Benchmarking Adversarial Patch Against Aerial Detection

Jiawei Lian, *Graduate Student Member, IEEE*, Shaohui Mei, *Senior Member, IEEE*, Shun Zhang, *Member, IEEE*, and Mingyang Ma, *Graduate Student Member, IEEE*

***Abstract*— Deep neural networks (DNNs) have become essential for aerial detection. However, DNNs are vulnerable to adversarial examples, which pose great security concerns for security-critical systems. Researchers recently devised adversarial patches to evaluate the vulnerability of DNN-based aerial detection methods physically. Nonetheless, adversarial patches generated by the existing algorithms are not strong enough and extremely time-consuming. Moreover, complicated physical factors are not accommodated well during the optimization process. In this article, a novel adaptive-patch-based physical attack (AP-PA) framework is proposed to alleviate the above problems, which achieves state-of-the-art performance in both accuracy and efficiency. Specifically, AP-PA aims to generate adversarial patches that are adaptive in both physical dynamics and varying scales, and by which the particular targets can be hidden from being detected. Furthermore, the adversarial patch is also gifted with attack effectiveness against all targets of the same class with a patch outside the target (no need to smear targeted objects) and robust enough in the physical world. In addition, a new loss is devised to consider more available information of detected objects to optimize the adversarial patch, which can significantly improve the patch's attack efficacy (average precision drop up to 87.86% and 85.48% in white-box and black-box settings, respectively) and optimizing efficiency. We also establish one of the first comprehensive, coherent, and rigorous benchmarks to evaluate the attack efficacy of adversarial patches on aerial detection tasks. Finally, several proportionally scaled experiments are performed physically to demonstrate that the elaborated adversarial patches can successfully deceive aerial detection algorithms in dynamic physical circumstances. The code is available at https://github.com/JiaweiLian/AP-PA.**

***Index Terms*— Adaptive, adversarial examples, adversarial patch, aerial detection, benchmark, deep neural networks (DNNs), physical attack.**

I. Introduction

With the development of deep neural networks (DNNs), DNN-based aerial detection approaches [1], [2], [3], [4], [5], [6] have shown excellent performance both in accuracy and efficiency. However, DNNs are vulnerable to elaborately designed adversarial examples [7], [8]. By adding a small malicious perturbation to clean examples, the DNN-based systems can make a completely different prediction, which may cause severe consequences in some security-critical areas. In this context, adversarial robustness is regarded as the key performance of the DNN-based intelligent aerial detection systems. The study of new adversarial attack methods provides a data basis for improving the adversarial robustness of DNNs, which also provides ideas for explaining the vulnerability of DNNs to adversarial examples. Nonetheless, most of the existing adversarial attack methods focus on digital attacks and individual object detectors. In addition, attacking object detectors is more challenging than attacking image classifiers, especially extending the digital attack to the physical world, because it requires the adversarial perturbation to be robust enough to survive real-world distortions from many uncontrollable physical dynamics.

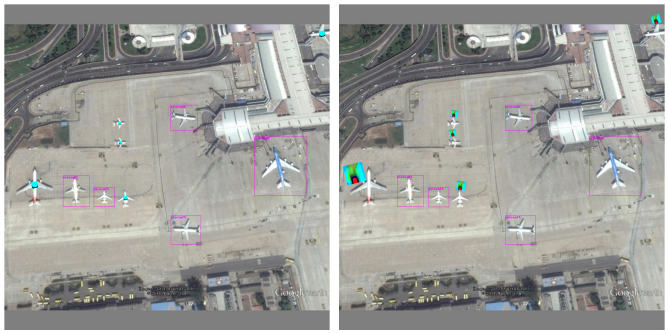
In the physical world, however, the DNN-based aerial detection systems work by directly scanning objects. So most of the existing works change the object's appearance in the physical scenarios to provide adversarial examples to the remote sensing detection devices, which poses great challenges, especially needs to solve complicated physical conditions such as different viewing distances, object scales, and lighting conditions. To make adversarial examples practical in real scenarios, some latest works propose the adversarial patch [9]. They elaborate an adversarial patch that does not attempt to sway an existing object to another subtly. Instead, this attack method generates an image-agnostic patch that is extremely salient to a neural network. This patch can then be pasted anywhere within the field of view of the classifier and renders the classifier to predict a targeted wrong class. Up to now, adversarial patches have been applied to different tasks, such as face recognition [10], [11], [12], object detection [13], [14], pedestrian detection [15], [16], image retrieval [12], [17], and aerial detection [18], [19].

Albeit the great success of adversarial patches for physical attacks, they have several limitations. First, the procedure of generating an adversarial patch is extremely time-consuming, because the adversarial patch is iteratively optimized on a large amount of data. Second, the adversarial patch will face a complex transformation from the digital domain transferring to the physical world. Hence, these operations lead to high computation costs. Third, the patch's pixel values will inevitably become distorted due to the limitation of patch printing devices and image capture devices. Finally, the current adversarial patches are painted or pasted on the surface of objects, which is not flexible and convenient enough to be

Manuscript received 2 September 2022; revised 20 October 2022; accepted 24 November 2022. Date of publication 28 November 2022; date of current version 8 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62171381 and Grant 62271409 and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding author: Shaohui Mei.)*

The authors are with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China (e-mail: lianjiawei@mail.nwpu.edu.cn; meish@nwpu.edu.cn; szhang@nwpu.edu.cn; mamingyang@mail.nwpu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2022.3225306



(a) (b)

Fig. 1. Attack effectiveness of adversarial patches with different positions (patches are pasted on targets or placed outside targets). (a) Patches on targets. (b) Patches outside targets.

applied in real scenarios and is prone to arouse human suspicion.

Considering the above reasons, this article dedicates to solving the following problems: under real physical scenarios, how to generate the adversarial patch to achieve stealthy attack effectiveness easily; in addition, the adversarial patch is robust to complex physical changes and can be used flexibly and conveniently.

Technically, to search for the appropriate physical attack framework, we propose to construct a novel state-of-the-art method to generate an adversarial patch to hide the targets from being detected, as shown in Fig. 1. We devote to designing a physically robust adversarial patch, in which the physical varying factors and different object scales are properly accommodated and the patch is gifted with strong attack effectiveness against all the targets of the same class. Moreover, we make full use of the information from all the detected objects to optimize the adversarial patch, which can significantly improve the adversarial patch's attack efficacy and optimize efficiency. To make a comprehensive evaluation, we also establish one of the first coherent and rigorous benchmarks to evaluate the attack efficacy of adversarial patches on aerial detection tasks. Our method is comprehensively verified in several state-of-the-art object detection frameworks, such as one-stage detectors, two-stage detectors, convolutional neural network (CNN)-based detectors, and Transformer-based detectors. Extensive experiments demonstrate that the proposed method is effective and robust in complex physical conditions and has a certain transferability for different aerial object detectors.

In summary, our contributions are fourfold.

1) A novel adaptive-patch-based framework AP-PA is devised to conduct the physical attacks and it achieves state-of-the-art attack performance. On one hand, our method can elaborate an adaptive adversarial patch accommodating both the physical dynamics and varying scales to hide the particular targets from being detected. On the other hand, the adversarial patch is gifted with strong attack effectiveness against all the targets of the same class. In addition, the patch can be easily and conveniently used in real scenarios, simply placed beside targets, making the attack happen.
2) A new objective loss is proposed to make full use of the detected information, which can not only accelerate the optimization process of the adversarial patch but also strengthen its attack efficacy both in the white-box [average precision (AP) drop up to 87.86%] and black-box settings (AP drop up to 85.48%). Moreover, the elaborated adversarial patches can also transfer their attack effectiveness well between different aerial detectors.
3) To the best of our knowledge, we are the first to comprehensively benchmark adversarial patches against several mainstream aerial detection methods with different frameworks (one-stage, two-stage, CNN-based, and Transformer-based detectors). In addition, we also delve into the impact of the resolution and location of the adversarial patch on attack efficacy.
4) We verify the proposed method on different types of object detectors, and the experimental results show that our method naturally maintains attack efficacy with a certain generalization between different detectors. In addition, we also conduct proportionally scaled validation experiments in the physical world, demonstrating that the adversarial patch crafted by AP-PA can be robust enough to fool aerial detectors in dynamic physical conditions successfully.

The remainder of this article is organized as follows. Section II briefly reviews the related work of adversarial attacks. Then, we introduce the details of the proposed framework AP-PA for generating adversarial patches against aerial detection tasks in Section III. We evaluate the proposed attack method and demonstrate the effectiveness of our generated adversarial patches in Section IV. Finally, we conclude our proposed AP-PA and discuss some future work concerning adversarial patches in Section V.

