entropy (CE), and $\hat{l}_i'$ is the wrong label randomly generated for the ith bounding box. The objective function is designed to narrow the score gap between the correct and wrong labels predicted by the model. However, this objective function does not consider the attack on regression loss, resulting in many candidate boxes with incorrect category prediction in the final result.

The object-vanishing loss without labels was first proposed in RPAttack [41]. The confidence score of each bounding box should be reduced to hide the objects from the detector. Based on this, the objective function is defined as

$$J_{vanishing}(v) = -\frac{1}{S} \sum_{i=1}^{S} \ell_{MSE}(\hat{C}_i, 0) \tag{6}$$

where $\ell_{MSE}(\cdot)$ is the mean square error (MSE) and $\hat{C}_i$ is the confidence score of the ith bounding box. Notably, this objective function can achieve the effect of target camouflage. Since object-vanishing loss is closest to our target camouflage requirements, we use this loss as the default objective function in the following.

3. Methodology

The proposed CamoNet for RSI target camouflage is introduced in this section, which mainly consists of two parts. First, we performed the gradient attack on the detector to obtain a perturbation of the same size as the input image. Then, we eliminated the meaningless perturbation in the RSI's background region through postprocessing. The pipeline of CamoNet is shown in Figure 1. We introduce these two phases in detail in the following subsections.

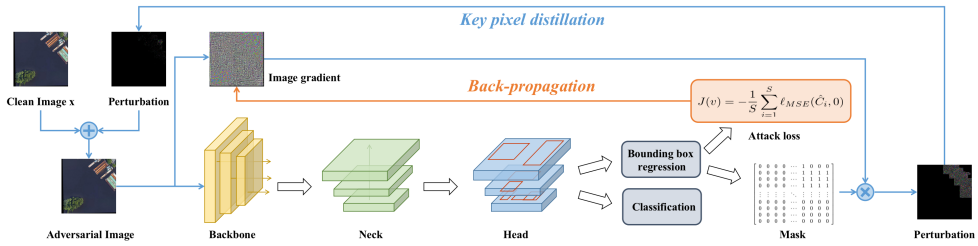

Figure 1. The pipeline of CamoNet.

3.1. Gradient Attack Phase

In the gradient attack stage, the attack, which works well in image classification, is migrated to the object detection task. The gradient attack can be simply summarized as calculating the gradient of the objective function for each pixel of the input image of the model and then adding the gradient to the input image as a disturbance to obtain an adversarial sample, which can invalidate the model. The attack goal for the object detection task is to make the detector unable to detect any targets. It should minimize the confidence of each bounding box because the confidence of a bounding box indicates whether it contains objects and the degree of accuracy. To achieve the effect of target camouflage, we choose $J_{vanishing}$ as our objective function, which is calculated according to the confidence score of the bounding boxes output by the detector. Attacks in most image classification tasks can be directly used in CamoNet. The performance of these attacks in target camouflage is quantitatively analyzed in the experiment section.

As intuitively presented in Figure 2, random initialization in the image space does not necessarily produce initialization with high diversity as measured in the detection space. To address this problem, we propose a DSI strategy to maximize diversity in the detector's outputs among the generated samples (green points in Figure 2). It can boost the performance of the gradient attack algorithms designed for classification in the object detection task. DSI can be incorporated into most adversarial attack methods. One strong and popular example is the PGD attack, called DSI-PGD.

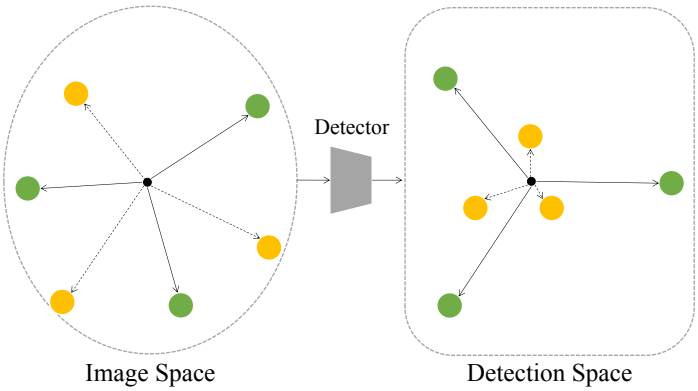

Figure 2. Illustration of the differences between random initialization (yellow points) and DSI (green points). The black point corresponds to an original image. Perturbations by DSI in the image space are crafted by maximizing the distance in the detection space of the detector.

Given the direction of diversification $\boldsymbol{w}_{\mathrm{d}} \in \mathbb{R}^{C}$, the normalized perturbation vector of DSI can be defined as follows:

$$v_{\mathrm{DSI}}(\boldsymbol{I}, J, \boldsymbol{w}_{\mathrm{d}}) = \frac{\nabla_{\boldsymbol{I}}(\boldsymbol{w}_{\mathrm{d}}{}^{\mathsf{T}}(\theta, \boldsymbol{I}, \boldsymbol{y}))}{\|\nabla_{\boldsymbol{I}}(\boldsymbol{w}_{\mathrm{d}}{}^{\mathsf{T}} J(\theta, \boldsymbol{I}, \boldsymbol{y}))\|_2}, \tag{7}$$

where $\boldsymbol{w}_{\mathrm{d}}$ is sampled from the uniform distribution over $[-1, 1]^{C}$.

In CamoNet, we utilize DSI to generate output-diversified starting points. Given an original input $\boldsymbol{I}_{\mathrm{org}}$ and the direction for DSI $\boldsymbol{w}_{\mathrm{d}}$, we try to find a restart point $\boldsymbol{I}$ that is as far

away from $\boldsymbol{I}_{\text{org}}$ as possible by maximizing $\boldsymbol{w}_{\text{d}}{}^{\intercal}(J(\theta, \boldsymbol{I}, \boldsymbol{y})) - J(\theta, \boldsymbol{I}_{\text{org}}, \boldsymbol{y}))$ via the following iterative update:

$$\boldsymbol{I}_{\boldsymbol{k+1}} = \text{Proj}_{B(\boldsymbol{I}_{\text{org}})}(\boldsymbol{I}_{\boldsymbol{k}} + \eta_{\text{DSI}} \cdot \text{sign}(v_{\text{DSI}}(\boldsymbol{I}_{\boldsymbol{k}}, J, \boldsymbol{w}_{\text{d}}))) \tag{8}$$

where $\text{Proj}_{B(\boldsymbol{I}_{\text{org}})}(\boldsymbol{I}^{\text{adv}}) \triangleq \arg\min_{\boldsymbol{I}' \in B(\boldsymbol{I}_{\text{org}})} \|\boldsymbol{I}^{\text{adv}} - \boldsymbol{I}'\|_p$, $B(\boldsymbol{I}_{\text{org}})$ is the set of allowed perturbations and η_{DSI} is a step size. After some steps of DSI, the PGD attack can restart from $\boldsymbol{I}_{\boldsymbol{k+1}}$ obtained by DSI. One step of ODI costs roughly the same time as one iteration of the PGD attack.