II. Related Work

In this section, we first provide the background knowledge of the adversarial attack. In addition, we also review the related works about digital attacks, physical attacks, and physical attacks in aerial detection, respectively.

A. Digital Attacks

Most existing works concerning adversarial attacks focus on image classification in the digital domain [7], [8], [20], [21], [22], [23], [24], [25]. Given an image classifier $f(\boldsymbol{x}) : \boldsymbol{x} \in \boldsymbol{X} \rightarrow y \in \boldsymbol{Y}$ that outputs a prediction y as a result for an input image $\boldsymbol{x}$, the purpose of the adversarial attack is to elaborate an adversarial example $\boldsymbol{x}^*$ near to clean example $\boldsymbol{x}$ but leading to the classifier making a wrong prediction. Technically, the adversarial attack methods can be divided into **nontargeted** and **targeted** ones according to the attacker's intentions. For a properly classified input image $\boldsymbol{x}$ with ground-truth label y such that $f(\boldsymbol{x}) = y$, the nontargeted attack methods design adversarial example $\boldsymbol{x}^*$ by adding imperceptible perturbation to clean images $\boldsymbol{x}$, but fools the classifier as $f(\boldsymbol{x}^*) \neq y$, mainly used for image classification, automatic driving, and object detection tasks; while the targeted adversarial attack methods

aim to misguide the classifier by predicting a particular label as $f(\boldsymbol{x}^*) = y^*$, where y^* is the target label specified by the attacker and $y^* \neq y$, which are often applied to attacking face recognition, image classification, and automatic driving tasks. Usually, the L_p norm is adopted as the visibility metric of the adversarial noise. For digital attack, the adversarial noise is required to be invisible to human eyes, namely, less than an allowed value ϵ as $\|\boldsymbol{x}^* - \boldsymbol{x}\|_p \leq \epsilon$.

The existing methods can be categorized into three types according to how the adversarial samples are generated. In this article, we focus on the nontargeted version of attack approaches, and the targeted version can be derived similarly.

1) Optimization-Based Methods: Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [7], Deepfool [26], C&W [27], etc. directly minimize the distance between clean and adversarial examples subject to the misclassification of adversarial examples, which can be defined as

$$\arg\min_{\theta} \lambda \cdot \|\boldsymbol{x}^* - \boldsymbol{x}\|_p - L\big(\boldsymbol{x}^*, y\big) \tag{1}$$

where $L(\boldsymbol{x}^*, y)$ is the loss function w.r.t. $\boldsymbol{x}^*$. Since it directly minimizes the distance between an adversarial example and the corresponding clean example, the L_p norm is not necessarily inferior to a specified value.

2) Gradient-Based One-Step Methods: The gradient-based one-step methods such as the fast gradient sign method (FGSM) [8] seek an adversarial example $\boldsymbol{x}^*$ by maximizing $L(\boldsymbol{x}^*, y)$. FGSM generates adversarial examples to meet the L_∞ norm limitation $\|\boldsymbol{x}^* - \boldsymbol{x}\|_p \leq \epsilon$ as

$$\boldsymbol{x}^* = \boldsymbol{x} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y)) \tag{2}$$

where $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y)$ is the gradient of the loss function w.r.t. $\boldsymbol{x}$. A generalization of FGSM is to meet the L_2 norm constraint $\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leq \epsilon$ as

$$\boldsymbol{x}^* = \boldsymbol{x} + \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y)\|_2}. \tag{3}$$

3) Gradient-Based Iterative Methods: I-FGSM [28], momentum iterative (MI)-FGSM [29], and projected gradient descent (PGD) [30] iteratively apply the one-step methods multiple times with a small step size α. The iterative attack method can be defined as

$$\boldsymbol{x}_0^* = \boldsymbol{x}, \quad \boldsymbol{x}_{t+1}^* = \boldsymbol{x}_t^* + \alpha \cdot \text{sign}\big(\nabla_{\boldsymbol{x}} L\big(\boldsymbol{x}_t^*, y\big)\big). \tag{4}$$

To make the generated adversarial perturbations imperceptible to humans, i.e., meet the L_p constraint, which can be achieved by simply clipping $\boldsymbol{x}_t^*$ into the ϵ vicinity of $\boldsymbol{x}$ or simply set $\alpha = \epsilon/T$ with T being the number of iterations.

B. Physical Attacks

Physical attacks play a progressively critical role considering their considerable practical values. To make adversarial perturbations effective in real scenarios, bountiful works have been introduced. In [28], the feasibility of physical attacks is verified by the fact that the adversarial examples being captured by the imaging device still have attack efficacy. The expectation over transformation (EOT) [31] algorithm makes adversarial examples robust to dynamic physical conditions.

The adversarial patch [9] is the most frequently used physical attack approach and has been widely applied in many computer vision tasks, such as automatic driving, face recognition, and object detection. We give the reviews in detail as follows:

For automatic driving systems, [32] devises robust physical perturbations to generate physical perturbations that can steadily fool a DNN-based classifier under physical dynamic conditions. The authors in [33] elaborate and camouflage adversarial noises into a natural appearance that looks legitimate to human observers to design adversarial traffic signs. The translucent patch in [34] is the first camera-based physical attack method, in which the patch is placed on a camera lens rendering the automatic driving system fail to detect the traffic sign. Some other works that take the safety of autonomous driving into account can also be found in [35] and [36].

For the face recognition systems, [10] shows how an attacker that is aware or unaware of the internals of a face recognition system can physically achieve impersonation and dodge attacks. Later, some researchers generate eyeglasses [37], makeup [38], lights [39], and hat [40] with adversarial perturbations attached to deceive the face recognition systems. In [12], they propose the meaningful adversarial sticker, by manipulating the fusing operation and parameters of real stickers on the objects instead of designing perturbation patterns like most existing works. Several other relevant studies [41], [42] place patches with an attacking effect onto the face or the wearable accessory.

For the objection detection systems, Song et al. [14] broadened physical attacks to more difficult object detection tasks and introduced the "disappearance attack". In their work [43], the authors delve into physical scenario attacks using printed patches and clothes and quantify their attack effectiveness with different metrics. Adversarial T-shirts [44] adopt the deformable adversarial patch on the T-shirts to attack the person detector. The work [15] applies patch-based adversarial examples to hide a person from being detected. Some other relevant research [45], [46] are also devoted to fooling object detectors.

C. Adversarial Attacks in Aerial Images

Most of the adversarial attack studies focus on Earth-based imagery, such as face recognition, person detection, and autonomous driving. Recently, some latest works [47], [48], [49], [50] devote to crafting imperceptible perturbations to attack aerial image classifiers in the digital domain, while physical attacks on satellite and aerial imagery have not been extensively exploited. Some researchers apply adversarial patches to fool aerial imagery classifiers [51] and detectors [19], [52], [53] in the digital domain without verifying the attack efficacy in the physical world. Du et al. [18] elaborate on an adversarial patch that encloses the target object, which is robust enough against atmospheric conditions and temporal variability.

III. Methodology

In this article, we propose a brand-new method called adaptive-patch-based physical attack (AP-PA), which aims to generate adversarial patches to hide objects from aerial