3.2. Postprocessing Phase

To further reduce the perturbation rate, the network performs a postprocessing operation. First, the network generates a mask according to the bounding box output by the detection model, which is a matrix with the same shape as the original image. At the position within the bounding box, the value of the mask matrix is 1, and the value at the remaining positions is 0. Then, the product of the mask and the gradient map is used as the initial perturbation applied to the original image. The mask can control the perturbation around the targets and does not involve the background area. The network repeats the above gradient attack and postprocessing until it reaches the maximum number of attack rounds.

The purpose of target camouflage is to deceive the detector and make the intelligent interpretation system fail. However, if there are many modified pixels in the image, the attack may be detected by humans. Only when the perturbation rate is as low as possible can the naked eye be deceived. To further reduce the perturbation rate, we introduced KPD to extract the most critical pixels. It adaptively filters the unimportant pixels in each patch according to their mean value and mean square error. The details of KPD are shown in Figure 3. It can further reduce the pixels of disturbance while ensuring the attack effect, making the attack more concealed and imperceptible to the naked eye.

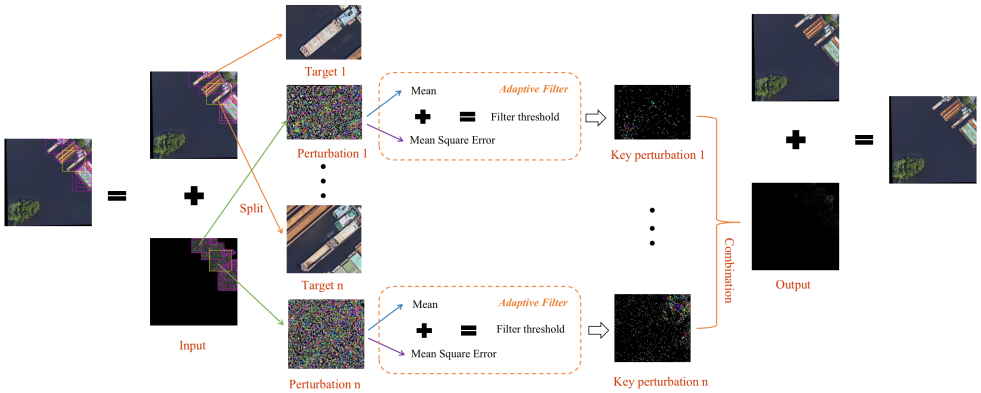

Figure 3. The detail of key pixel distillation (KPD) module.

KPD mainly screens and retains disturbances at positions with large gradients by setting a threshold. The positions that have a large impact on the gradient are the key positions for generating effective adversarial samples. Retaining the perturbation at key positions and clearing perturbation at positions that have little influence on the gradient can achieve the purpose of reducing the perturbation rate while ensuring the attack effect. Choosing the appropriate threshold is the core of module design. We tried a variety of threshold selection methods. Compared with using fixed hyperparameters to set the threshold, adaptively selecting the threshold according to the gradient obtained in each attack round can make the model more robust and not affect the attack effect due to the selection of different hyperparameters. We use the mean plus standard deviation as the threshold. The locations where the gradient is higher than the threshold are selected to retain the perturbation, and the locations where it is below the threshold are not added

to the perturbation. The mean value indicates the overall contribution of the gradients of all locations in the current area to the detection results. Positions with a gradient greater than the mean are more important, and positions with a gradient less than the mean are less important. Therefore, the preliminary screening retains the positions with a gradient greater than the mean. Then the standard deviation is used to further screen the positions that retain disturbances. The standard deviation indicates the difference in the degree of influence of the gradient of each position in the current area on the detection results. A high standard deviation indicates that the gradient distribution of these positions is uneven, and the importance varies greatly. Most of the contributions to the detection results are concentrated in a small number of positions with high gradients, so a higher threshold screening is needed. A low standard deviation indicates that the gradient distribution of these important positions is relatively average, and the importance of each position is not very different. That is, the contribution of these positions to the detection results is not very different. At this time, a lower threshold is used to retain the disturbance of more positions.

To ensure that the gradient attack achieves the best effect, KPD is triggered when the adversarial example obtained by the gradient attack makes the detection model unable to detect any target. The detail of the whole algorithm is shown in Algorithm 1.

Algorithm 1 CamoNet and key pixel distillation

Input: Objective function J; an input image $\boldsymbol{I}$; a perturbation of image $\boldsymbol{\Delta}$; bounding boxes *bboxes*; maximum iterations T; warm up iterations W; score threshold t; step α; direction of diversification $\boldsymbol{w}_{\mathrm{d}}$; normalized perturbation vector $v_{\mathrm{DSI}(\cdot)}$; warm up step η_{ODI}; postprocessing function $P(\cdot)$; iterations to recover the last mask Z.

Output: the adversarial example $\boldsymbol{I}^{'}$

generate mask $\boldsymbol{M}$ by taking *bboxes* as the masks.
$\boldsymbol{\Delta} \leftarrow \mathbf{0}, i \leftarrow 0, n \leftarrow 1, i_{zero} \leftarrow 0, \boldsymbol{M_{Last}} \leftarrow None$
while $i < T$ **and** $n > 0$ **do**
  $\boldsymbol{I} \leftarrow \boldsymbol{I}^{'}$
  if $i < W$ **then**
    $\boldsymbol{\Delta} \leftarrow \boldsymbol{\Delta} - \eta_{\mathrm{ODI}} \cdot sign(\nabla_{\boldsymbol{I}} v_{\mathrm{DSI}}(\boldsymbol{I}, J, \boldsymbol{w}_{\mathrm{d}}))$
  else
    $\boldsymbol{\Delta} \leftarrow \boldsymbol{\Delta} - \alpha \cdot sign(\nabla_{\boldsymbol{I}} J)$
  end if
  $\boldsymbol{\Delta} \leftarrow max(0, min(\boldsymbol{\Delta}, 255))$
  $n \leftarrow$ number of bounding boxes
  $i \leftarrow i + 1$
  $i_{zero} \leftarrow i_{zero} + 1$
  if $n = 0$ **then**
    $\boldsymbol{M_{Last}} \leftarrow \boldsymbol{M}$
    $\boldsymbol{M} \leftarrow P(\boldsymbol{M})$
    $i_{zero} \leftarrow 0$
  end if
  if $i_{zero} = Z$ **and** $\boldsymbol{M_{Last}} \neq None$ **then**
    $\boldsymbol{M} \leftarrow \boldsymbol{M_{Last}}$
    $i_{zero} \leftarrow 0$
  end if
  $\boldsymbol{I}' \leftarrow \boldsymbol{I} \otimes (1 - \boldsymbol{M}) + \boldsymbol{\Delta} \otimes \boldsymbol{M}$
end while
return $\boldsymbol{I}^{'}$

3.3. Dataset

DOTA [42] is a large-scale remote sensing dataset for oriented object detection in RSIs, comprising 15 different object categories. Figure 4 shows the visualization results of the

DOTA dataset. This dataset contains 2806 multiresolution RSIs obtained from different optical satellites. In the experiments, the original images in the DOTA dataset are cropped into 1024 × 1024 patches with an overlap of 256 pixels because the original images are of different and large sizes, which are unsuitable for training the object detector. *DIOR* [43] is another large-scale open-source dataset in RSI, which consists of 20 different object categories. Figure 5 shows the visualization results of the DIOR dataset. This dataset contains 23,463 aerial images obtained from different sensors and platforms with multiple resolutions. All the shapes of the images are 800 × 800 pixels. Table 1 shows the specific details of the two datasets. It is worth mentioning that this paper only uses horizontal box annotation.

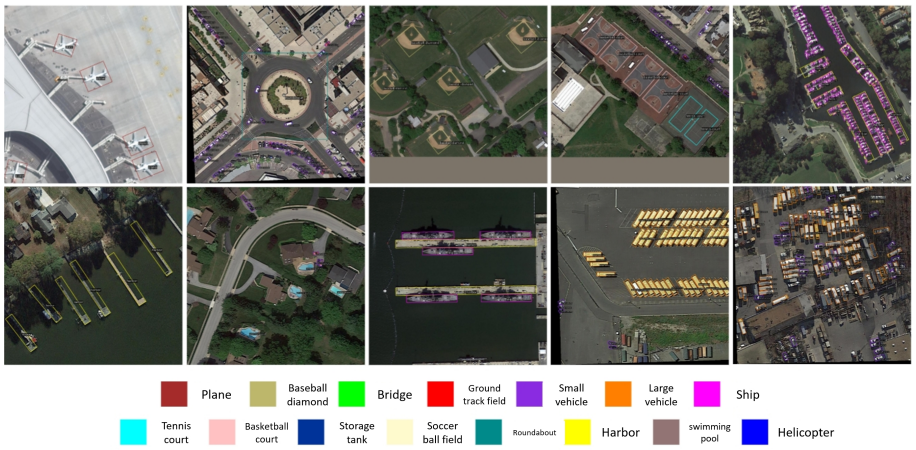

Figure 4. The visualization results of the DOTA dataset.

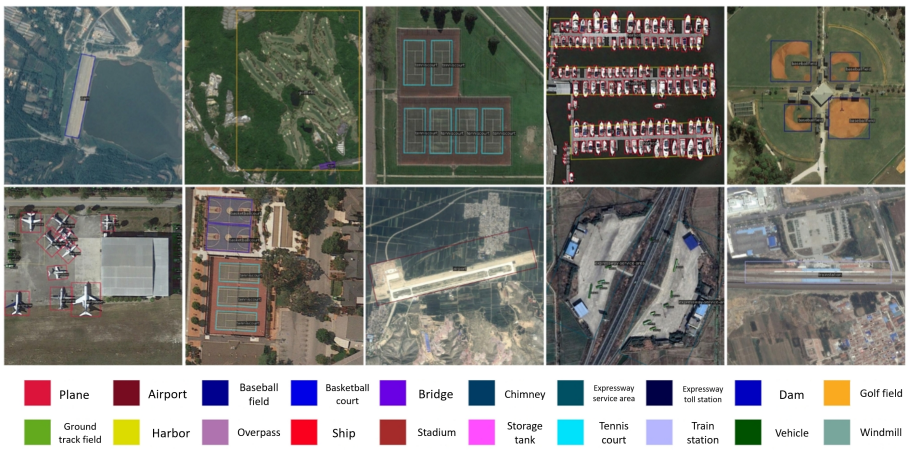

Figure 5. The visualization results of DIOR dataset.

Table 1. Details of the dataset used in this article.

Dataset	Categories	Images	Instances	Image Width	Year
DOTA	15	2806	188,282	800∼13,000	2018
DIOR-R	20	23,463	192,512	800	2021

3.4. Evaluation Metrics

We use average precision (AP) to quantitatively evaluate the detection performance. The detection result is true if the IoU overlap ratio between the ground truth and the prediction box is greater than 0.5. Otherwise, the prediction box is considered a false positive. In addition, if multiple prediction boxes overlap with the same ground truth, only one box with the highest score is considered a true positive, while the other boxes are

false positives. Precision measures the proportion of true positive detections, and recall measures the detection coverage of the detector for all the targets to be detected.

$$P = \frac{N_{tp}}{N_{pred}}, \tag{9}$$

$$R = \frac{N_{tp}}{N_{targets}}, \tag{10}$$

where N_{pred} represents the total number of predicted boxes, N_{tp} is the number of targets correctly detected, and N_{target} denotes the actual number of targets.

As soon as the threshold of IoU is determined, we can draw the model's precision–recall curve (PRC). The AP metric quantitatively evaluates the combined detection performance of the detector by calculating the area under the PRC as follows:

$$AP = \int_0^1 P(R)dR. \tag{11}$$

the higher the AP value is, the better the performance, and vice versa.