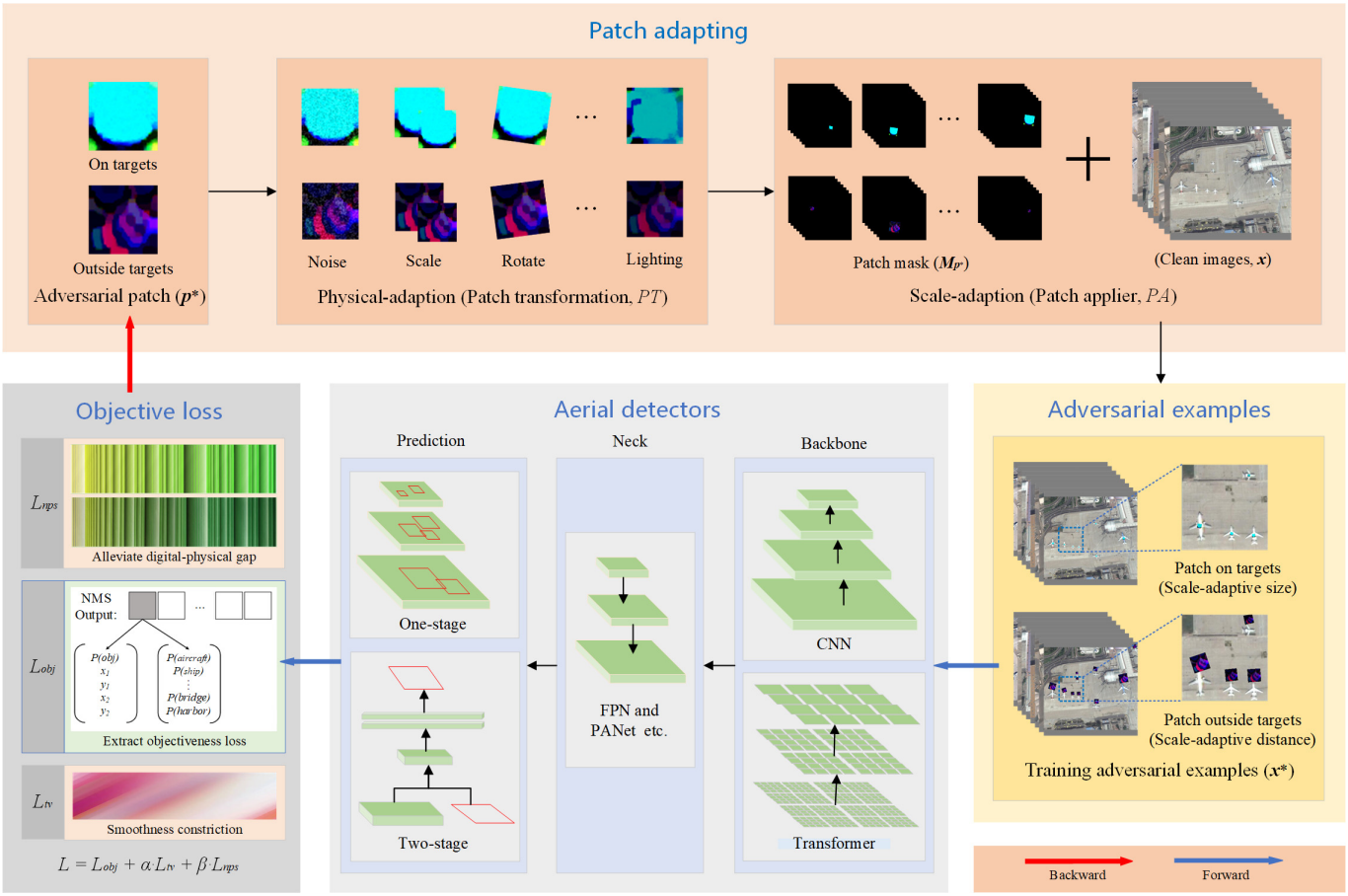


Fig. 2. Illustration of the proposed AP-PA method for physical attack. First, a series of transform operations are conducted to make the adversarial patch accommodate physical dynamic conditions. Second, paste adversarial patches on or outside targets in the proper size and location. Third, the adversarial examples will be fed into a target aerial detector. Next, the objectiveness scores extracted from the detection result are used as part of the total loss. Then, the pixel values of the adversarial patch are optimized to minimize the loss function, including adversarial objectiveness loss (L_{obj}), nonprintability score (L_{nps}), and total variation (L_{tv}). Finally, repeat the above procedures until the end of the training process.

detectors in the physical world. We choose aircraft as the target object, and the different target objects can be simply derived. In this section, we first define the problem to be solved, and then we give an overview of the proposed AP-PA. Finally, we introduce the design of the adaptive patch and objective function in detail, respectively.

A. Problem Formulation

In the aerial detection task, given a benign aerial image $\boldsymbol{x}$, the purpose of the adversarial attack is to make the aerial detection method ignore the target object of the maliciously designed aerial imagery $\boldsymbol{x}^*$. Technically, the adversarial example with elaborated patches can be formulated as

$$\boldsymbol{x}^* = \left(1 - \boldsymbol{M}_{p^*}\right) \odot \boldsymbol{x} + \boldsymbol{M}_{p^*} \odot \boldsymbol{p}^* \tag{5}$$

where $\odot$ and $\boldsymbol{p}^*$ mean the Hadamard product and adversarial patch, respectively. Mask matrix $\boldsymbol{M}_{p^*}$ is used to constrict the size, shape, and location of the adversarial patch, where the value of the patch position area is 1.

The existing studies mainly focus on optimizing adversarial patches with them pasted on the targets. In contrast, our approach also delves into the physical attack with the patch outside the targets. In Sections III-B–III-D, we will illustrate how to obtain an excellent adversarial patch speedily with strong adversarial attack efficacy.

B. Overview of AP-PA

The pipeline of the proposed AP-PA physical attack method is displayed in Fig. 2. Our purpose is to craft an adversarial patch, which is robust enough in real scenarios and with strong attack effectiveness, and the patch can also be applied outside the target objects. To reach this, the weights and bias of the targeted aerial detectors should be fixed during the training process; instead, the pixel values of the adversarial patch should be updated iteratively, which means we are "training" a patch instead of a model with a big set of images containing bountiful of aircraft.

We give a detailed description of the optimizing procedures of the AP-PA approach in the Algorithm 1. Specifically, given an original patch $\boldsymbol{p}^0$, after a series of physical and scale-adaptive transformations, the patch $\boldsymbol{p}^*$ will be placed on the clean image $\boldsymbol{x}$ to form an adversarial image $\boldsymbol{x}^*$. Next, the adversarial example will be fed into a targeted aerial detector. Then, the objectiveness scores extracted from the detection

result can be used as part of the total loss. The next step is backpropagation. The adversarial patch $\boldsymbol{p}^*$ will be updated. Finally, repeat the above steps until the end of the training process.

Algorithm 1 AP-PA

Input: Detector $D(\cdot)$, benign aerial image $\boldsymbol{x}$, ground truth $\boldsymbol{y}$, original patch $\boldsymbol{p}^0$, the adversarial attack loss function $L(\cdot)$, the number of epochs N_{epo}, the number of iterations of each epoch N_{ite}, image size s, hyperparameters α, β, and η

Output: Adversarial patch $\boldsymbol{p}^*$

Initialize $\boldsymbol{p}^0$ randomly in [0, 255], $\boldsymbol{p}^* = \boldsymbol{p}^0$;
for $i = 0$ to N_{epo} **do**
  for $j = 0$ to N_{ite} **do**
    # Patch transformation
    $\boldsymbol{p}^* = PT(\boldsymbol{p}^*)$;
    # Patch applier
    $\boldsymbol{x}^* = PA(\boldsymbol{p}^*, \boldsymbol{l}_{\boldsymbol{p}^*}, w_{\boldsymbol{p}^*}, h_{\boldsymbol{p}^*})$;
    # Detection
    $\boldsymbol{r} = D(\boldsymbol{x}^*)$;
    # Extract objectiveness loss
    $L_{obj} = E(\boldsymbol{r})$;
    # Total loss
    $L = L_{obj} + \alpha \cdot L_{tv} + \beta \cdot L_{nps}$;
    # Update patch
    $\boldsymbol{p}^*_{i,j+1} = \boldsymbol{p}^*_{i,j} + \eta \cdot \nabla_{\boldsymbol{p}^*_{i,j}} L$;
  end for
end for
$\boldsymbol{p}^* = \boldsymbol{p}^*_{N_{epo}, N_{ite}}$;
return $\boldsymbol{p}^*$

C. Patch Adapting

To make the adversarial patch crafted by the AP-PA algorithm successfully fool aerial detection systems in real scenarios, we accommodate the dynamic conditions of the physical world while optimizing the adversarial patches. The real scenarios usually contain varying conditions, including dynamic viewpoint, natural noise, varying lighting, etc. We adopt several physical transformations to simulate such dynamic factors. Technically, we take the transformations of accommodating physical fluctuated conditions into account, such as adding noise, varying scales, random rotation, and lighting shift. The above physical adaptive operations are packed in patch transformation function $PT(\cdot)$. Then the adversarial example can be written as

$$\boldsymbol{x}^* = \left(1 - \boldsymbol{M}_{\boldsymbol{p}^*}\right) \odot \boldsymbol{x} + \boldsymbol{M}_{\boldsymbol{p}^*} \odot PT\left(\boldsymbol{p}^*\right). \tag{6}$$

Next, we focus on how to place the adversarial patches in proper position with the adaptive size due to the varying scales of objects, as shown in Fig. 3.