For a fair quantitative comparison, PSNR [44] is used for object-vanishing attack assessment. The evaluation indicator is calculated as follows:

$$MSE = \frac{1}{MN}\sum_{i-1}^{M}\sum_{j-1}^{N}(x(i,j) - y(i,j))^2 \tag{12}$$

$$PSNR(dB) = 10 \cdot \log_{10}(\frac{MAX^2}{MSE}), \tag{13}$$

where x and y are images of size $M \times N$ and the corresponding adversarial example, respectively, and MAX^2 is the maximum pixel value of the image. The higher the value of PSNR, the lower the image noise. In the field of image compression and reconstruction, it is believed that when PSNR is close to 50, the image error is very small. When PSNR is between 30 and 50, it is difficult for human eyes to detect the image difference.

4. Experimental Results

In the experiments, ImageNet pretrained weights are used to initialize the backbone. Most models are trained in 12 periods with a batch size of 2. RTMDet [45] is trained in 36 epochs. This paper uses stochastic gradient descent (SGD) as the default optimizer. The weight decay and momentum of SGD are 1.0×10^{-4} and 0.9, respectively. The initial learning rate is set to 2.5×10^{-3}. The learning rate decays by a factor of 0.1 at 8 and 11 epochs. Our experiments are carried out on an Ubuntu system with one Tesla V100 GPU.

4.1. Objective Function Studies

In this part of the experiment, we use different objective functions to attack the Faster R-CNN object detectors trained on the DOTA training set and then compare and evaluate their attack performance on 100 images selected from the DOTA verification set. Figure 6 provides an illustration to intuitively show the difference between the two objective functions. The first row show the raw images, the second row is the visualization of the detection results, and the third row is the heatmaps obtained by AblationCAM. The vanishing loss eliminates the ability to recognize the target the victim model should have. The mislabeling loss fools the detector into mislabeling detected objects. It can be seen from the heatmaps that the model's attention after the vanishing loss completely disappears so that the model cannot detect any target. After the mislabeling loss, the model's attention is distracted, resulting in many false alarms.

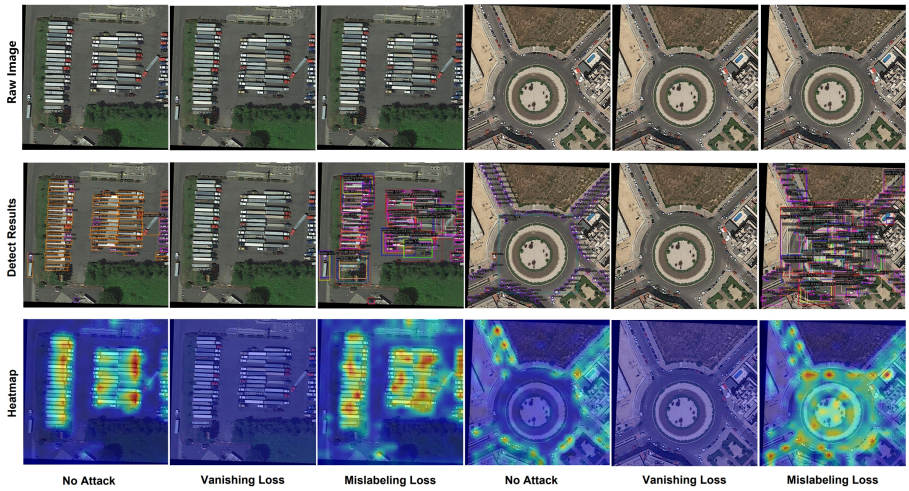

Figure 6. Attacks with two different objective functions. In the heatmap, red represents a higher response, while blue represents a lower response. Green indicates a corresponding degree between red and blue.

The quantitative results are shown in Table 2. It turns out that mislabeling and vanishing losses drastically reduce the AP of the victim detector. The AP of mislabeling loss reaches a score of 21.2 (decreased by 69.4%). It is worth noting that the vanishing loss breaks down the detection capability of the victim detector by reducing AP from 69.3% to 0.6% (decreased by 99.1%). It also shows that when mislabeling and vanishing loss are applied simultaneously, the result can be worse than using vanishing loss alone. This is because the two objective functions are mutually exclusive. Compared with the perfect detection results under no attack, the vanishing loss removes the victim's ability to recognize any object. The mislabeling loss fools the detector into mislabeling detected objects (e.g., a vehicle as an airplane). Since vanishing loss is more suitable for camouflage and hiding targets, the following experiments use vanishing loss as the default object function.

Table 2. Comparison of different objective functions on the AP.

Object Functions	$AP \downarrow$
No Attack	69.3
Mislabeling Loss	21.2
Vanishing Loss	0.6
Vanishing and Mislabeling Loss	20.9

4.2. Gradient Attack Studies

In this part of the experiment, we use different adversarial attack methods to attack the Faster R-CNN object detectors trained on the DOTA training set and then compare and evaluate their attack performance on 100 images selected from the DOTA verification set. Since the object detection attack framework is based on the gradient, these gradient-based adversarial attack methods widely used in classification tasks can be directly used. The results are shown in Table 3. PGD and DIS-PGD obviously decrease the AP index. The AP of PGD reaches a score of 0.6, and the AP of DIS-PGD reaches a score of 0.2. Compared with PGD, DIS-PGD sacrifices PSNR but has a faster attack speed. However, the PSNR is high enough to make it difficult for human eyes to detect differences in images. Although FGSM has a much higher PSNR and attack speed than other methods, its attack success rate is low. The attack speed of MI-FGSM is the slowest among the four methods. Given the priority of the attack success rate, DIS-PGD is chosen as the default adversarial attack method in the following experiments.

Table 3. Comparison of the PSNR, time cost, and AP of different adversarial attack methods.