For the patch on target, our goal is to paste an adversarial patch in the center of the object with a proper size. To achieve that, we use the coordinate (x_1, y_1, x_2, y_2) of the detection result to compute the center coordinate

(a) (b)

Fig. 3. Visual examples of scale-adaptive patch size and distance. (a) Patches on targets. (b) Patches outside targets.

of adversarial patch $\boldsymbol{l}_{\boldsymbol{p}^*}$ as

$$\boldsymbol{l}_{\boldsymbol{p}^*} = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right). \tag{7}$$

Then, considering the different scales of the objects, a scale-adaptive patch method is proposed to tackle this problem. To make the area of the adversarial patch and targeted object keep a proper ratio r_s

$$r_s = \frac{w_{\boldsymbol{p}^*} \cdot h_{\boldsymbol{p}^*}}{w_{\boldsymbol{t}} \cdot h_{\boldsymbol{t}}} \tag{8}$$

where the scale-adaptive patch size can be calculated by

$$w_{\boldsymbol{p}^*} = h_{\boldsymbol{p}^*} = \sqrt[2]{r_s \cdot w_{\boldsymbol{t}} \cdot h_{\boldsymbol{t}}}. \tag{9}$$

Next, the mask of adversarial patch $\boldsymbol{M}_{\boldsymbol{p}^*}$ is formulated as

$$\boldsymbol{M}_{\boldsymbol{p}^*} = \mathrm{PA}\left(\boldsymbol{p}^*, \boldsymbol{l}_{\boldsymbol{p}^*}, w_{\boldsymbol{p}^*}, h_{\boldsymbol{p}^*}\right) \tag{10}$$

where the patch applier function $\mathrm{PA}(\boldsymbol{p}^*, \boldsymbol{l}_{\boldsymbol{p}^*}, w_{\boldsymbol{p}^*}, h_{\boldsymbol{p}^*})$ aims to paste adversarial patch on the corresponding position with an adaptive size. Finally, the original formulated problem (5) can be transformed as

$$\begin{aligned}\boldsymbol{x}^* = &\left(1 - \mathrm{PA}\left(\boldsymbol{p}^*, \boldsymbol{l}_{\boldsymbol{p}^*}, w_{\boldsymbol{p}^*}, h_{\boldsymbol{p}^*}\right)\right) \odot \boldsymbol{x} \\ &+ \mathrm{PA}\left(\boldsymbol{p}^*, \boldsymbol{l}_{\boldsymbol{p}^*}, w_{\boldsymbol{p}^*}, h_{\boldsymbol{p}^*}\right) \odot PT\left(\boldsymbol{p}^*\right).\end{aligned} \tag{11}$$

For patch outside the target, our strategy is to put the adversarial patch on the top of the target in a scale-adaptive distance $d_{\boldsymbol{p}^*}$, i.e., to make $d_{\boldsymbol{p}^*}$ and the height of the target keep an appropriate ratio r_d

$$r_d = \frac{y_2 - y_1}{d_{\boldsymbol{p}^*}} \tag{12}$$

where scale-adaptive distance $d_{\boldsymbol{p}^*}$ can be acquired by

$$d_{\boldsymbol{p}^*} = \frac{y_2 - y_1}{r_d}. \tag{13}$$

Next, the central position of the adversarial patch is given by

$$\boldsymbol{l}_{\boldsymbol{p}^*} = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} - d_{\boldsymbol{p}^*}\right). \tag{14}$$

The rest steps can be derived from the procedures of training the adversarial patch on the target.

D. Objective Function Design

In this article, we aim to design a novel algorithm that can improve the attack efficacy and generate efficiency of the adversarial patch, which can deceive aerial detectors in the physical world. To achieve that, we adopt an optimization process (update patch pixel values) to train an adversarial patch, which can significantly drop the AP of the particular target of aerial detection.

Our objective function contains three parts:

1) Objectiveness Loss L_{obj}: We use the mean of all objectiveness scores of detected objects after nonmaximum suppression operation as adversarial objective loss, which can be written as

$$L_{\text{obj}} = E(\boldsymbol{r}) = \frac{1}{n}\sum_{i=1}^{n} P_i(\text{obj}) \tag{15}$$

where $\boldsymbol{r}$ is the detection results of aerial detectors, and $E(\boldsymbol{r})$ means extracting objectiveness loss L_{obj} from $\boldsymbol{r}$ that contains n detected object(s), including the coordinate (x_1, y_1, x_2, y_2), objective score $P(\text{obj})$, and class scores such as $(P(\text{aircraft}), P(\text{ship}), \ldots, P(\text{bridge}), P(\text{harbor}))$ of each object. The purpose of the adversarial patch is to hide aircraft in the aerial image. To achieve this, we aim to lower the object or class score predicted by the aerial detector. The reason why we do not consider class scores in the loss function is that minimizing the class score of aircraft tends to increase the score of a different class. Moreover, [15] demonstrates that taking the class score into account cannot acquire a stronger attack efficacy.

2) Total Variation Loss L_{tv}: To overcome the problem that the value gap between adjacent pixels is difficult to capture by image acquisition devices, we add total variation as described in [10] into the objective function. L_{tv} tends to guarantee that the optimizer favors the adversarial patch with a smooth pattern and color shift. This loss can be calculated from adversarial patch $\boldsymbol{p}^*$ as follows:

$$L_{tv} = \sum_{i,j} \sqrt{\left(p_{i+1,j} - p_{i,j}\right)^2 + \left(p_{i,j+1} - p_{i,j}\right)^2} \tag{16}$$

where $p_{i,j}$ represents the pixel value of the ith row, jth column of the adversarial patch.

3) Nonprintable Score Loss L_{nps}: Due to the colors' shift of the adversarial patch from the digital domain transform to the physical domain, the nonprintability score in the work [10] is introduced to show how well the colors in the adversarial patch can be printed in the physical world, which represents the distance of adversarial patch between the digital domain and the physical world printed by a normal printer. Here, L_{nps} is formulated as

$$L_{\text{nps}} = \sum_{i,j} \min_{c_{\text{print}} \in C} \left| p_{i,j} - c_{\text{print}} \right| \tag{17}$$

where c_{print} is one color in a group of physical printable colors set C. Taking this loss into account makes the pixel values of our elaborated adversarial patch favor printable colors from printable colors set C.

TABLE I
DETAILED DESCRIPTION OF THE RSOD AND DOTA DATASETS

Datasets	Categories	Images	Instances	Image width	Year
RSOD	4	976	6950	∼1000	2017
DOTA	15	2806	188282	800-4000	2018

Out of the above three components follows the total objective function, written as:

$$L = L_{\text{obj}} + \alpha \cdot L_{tv} + \beta \cdot L_{\text{nps}}. \tag{18}$$

We use hyperparameters α and β to scales L_{tv} and L_{nps}, respectively, and add up the three parts, and then optimize L with Adam [54]. Our proposed AP-PA aims to minimize the objective function L and optimize the adversarial patch, so we freeze all the weights and biases in the aerial detection model and only update the pixel values of the adversarial patch. The initial patch $\boldsymbol{p}^0$ is gifted with random values at the start of the optimizing process.

IV. EXPERIMENTS

In this part, we perform comprehensive experiments to verify the attack efficacy of the proposed AP-PA algorithm. We first describe the experimental settings in detail in Section IV-A. Then we specify the results for the digital attack in Section IV-B and the physical attack in Section IV-C.

A. Experimental Settings

1) Target Models: We choose several representatives, aerial detection models, such as one-stage detectors (you only look once version 2 (YOLOv2) [55], YOLOv3 [56], YOLOv5 [57], and single shot detector (SSD) [58]), two-stage detector (Faster R-CNN [59]), and Transformer-based detector (Swin Transformer [60]) as the attack target models. The open-source codes [YOLOv2,[1] YOLOv3,[2] YOLOv5,[3] multimedia detection (MMD)[4]] are adopted to train the aforementioned aerial detectors.

2) Datasets: We conduct experiments on two public datasets: remote sensing object detection dataset (RSOD)[5] and dataset for object detection in aerial images (DOTA)[6] [61]. The detailed information of the above two datasets is described in Table I. We use the DOTA dataset to train aerial detectors because of its diverse object categories and rich data volumes, and the RSOD dataset is adopted to optimize adversarial patches due to its separate aircraft images.

3) Metrics: Three metrics, recall, precision, and AP, are adopted to evaluate the attack effectiveness of adversarial patches. To verify the attack efficacy toward aerial detectors, it is regarded as a successful attack if the aircraft can be ignored by the aerial detector.