Attacks	PSNR ↑	Time Cost (s/img) ↓	AP(%) ↓
No Attack	-	-	69.3
FGSM	57.79	1.5	56.0
MI-FGSM	35.62	23.8	2.4
PGD	46.73	17.9	0.6
DIS-PGD	38.84	16.5	0.2

4.3. Postprocessing Studies

To further improve the success rate of ground target camouflage, the KPD module is introduced, which can reduce the disturbance rate and make camouflage more difficult to detect. To explore different postprocessing strategies, the proposed target camouflage network is applied to the DOTA dataset. The results are shown in Table 4. The entire bounding box areas are taken as masks when no postprocessing strategy is specified. It has the highest pixel perturbation rate of 9.2%. When postprocessing operations are performed, the model selects these critical pixels, and the disturbance rate further decreases. A 3× mean value (global) means that the model retains pixels larger than the 3× mean value of all mask pixels. A 3× mean value (box) means the model retains pixels larger than the 3× mean value of mask pixels of the current bounding box. Therefore, the latter has a higher time complexity and slower speed than the former. Adaptive indicates that the model adaptively retains pixels larger than the sum of the mean and mean square error of the mask pixels of the current bounding box, which has a dynamic threshold. Surprisingly, although the three strategies reduce the perturbation rate, they do not increase *AP*, which means that the attack success rate is not affected.

Table 4. Comparison of different postprocessing strategies on DOTA dataset.

Postprocessing	Perturbation (%) ↓	PSNR ↑	Time Cost (s/img) ↓	*AP* ↓
None	9.2	42.5	10.1	1.1
3× Mean value (global)	3.8	44.3	16.3	1.1
3× Mean value (box)	2.7	44.7	19.1	1.1
Adaptive	5.6	41.7	32.1	1.1

To show the differences between different postprocessing strategies more intuitively, the enlarged visible results are shown in Figure 7. The PSNR of the 3× mean value (box) is the highest among the three postprocessing methods, and is 44.7. It is difficult for the naked eye to find the added disturbance of the 3× mean value (box). Although KPD reduces the pixel perturbation rate, its disturbance seems more obvious than that without adding a postprocessing module. It is worth mentioning that these disturbances are so subtle that it is difficult to find them directly from large RSIs.

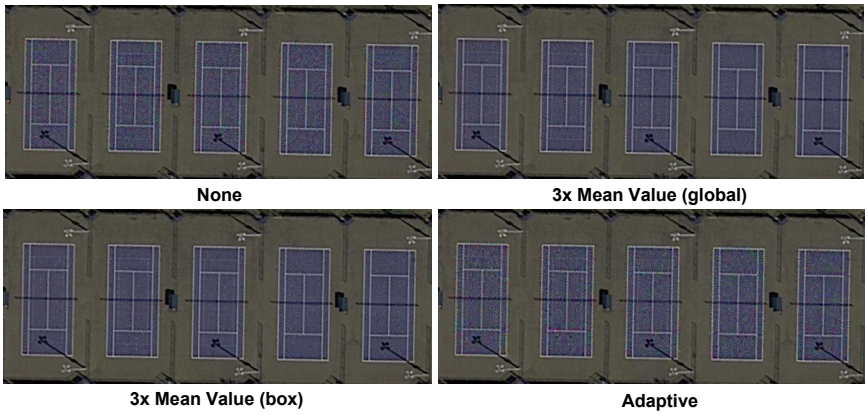

Figure 7. Visualizing different postprocessing strategies on the DOTA dataset.

To further verify the generality of the KPD module, the proposed target camouflage network is also applied to the DIOR datasets. The results are shown in Table 5. We also visualize the enlarged results on the DIOR dataset in Figure 8 and find that the rules are consistent with those on the DOTA dataset. The PSNR of the 3× mean value (global) is the highest among the three postprocessing methods, and is 38.6. It is worth mentioning that the KPD strategy reduces the perturbation rate and AP, while the 3× mean value strategy improves AP. This may be because the filter threshold of the 3× mean value strategy is fixed, and parameters need to be adjusted based on different datasets. However, the filter threshold of the adaptive strategy is automatically adjusted so it has stronger generality.

Table 5. Comparison of different postprocessing strategies on DIOR dataset.

Postprocessing	Perturbation (%) ↓	PSNR ↑	Time Cost (s/img) ↓	AP ↓
None	20.7	34.6	**8.1**	2.1
3× Mean value (global level)	4.1	38.6	17.1	12.6
3× Mean value (box level)	5.3	38.2	19.6	10.8
Adaptive	11.1	33.2	33.7	1.3

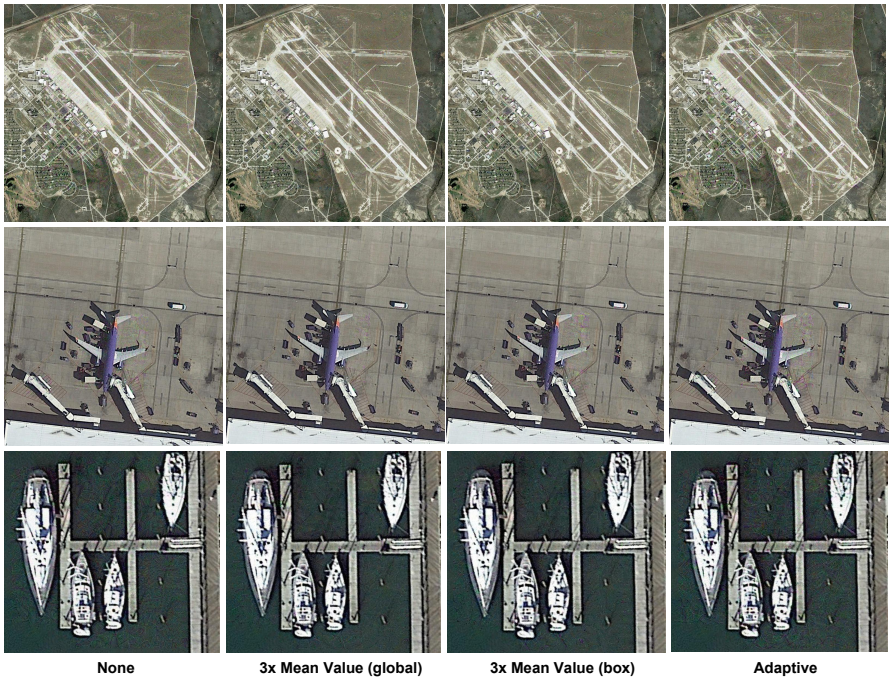

Figure 8. Visualizing different postprocessing strategies on the DIOR dataset.

4.4. Image Forensics Studies

As shown in Figure 9 We applied four traditional image forensic techniques [46] to the attacked images to check if they can detect changes in the image. They include clone detection, error level analysis, noise analysis, and principal component analysis. The following is a brief introduction to these four methods:

Clone Detection—The clone detector highlights copied regions within an image. These can be a good indicator that a picture has been manipulated.

Error Level Analysis—This tool compares the original image to a recompressed version. This can make manipulated regions stand out in various ways. For example, they can be darker or brighter than similar regions that have not been manipulated.

Noise Analysis—This tool is basically a reverse denoising algorithm. Rather than removing the noise, it removes the rest of the image. It uses a super simple separable median filter to isolate the noise. It can be useful for identifying manipulations of the image such as airbrushing, deformations, warping, and perspective-corrected cloning.

Principal Component Analysis—This tool performs principal component analysis on the image. This provides a different angle to view the image data, which makes discovering certain manipulations and details easier.

By comparing the results of the attacked images and the raw images, we found that the images with the added "small" disturbance only differ from the original images in error level analysis and noise analysis, but still cannot determine whether the image has been modified. This indicates that these traditional forensic techniques cannot detect attacks generated by CamoNet.

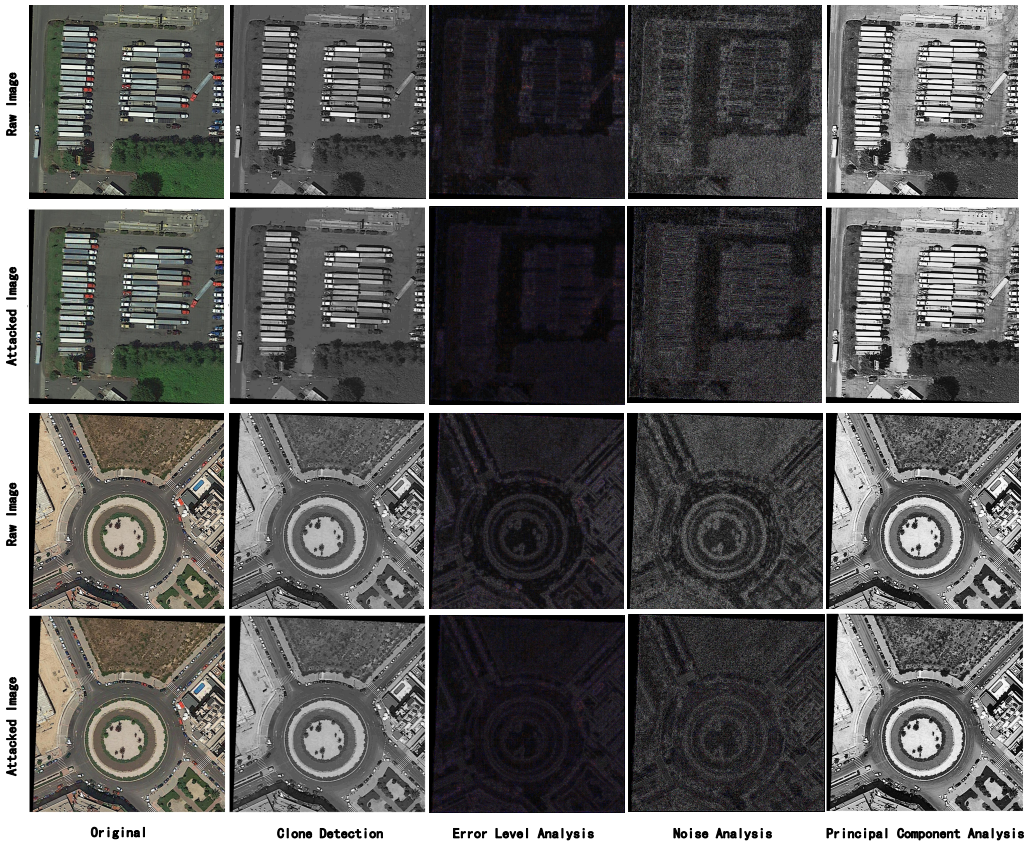

Figure 9. Image forensics.

5. Discussion

In this section, we discuss the transferability of CamoNet. Specifically, we use a target camouflage network to attack different object detectors trained on the DOTA training set and then compare and evaluate their attack performance on the full DOTA verification set. Specifically, the target camouflage network is applied to three dominant detection algorithms (RetinaNet, Faster R-CNN, and RTMDet) and three backbones (ResNet-50, swin-t, and CSPNext-t). The results are shown in Table 6. AP_{Gap} equals AP_{Clean} minus AP_{Attack}, which can measure the effect of target camouflage. Compared to the clean AP_{Clean}, the target camouflage network drastically reduces the attack AP_{Attack} of all victim detectors. For instance, vanishing attacks break down the detection capability of the four detectors: RetinaNet (ResNet-50), Faster R-CNN (ResNet-50), Faster R-CNN (swin-t), and RTMDet (CSPNext-t), by reducing their mAP from 60.5%, 66.0%, 68.9%, and 64.4% to 0.7%, 1.1%, 0.5%, and 0.0%, respectively. Faster R-CNN has a faster attack speed and higher PSNR than RetinaNet. Swin-t has a faster attack speed and lower AP_{Attack} than ResNet-50. It is worth mentioning that the AP_{Attack} of RTMDet can reach 0, which means the attack success rate is 100%. The experimental results show that our method can work on various types of detectors and verify the generality of the target camouflage network.

Table 6. Applying the proposed target camouflage network to different detectors on the DOTA dataset.

Detectors	Backbone	PSNR ↑	Time (s/img) ↓	AP_{Clean} ↑	AP_{Attack} ↓	AP_{Gap} ↑
RetinaNet [12]	r50	36.8	18.7	60.5	0.7	59.8
Faster R-CNN [13]	r50	42.5	10.1	66.0	1.1	64.9
Faster R-CNN	swin-t [47]	42.3	5.6	68.9	0.5	68.4
RTMDet [45]	cspnext-t	38.2	6.7	64.4	0.0	64.4

To study the transferability, we use the adversarial examples generated by the attack detector to attack different victim detectors. Both victim detectors and attack detectors are trained on the DOTA training set and evaluated on the DOTA verification set. The results are shown in Table 7. When the victim detector is the attack detector, which is usually called a white-box attack in the field of adversarial attack, the effect of target camouflage is the best. When the victim detector and the attack detector are different, which is usually called a black-box attack, the ability of target camouflage declines. After the attack of four different detectors, the minimum value of AP_{Gap} of the victim detector is 26.4%, 21.7%, 18.5%, and 21.1%, respectively, which means that the target camouflage network does not completely fail even under the black-box attack.