[1]https://github.com/ringringyi/DOTA_YOLOv2
[2]https://github.com/ultralytics/yolov3
[3]https://github.com/ultralytics/yolov5
[4]https://github.com/open-mmlab/mmdetection
[5]https://github.com/RSIA-LIESMARS-WHU/RSOD-Dataset-
[6]https://captain-whu.github.io/DOTA/index.html

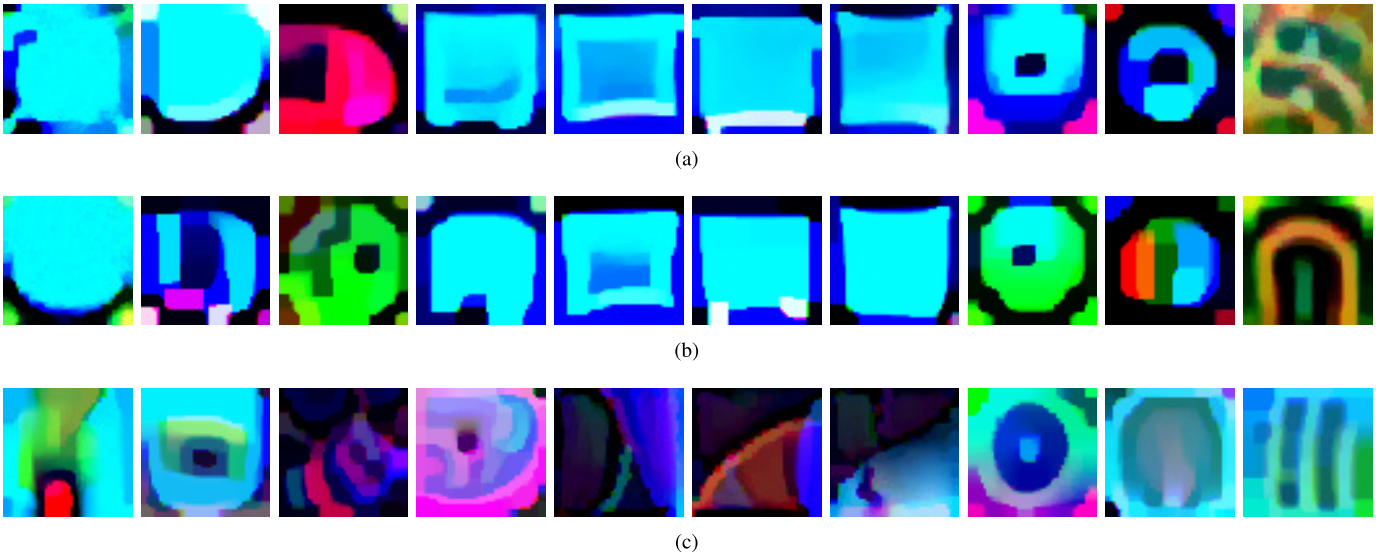


Fig. 4. Elaborated adversarial patches by different methods. From left to right, the target detectors are YOLOv2, YOLOv3, YOLOv5n, YOLOV5s, YOLOv5m, YOLOv5l, YOLOv5x, Faster R-CNN, SSD, and Swin Transformer, respectively. (a) Thys et al. (CVPR) [15]. (b) AP-PA (patch on targets). (c) AP-PA (patch outside targets).

4) Implementation: We refer to the settings in the work [15] and set α and β of (18) to 2.5 and 0.01, respectively, to balance the three parts of the objective loss L. In addition, we empirically set the maximum number of epochs to 600. The iteration T equals the number of training data divided by the batch size, and the thresholds of the intersection of union (IOU) and objectiveness confidence are 0.45 and 0.4, respectively, for all the aerial detectors in both testing and training. In this article, the experiments are conducted with the PyTorch platform [62] using NVIDIA GeForce RTX3080 (10 GB) graphics processing units (GPUs).

B. Experimental Results in Digital Domain

1) Elaborated Adversarial Patches: First, we present the elaborately crafted adversarial patches by our proposed AP-PA method against YOLOv2, YOLOv3, YOLOv5n, YOLOV5s, YOLOv5m, YOLOv5l, YOLOv5x, Faster R-CNN, SSD, and Swin Transformer, respectively, as shown in Fig. 4.

Some interesting properties of adversarial patches can be observed in Fig. 4 as follows.

1) The more similar the aerial detectors, the more similar the corresponding adversarial patches, which have the same pattern style, such as the adversarial patches generated by YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x both by the work [15] and AP-PA (On targets). For AP-PA (Outside targets), the adversarial patches corresponding to YOLOv5n, YOLOv5m, YOLOv5l, and YOLOv5x have the same styles;
2) Different aerial detectors craft adversarial patches with different styles. For example, a one-stage detector [you only look once (YOLO)] and a two-stage detector (Faster R-CNN) generate adversarial patches with totally different styles. Moreover, the CNN-based and Transformer-based detectors also generate adversarial patches with different pattern styles.
3) The position of the adversarial patch has a significant influence on the pattern style of adversarial patches. Specifically, there is a slight difference between the patch crafted by Thys et al. [15] and our AP-PA with the patch on targets, while a huge gap exists between patches on and outside targets.

Based on the above observations, there is one possible explanation. Since the optimization process for adversarial patches is very similar to model training, the difference is that the model parameters are updated in the model training process, while the pixel values of the adversarial patch are updated in the training process of the adversarial patch. Intuitively, models trained with different datasets own different predicted outputs toward the same input, and likewise, adversarial patches trained with different models, such as a one-stage detector (YOLO) and a two-stage detector (Faster R-CNN), have different pattern styles and vice versa, since these patterns are also an abstract representation of the training results.

2) Attack Efficacy: We report the attack effectiveness of our proposed AP-PA method against YOLOv2, YOLOv3, YOLOv5n, YOLOV5s, YOLOv5m, YOLOv5l, YOLOv5x, Faster R-CNN, SSD, and Swin Transformer, respectively. We place the adversarial patch on and outside the target to perform a dodging attack under the aerial detection task and evaluate the recall, precision, and AP, respectively. The quantitative descriptions of attack efficacy in detail are displayed in Tables II and III. Moreover, visual representations of Tables II and III are displayed in Figs. 5 and 6, respectively, to show the patches' attack transferability between different aerial detectors.

It is clear that in all the white-box attack scenarios where the target model is the same as the model for generating adversarial patches (proxy model), our proposed AP-PA method can dramatically drop the detection AP of aerial

TABLE II

EXPERIMENTAL RESULTS OF THE ADVERSARIAL ATTACK WITH THE PATCHES ON TARGETS

Patches \ Detectors	YOLOv2	YOLOv3	YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x	Faster R-CNN	SSD	Swin Transformer
YOLOv2 [55]	**6.33**%	65.80%	80.18%	73.05%	80.77%	78.83%	78.44%	68.88%	47.80%	85.98%
YOLOv3 [56]	19.38%	**59.24**%	75.36%	66.43%	74.20%	72.54%	75.40%	35.35%	28.88%	82.78%
YOLOv5n [57]	63.90%	88.57%	**83.94**%	85.28%	91.60%	89.49%	92.36%	37.16%	39.97%	81.17%
YOLOv5s [57]	9.25%	65.17%	78.28%	**63.60**%	75.13%	74.07%	76.48%	53.72%	38.13%	83.81%
YOLOv5m [57]	12.79%	66.94%	78.47%	67.49%	**73.53**%	75.23%	78.31%	54.49%	42.00%	83.75%
YOLOv5l [57]	11.69%	65.50%	78.31%	67.17%	74.20%	**72.08**%	75.30%	56.37%	41.16%	84.34%
YOLOv5x [57]	8.71%	65.89%	77.64%	65.67%	75.08%	74.53%	**73.90**%	55.30%	40.28%	83.62%
Faster R-CNN [59]	14.27%	76.86%	84.57%	80.58%	85.02%	84.35%	84.84%	**32.90**%	34.29%	80.05%
SSD [58]	27.54%	72.54%	81.50%	77.81%	80.22%	80.60%	81.73%	30.98%	**25.14**%	78.99%
Swin Transformer [60]	66.06%	81.43%	84.70%	83.82%	85.87%	86.31%	85.75%	29.63%	33.90%	**73.61**%
Noise	94.19%	95.09%	94.88%	95.24%	96.37%	96.71%	96.60%	81.42%	75.72%	89.05%

White-box attack results are highlighted in **bold**, and the rest results belong to the black-box attack.
The noise is added to compare the effect of patch occlusion.