Table 7. Study of the transferability of the target camouflage network between different detectors.

Attack Detector	Victim Detector	AP_{Clean} ↑	AP_{Attack} ↓	AP_{Gap} ↑
RetinaNet	RetinaNet	60.5	0.7	59.8
	Faster R-CNN (r50)	66.0	8.7	57.3
	Faster R-CNN (swin-t)	68.9	36.8	32.1
	RTMDet	64.4	38.0	26.4
Faster R-CNN (r50)	RetinaNet	60.5	19.2	41.3
	Faster R-CNN (r50)	66.0	1.1	64.9
	Faster R-CNN (swin-t)	68.9	47.2	21.7
	RTMDet	64.4	42.5	21.9
Faster R-CNN (swin-t)	RetinaNet	60.5	39.4	21.1
	Faster R-CNN (r50)	66.0	43.8	22.2
	Faster R-CNN (swin-t)	68.9	0.5	68.4
	RTMDet	64.4	45.9	18.5
RTMDet	RetinaNet	60.5	39.4	21.1
	Faster R-CNN (r50)	66.0	42.8	23.2
	Faster R-CNN (swin-t)	68.9	44.1	24.8
	RTMDet	64.4	0.0	64.4

In addition, we study the transferability of different postprocessing strategies on the DOTA dataset. The results are shown in Table 8. We use the adversarial examples generated by the Faster R-CNN detector to attack different victim detectors and apply postprocessing strategies to them. AP_{Post} indicates the AP of the victim detector after the postprocessing strategy is adopted. AP_{Gap} equals AP_{Attack} minus AP_{Post}, which can measure the effectiveness of the postprocessing strategy. Total AP_{Gap} equals the sum of four victim detectors' AP_{Gap} values, which can measure the transferability of the postprocessing strategy. The 3× mean value strategies have negative AP_{Gap}. This means that although they reduce the perturbation rate, they also weaken the effect of target camouflage. In contrast, adaptive strategy has a positive AP_{Gap}. The AP_{Gap} of RetinaNet, Faster R-CNN (r50), Faster R-CNN (swin-t), and RTMDet are 1.9%, 0.0%, 1.7%, and 0.9%, respectively. This means that the KPD strategy reduces the perturbation rate and enhances the transferability of the target camouflage network.

Table 8. Study of the transferability of different postprocessing strategies on the DOTA dataset.

Postprocessing	Victim Detector	AP_{Attack}	AP_{Post}	$AP_{Gap}\downarrow$	Total $AP_{Gap}\uparrow$
Mean value (global level)	RetinaNet	19.2	24.3	-5.1	-17.0
	Faster R-CNN (r50)	1.1	1.1	0.0	
	Faster R-CNN (swin-t)	47.2	52.3	-5.1	
	RTMDet	42.5	49.3	-6.8	
Mean value (box level)	RetinaNet	19.2	22.7	-3.5	-13.0
	Faster R-CNN (r50)	1.1	1.1	0.0	
	Faster R-CNN (swin-t)	47.2	51.2	-4.0	
	RTMDet	42.5	48.0	-5.5	
Adaptive	RetinaNet	19.2	17.3	1.9	4.5
	Faster R-CNN (r50)	1.1	1.1	0.0	
	Faster R-CNN (swin-t)	47.2	45.5	1.7	
	RTMDet	42.5	41.6	0.9	

However, CamoNet inevitably has some disadvantages that need to be addressed. In the gradient attack phase, when using adversarial samples generated by a certain model to attack other models, the success rate of the attack decreases significantly. In the postprocessing phase, although adaptive strategies have better mobility, their disturbance rate is higher than strategies with fixed thresholds. This means that this postprocessing strategy is more easily visible to the naked eye.

6. Conclusions

In this paper, we provide a new view of target camouflage for objection detection in RSI. Unlike traditional methods, the new target camouflage occurs after optical imaging. A novel target camouflage network is proposed, which can successfully hide targets with only subtle perturbations. We propose a detection space initialization strategy to maximize diversity in the detector's outputs among the generated samples, which enhances the performance of gradient attacks for object detection. We also design a key pixel distillation module that adaptively filters out unimportant perturbation pixels. It can further reduce the range of disturbance while ensuring the attack effect, making the attack more concealed and imperceptible to the naked eye. Our network can be implemented on local storage servers or edge computing devices with practical application scenarios. The experimental results on DOTA and DIOR verify the superiority of the proposed target camouflage network.

In future work, we will consider how to design better gradient attack and postprocessing strategies to enable CamoNet to have more extensive application scenarios.

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

Funding: This research was funded by National Natural Science Foundation of China under Grants 61971279, 62022054, and U2230201.

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

References

1. Ghamisi, P.; Rasti, B.; Yokoya, N.; Wang, Q.; Hofle, B.; Bruzzone, L.; Bovolo, F.; Chi, M.; Anders, K.; Gloaguen, R.; et al. Multisource and Multitemporal Data Fusion in Remote Sensing: A Comprehensive Review of the State of the Art. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 6–39. [CrossRef]
2. Xu, Y.; Du, B.; Zhang, L.; Cerra, D.; Pato, M.; Carmona, E.; Prasad, S.; Yokoya, N.; Hänsch, R.; Le Saux, B. Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 IEEE GRSS data fusion contest. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2019**, *12*, 1709–1724. [CrossRef]

3. Shi, J.; Liu, W.; Shan, H.; Li, E.; Li, X.; Zhang, L. Remote Sensing Scene Classification Based on Multibranch Fusion Attention Network. *IEEE Geosci. Remote. Sens. Lett.* **2023**, *20*, 3001505. [CrossRef]
4. Hou, Y.-E.; Yang, K.; Dang, L.; Liu, Y. Contextual Spatial-Channel Attention Network for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 6008805. [CrossRef]
5. Sun, S.; Yang, Z.; Ma, T. Lightweight Remote Sensing Road Detection Network. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6510805. [CrossRef]
6. Zhou, J.; Zhang, R.; Zhao, W.; Shen, S.; Wang, N.; APS-Net: An Adaptive Point Set Network for Optical Remote-Sensing Object Detection. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 6001405. [CrossRef]
7. Ghorbanzadeh, O.; Tiede, D.; Wendt, L.; Sudmanns, M.; Lang, S. Transferable instance segmentation of dwellings in a refugee camp-integrating CNN and OBIA. *Eur. J. Remote Sens.* **2021**, *54*, 127–140. [CrossRef]
8. Pan, S.; Tao, Y.; Nie, C.; Chong, Y. PEGNet: Progressive Edge Guidance Network for Semantic Segmentation of Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 637–641. [CrossRef]
9. Shi, F.; Zhang, T. An Anchor-Free Network With Box Refinement and Saliency Supplement for Instance Segmentation in Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6516205. [CrossRef]
10. Kokila, S.; Jayachandran, A. Hybrid Behrens-Fisher- and Gray Contrast–Based Feature Point Selection for Building Detection from Satellite Images. *J. Geovis. Spat. Anal.* **2023**, *7*, 8. [CrossRef]
11. Ghasemloo, N.; Matkan, A.A.; Alimohammadi, A.; Aghighi, H.; Mirbagheri, B. Estimating the Agricultural Farm Soil Moisture Using Spectral Indices of Landsat 8, and Sentinel-1, and Artificial Neural Networks. *J. Geovis. Spat. Anal.* **2022**, *6*, 19. [CrossRef]
12. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef]
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
14. Ji, H.; Gao, Z.; Mei, T.; Li, Y. Improved Faster R-CNN With Multiscale Feature Fusion and Homography Augmentation for Vehicle Detection in Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1761–1765. [CrossRef]
15. Ji, H.; Gao, Z.; Mei, T.; Ramesh, B. Vehicle Detection in Remote Sensing Images Leveraging on Simultaneous Super-Resolution. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 676–680. [CrossRef]
16. Czaja, W.; Fendley, N.; Pekala, M.; Ratto, C.; Wang, I. Adversarial examples in remote sensing. In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '18), Seattle, WA, USA, 6–9 November 2018; pp. 408–411.
17. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
18. Xu, Y.; Ghamisi, P. Universal Adversarial Examples in Remote Sensing: Methodology and Benchmark. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5619815. [CrossRef]
19. 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.* **2021**, *59*, 1604–1617. [CrossRef]
20. Lu, J.; Sibai, H.; Fabry, E. Adversarial examples that fool detectors. *arXiv* **2017**, arXiv:1712.02494.
21. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May, 2015.
22. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
23. Dong, Y. Boosting Adversarial Attacks with Momentum. In Proceedings of the IEEE/CVF Conference Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 9185–9193.
24. 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.
25. Chen, S.; Cornelius, C.; Martin, J.; Chau, D. Robust physical adversarial attack on faster R-CNN object detector. *arXiv* **2018**, arXiv:1804.05810.
26. Li, Y.; Tian, D.; Chang, M.; Bian, X.; Lyu, S. Robust adversarial perturbation on deep proposal-based models. *arXiv* 2018, arXiv:1809.05962.
27. Zhang, H.; Zhou, W.; Li, H. Contextual adversarial attacks for object detection. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
28. Li, Y.; Bian, X.; Lyu, S. Attacking object detectors via imperceptible patches on background. *arXiv* **2018**, arXiv:1809.05966.
29. 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.
30. Wang, Y.; Tan, Y.A.; Zhang, W.; Zhao, Y.; Kuang, X. An adversarial attack on DNN-based black-box object detectors. *J. Network Comput. Appl.* **2020**, *161*, 102634. [CrossRef]
31. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. Neur. Net. Lear.* **2019**, *30*, 2805–2824. [CrossRef]
32. Sun, X.; Cheng, G.; Pei, L.; Li, H.; Han, J. Threatening Patch Attacks on Object Detection in Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5609210. [CrossRef]

33. Lian, J.; Wang, X.; Su, Y.; Ma, M.; Mei, S. CBA: Contextual Background Attack Against Optical Aerial Detection in the Physical World. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5606616. [CrossRef]
34. Yu, Y.; Lee, H.J.; Lee, H.; Ro, Y.M. Defending Person Detection Against Adversarial Patch Attack by Using Universal Defensive Frame. *IEEE Trans. Image Proc.* **2022**, *31*, 6976–6990. [CrossRef]
35. Kang, C.; Dong, Y.; Wang, Z.; Ruan, S.; Su, H.; Wei, X. DIFFender: Diffusion-Based Adversarial Defense against Patch Attacks in the Physical World. *arXiv* **2023**, arXiv:2306.09124.
36. Skelhorn, J.; Candy Rowe, C. Cognition and the evolution of camouflage. *Proc. R. Soc. Biol. Sci.* **2016**, *283*, 1825. [CrossRef]
37. 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]
38. Zhang, Y.; Zhang, Y.; Qi, J.; Bin, K.; Wen, H.; Tong, X.; Zhong, P. Adversarial Patch Attack on Multi-Scale Object Detection for UAV Remote Sensing Images. *Remote Sens.* **2022**, *14*, 5298. [CrossRef]
39. Tang, G.; Yao, W.; Jiang, T.; Zhou, W.; Yang, Y.; Wang, D. Natural Weather-Style Black-Box Adversarial Attacks Against Optical Aerial Detectors. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5620911. [CrossRef]
40. Li, Y.; Fang, Y.; Li, W.; Jiang, B.; Wang, S.; Li, Z. Learning Adversarially Robust Object Detector with Consistency Regularization in Remote Sensing Images. *Remote Sens.* **2023**, *15*, 3997. [CrossRef]
41. Huang, H.; Wang, Y.; Chen, Z.; Tang, Z.; Zhang, W.; Ma, K.K. Rpattack: Refined Patch Attack on General Object Detectors. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; pp. 1–6.
42. Xia, G.; Bai, X.; Ding, J.; Zhu, Z.; Serge, B.; Luo, J.; Mihai, D.; Marcello, P.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
43. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photo. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
44. Zhou, W.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Proc.* **2004**, *13*, 600–612.
45. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *arXiv* **2022**, arXiv:2212.07784.
46. Lukas, J.; Fridrich, J.; Goljan, M. Digital camera identification from sensor pattern noise. *IEEE Trans. Inf. Forensics Secur.* **2006**, *1*, 205–214. [CrossRef]
47. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 9992–10002.