TABLE III

EXPERIMENTAL RESULTS OF THE ADVERSARIAL ATTACK WITH PATCHES OUTSIDE TARGETS

Patches \ Detectors	YOLOv2	YOLOv3	YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x	Faster R-CNN	SSD	Swin Transformer
YOLOv2 [55]	**20.72**%	72.18%	50.86%	64.55%	64.87%	65.96%	67.08%	42.23%	48.27%	68.88%
YOLOv3 [56]	54.83%	**64.50**%	57.03%	66.91%	67.38%	72.76%	68.05%	46.40%	49.35%	57.87%
YOLOv5n [57]	58.02%	64.36%	**39.41**%	66.07%	66.60%	69.62%	67.51%	40.87%	46.64%	66.90%
YOLOv5s [57]	55.53%	68.68%	58.07%	**60.68**%	70.35%	73.60%	71.47%	34.58%	47.98%	59.42%
YOLOv5m [57]	55.26%	63.88%	51.83%	53.57%	**55.21**%	62.67%	61.53%	41.23%	41.06%	63.61%
YOLOv5l [57]	56.50%	67.09%	57.96%	57.39%	63.93%	**54.17**%	63.86%	33.96%	42.60%	63.08%
YOLOv5x [57]	52.90%	65.70%	57.87%	61.31%	64.42%	69.03%	**62.65**%	40.45%	48.60%	65.09%
Faster R-CNN [59]	53.15%	72.19%	56.72%	70.27%	72.01%	75.19%	74.19%	**30.27**%	49.42%	57.72%
SSD [58]	44.84%	68.52%	54.12%	66.02%	62.93%	71.29%	67.11%	42.62%	**44.62**%	62.94%
Swin Transformer [60]	47.02%	72.05%	61.75%	71.20%	69.90%	72.57%	70.05%	47.02%	51.84%	**57.91%**

For adversarial patches outside targets, there is no need for considering the impact of occlusion.

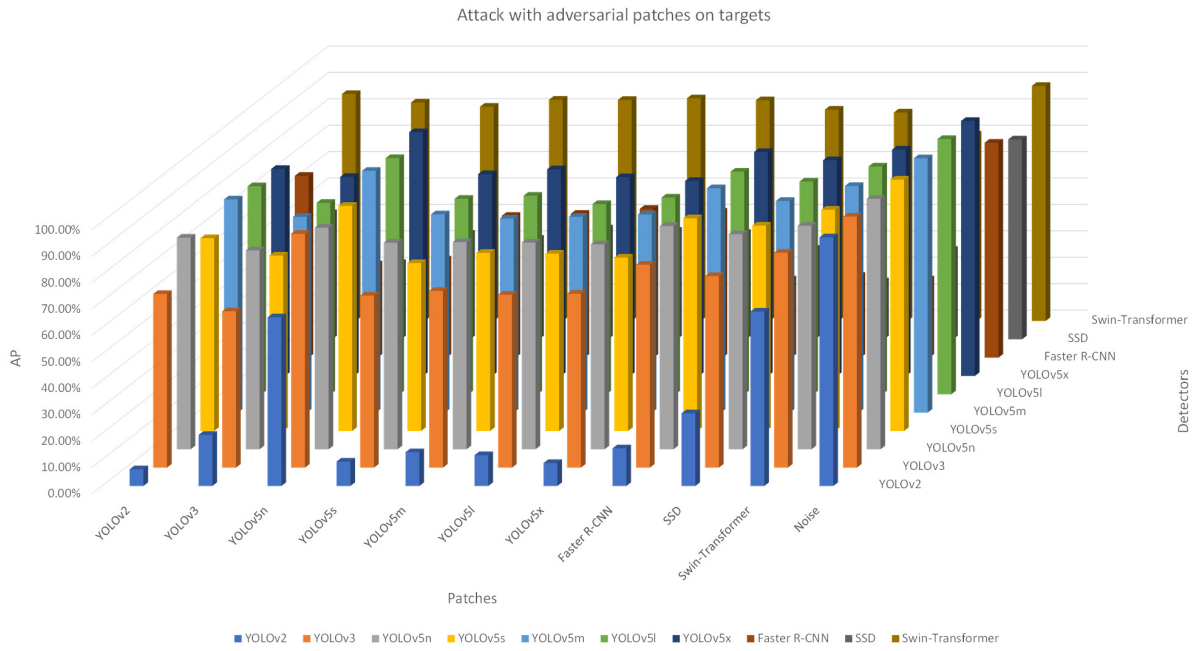


Fig. 5. Transferability of adversarial attack between different aerial detectors with patches on targets.

detectors, especially for some relatively earlier methods such as YOLOv2 (6.33% and 20.72%), Faster R-CNN (32.90% and 30.27%), and SSD (21.14% and 44.62%). For black-box attack scenarios, the target model is different from the proxy model.

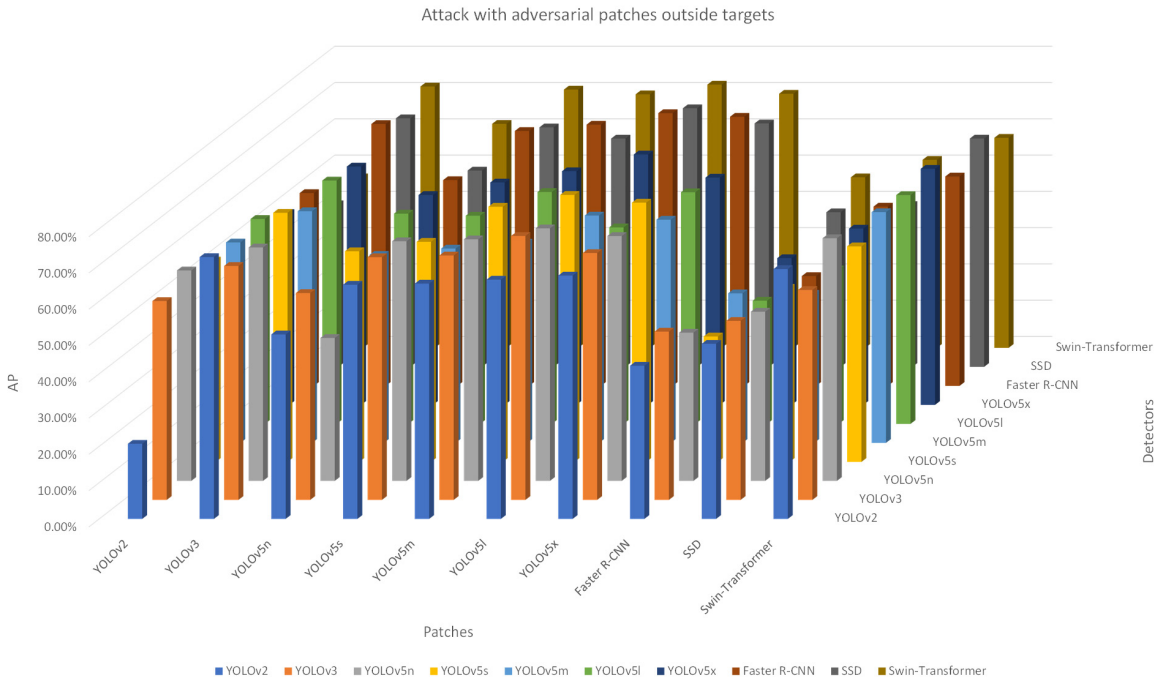


Fig. 6. Transferability of adversarial attack between different aerial detectors with patches outside targets.

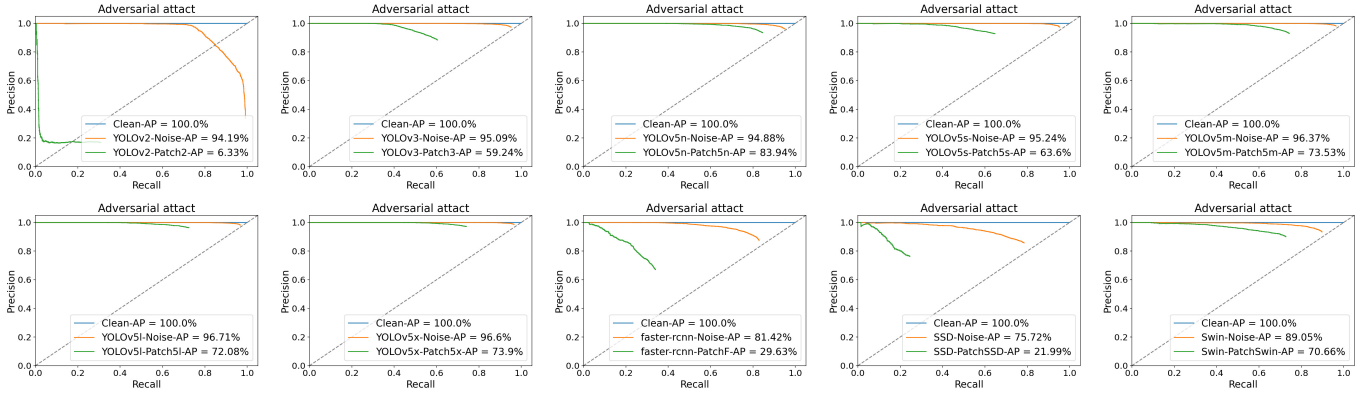


Fig. 7. P–R curves of adversarial attack against different aerial detectors with patches on targets.

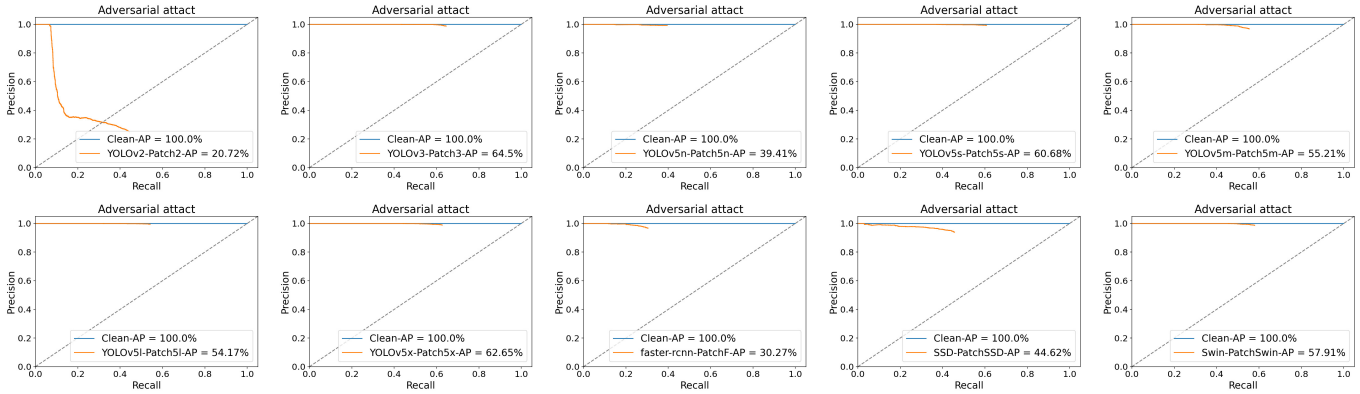


Fig. 8. P–R curves of adversarial attack against different aerial detectors with patches outside targets.

Similarly, the AP-PA can achieve strong attack effectiveness and good transferability, as shown in Figs. 5 and 6. An exciting discovery is that the effect of a transfer-based attack is closely related to the target model's robustness. In other words,

Fig. 9. Visual examples of attack with adversarial patches on targets.

Fig. 10. Visual examples of attack with adversarial patches outside targets.

the more robust the target model, the worse performance a transfer-based attack can achieve and vice versa.

The P–R curve of each aerial detector is displayed in Figs. 7 and 8. From these P–R curves, we can observe the influence of elaborated adversarial patches compared with random noise patches (contrasting the effect of occlusion). A widely adopted way to choose an excellent point on the P–R curve for detection is to draw a diagonal line on the P–R curve. In this work, we use 0.4 as a reference and set the detection results of clean images as 100% AP. The proposed AP-PA can significantly drop the precision and recall of the aerial detectors, no matter the patches on or outside targets. Finally, several visual examples of attack are given in Figs. 9 and 10.

3) Comparisons: We adopt a state-of-the-art physical attack approach from the work [15] IEEE Conference on Computer Vision and Pattern Recognition (CVPR) as the comparison algorithm due to the existing related physical attack methods [18], [19], [52], [53] against aerial detectors all derived from this method. The target detector is still YOLOv2, YOLOv3, YOLOv5n, YOLOV5s, YOLOv5m, YOLOv5l, YOLOv5x, Faster R-CNN, SSD, and Swin Transformer. In addition, we also take patches with different positions into comparison.

In Table IV, we compare the attack efficacy of the proposed AP-PA with another state-of-the-art method from [15] (CVPR). We can see that the proposed AP-PA approach can generate adversarial patches with more robust fooling effectiveness, which can significantly drop the detection AP of aerial detectors in both white-box and black-box settings than the method in [15] in most circumstances. Moreover, we can also see better attack transferability using adversarial patches elaborated by our proposed AP-PA approach.

Three adversarial patches from different algorithms against Faster R-CNN are chosen to compare the optimization process.

TABLE IV

COMPARISON OF EXPERIMENTAL RESULTS IN THE DIGITAL DOMAIN

Patches \ Detectors	—	YOLOv2	YOLOv3	YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x	Faster R-CNN	SSD	Swin Transformer
YOLOv2 [55]	Thys *et al.*	9.64%	69.35%	83.11%	72.54%	82.09%	80.59%	80.08%	71.74%	51.47%	86.72%
	Ours(On)	**6.33%**	**65.80%**	80.18%	73.05%	80.77%	78.83%	78.44%	68.88%	**47.80%**	85.98%
	Ours(Out)	20.72%	72.18%	**50.86%**	**64.55%**	**64.87%**	**65.96%**	**67.08%**	**42.23%**	48.27%	**68.88%**
YOLOv3 [56]	Thys *et al.*	**6.87%**	63.06%	80.29%	71.53%	79.60%	76.82%	78.41%	64.41%	46.10%	86.25%
	Ours(On)	19.38%	**59.24%**	75.36%	**66.43%**	74.20%	**72.54%**	75.40%	**35.35%**	**28.88%**	82.78%
	Ours(Out)	54.83%	64.50%	**57.03%**	66.91%	**67.38%**	72.76%	**68.05%**	46.40%	49.35%	**57.87%**
YOLOv5n [57]	Thys *et al.*	58.17%	70.39%	77.15%	66.75%	80.96%	76.81%	78.96%	**26.64%**	**33.83%**	85.46%
	Ours(On)	63.90%	88.57%	83.94%	85.28%	91.60%	89.49%	92.36%	37.16%	39.97%	81.17%
	Ours(Out)	**58.02%**	**64.36%**	**39.41%**	**66.07%**	**66.60%**	**69.62%**	**67.51%**	40.87%	46.64%	**66.9%**
YOLOv5s [57]	Thys *et al.*	9.98%	66.73%	79.82%	67.12%	76.01%	76.11%	77.54%	60.81%	45.48%	84.29%
	Ours(On)	**9.25%**	**65.17%**	78.28%	63.60%	75.13%	74.07%	76.48%	53.72%	**38.13%**	83.81%
	Ours(Out)	55.53%	68.68%	**58.07%**	**60.68%**	**70.35%**	**73.60%**	**71.47%**	**34.58%**	47.98%	**59.42%**
YOLOv5m [57]	Thys *et al.*	15.34%	69.62%	80.61%	70.82%	75.55%	77.28%	80.20%	61.95%	47.18%	84.81%
	Ours(On)	**12.79%**	66.94%	78.47%	67.49%	73.53%	75.23%	78.31%	54.49%	42.00%	83.75%
	Ours(Out)	55.26%	**63.88%**	**51.83%**	**53.57%**	**55.21%**	**62.67%**	**61.53%**	**41.23%**	**41.06%**	**63.61%**
YOLOv5l [57]	Thys *et al.*	13.03%	68.15%	80.96%	70.14%	76.56%	75.33%	78.69%	64.45%	49.17%	86.22%
	Ours(On)	**11.69%**	**65.50%**	78.31%	67.17%	74.20%	72.08%	75.30%	56.37%	**41.16%**	84.34%
	Ours(Out)	56.50%	67.09%	**57.96%**	**57.39%**	**63.93%**	**54.17%**	**63.86%**	**33.96%**	42.60%	**63.08%**
YOLOv5x [57]	Thys *et al.*	15.24%	69.47%	81.92%	69.74%	77.52%	76.77%	76.88%	64.93%	44.51%	83.84%
	Ours(On)	**8.71%**	65.89%	77.64%	65.67%	75.08%	74.53%	73.90%	55.30%	**40.28%**	83.62%
	Ours(Out)	52.90%	**65.70%**	**57.87%**	**61.31%**	**64.42%**	**69.03%**	**62.65%**	**40.45%**	48.60%	**65.09%**
Faster R-CNN [59]	Thys *et al.*	**12.99%**	**68.13%**	77.56%	73.91%	77.66%	80.20%	81.86%	46.93%	38.73%	84.78%
	Ours(On)	14.27%	76.86%	84.57%	80.58%	85.02%	84.35%	84.84%	32.90%	**34.29%**	80.05%
	Ours(Out)	53.15%	72.19%	**56.72%**	**70.27%**	**72.01%**	**75.19%**	**74.19%**	**30.27%**	49.42%	**57.72%**
SSD [58]	Thys *et al.*	**14.87%**	**66.80%**	77.39%	66.96%	75.32%	74.26%	76.43%	34.59%	**24.81%**	79.32%
	Ours(On)	27.54%	72.54%	81.50%	77.81%	80.22%	80.60%	81.73%	**30.98%**	25.14%	78.99%
	Ours(Out)	44.84%	68.52%	**54.12%**	**66.02%**	**62.93%**	**71.29%**	**67.11%**	42.62%	44.62%	**62.94%**
Swin Transformer [60]	Thys *et al.*	88.23%	92.57%	89.88%	90.55%	93.54%	93.68%	95.22%	56.75%	60.81%	81.97%
	Ours(On)	66.06%	81.43%	84.70%	83.82%	85.87%	86.31%	85.75%	**29.63%**	**33.90%**	73.61%
	Ours(Out)	**47.02%**	**72.05%**	**61.75%**	**71.20%**	**69.90%**	**72.57%**	**70.05%**	47.02%	51.84%	**57.91%**

Strongest attack results are highlighted in **bold**.
On and Out mean patches on and outside targets, respectively.

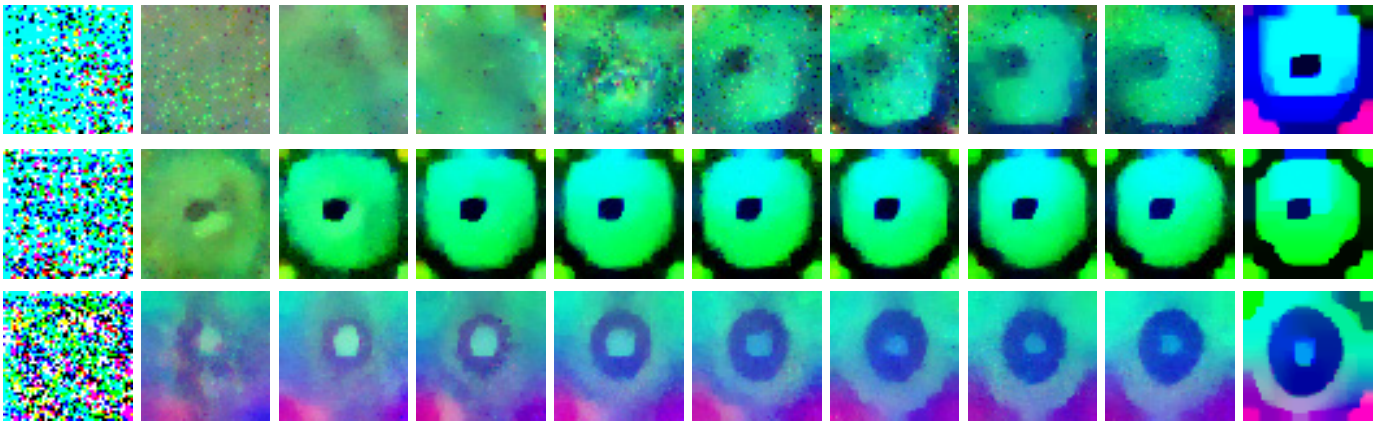


Fig. 11. Visualization of the adversarial patch (against Faster R-CNN) optimization process. From top to bottom are the patches crafted by Thys et al. (CVPR) [15], ours AP-PA (patch on targets), and AP-PA (patch outside targets), respectively. (a) 0. (b) 200. (c) 400. (d) 600. (e) 800. (f) 1000. (g) 1200. (h) 1400. (i) 1600. (j) Final.

The optimizing process of adversarial patches is visualized and displayed in Fig. 11. Several adversarial patches are selected (iterations: 0, 200, 400, 600, 800, 1000, 1200, 1400, 1600) from the optimizing process to analyze the evolutionary progress of adversarial patches. We can observe that our method can enormously accelerate the optimizing efficiency of adversarial patches because the selected adversarial patches have greater similarity with the final optimized adversarial patches with stable patterns.

The main reason for the better attack performance of our AP-PA is that we consider all the detected objects, i.e., using the mean scores of all the detected objects to optimize adversarial patches instead of the only one object with the biggest objectiveness score ([15], [18], [19], [52], [53]), which can not

TABLE V
COMPARISON OF EXPERIMENTAL RESULTS OF THE ADVERSARIAL PATCH WITH DIFFERENT RESOLUTIONS

Patches \ Detectors	—	YOLOv2	YOLOv3	YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x	Faster R-CNN	SSD	Swin Transformer
YOLOv2 [55]	150	18.74%	74.95%	**78.03%**	76.86%	**78.39%**	79.98%	81.55%	**14.64%**	**10.77%**	**78.16%**
	50	**6.33%**	**65.80%**	80.18%	**73.05%**	80.77%	**78.83%**	**78.44%**	68.88%	**47.80%**	85.98%
YOLOv3 [56]	150	**18.22%**	**59.18%**	77.88%	73.41%	77.11%	74.89%	78.10%	38.95%	**24.62%**	83.43%
	50	19.38%	59.24%	**75.36%**	**66.43%**	**74.20%**	**72.54%**	**75.40%**	**35.35%**	28.88%	**82.78%**
YOLOv5n [57]	150	**20.55%**	**74.20%**	**77.52%**	**73.00%**	**78.97%**	**81.15%**	**82.96%**	58.70%	**33.66%**	85.09%
	50	63.90%	88.57%	83.94%	85.28%	91.60%	89.49%	92.36%	**37.16%**	39.97%	**81.17%**
YOLOv5s [57]	150	14.36%	68.57%	80.73%	69.29%	75.96%	76.86%	78.88%	59.25%	**34.47%**	84.99%
	50	**9.25%**	**65.17%**	**78.28%**	**63.60%**	**75.13%**	**74.07%**	**76.48%**	**53.72%**	38.13%	**83.81%**
YOLOv5m [57]	150	14.02%	**66.10%**	78.81%	**67.47%**	**71.41%**	**72.71%**	**77.38%**	**49.37%**	**40.71%**	**83.10%**
	50	**12.79%**	66.94%	**78.47%**	67.49%	73.53%	75.23%	78.31%	54.49%	42.00%	83.75%
YOLOv5l [57]	150	16.67%	66.75%	81.09%	70.42%	74.84%	72.09%	78.31%	59.94%	42.09%	84.42%
	50	**11.69%**	**65.50%**	**78.31%**	**67.17%**	**74.20%**	**72.08%**	**75.30%**	**56.37%**	**41.16%**	**84.34%**
YOLOv5x [57]	150	18.84%	71.36%	84.22%	72.68%	78.54%	77.45%	77.62%	67.36%	44.79%	84.49%
	50	**8.71%**	**65.89%**	**77.64%**	**65.67%**	**75.08%**	**74.53%**	**73.90%**	**55.30%**	**40.28%**	**83.62%**
Faster R-CNN [59]	150	**12.46%**	**73.93%**	**83.44%**	**78.66%**	**83.21%**	**81.66%**	**84.25%**	**26.18%**	**24.18%**	80.51%
	50	14.27%	76.86%	84.57%	80.58%	85.02%	84.35%	84.84%	32.90%	**34.29%**	**80.05%**
SSD [58]	150	**9.78%**	**63.78%**	**75.01%**	**63.65%**	**73.21%**	**72.75%**	**76.93%**	**28.43%**	**13.46%**	80.02%
	50	27.54%	72.54%	81.50%	77.81%	80.22%	80.60%	81.73%	**30.98%**	25.14%	**78.99%**
Swin Transformer [60]	150	92.04%	94.12%	92.25%	93.35%	94.88%	95.65%	96.13%	75.01%	73.44%	86.47%
	50	**66.06%**	**81.43%**	**84.70%**	**83.82%**	**85.87%**	**86.31%**	**85.75%**	**29.63%**	**33.90%**	**73.61%**

Strongest attack results are highlighted in **bold**.

(a) (b)

Fig. 12. Comparison of visual examples before and after adversarial attacks with predicted probabilities. (a) Without patches. (b) With patches.

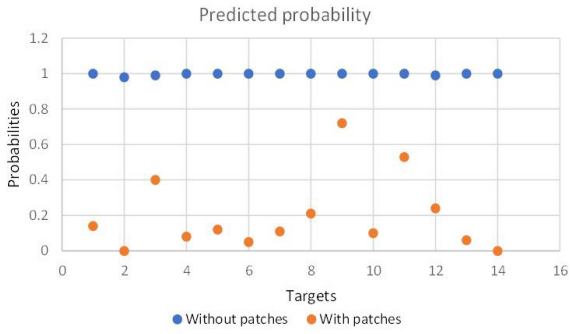


Fig. 13. Predicted probabilities of targets before and after adversarial attacks in the physical condition.

only considerably drop the number of detected objects but also significantly improve the optimizing efficiency of adversarial patches.

Unquestionably, the bigger the patch size, the stronger the attack efficacy. In this work, we also delve into the influences of another attribution of the adversarial patch, namely, the resolution of the adversarial patch. We compare different resolutions of adversarial patches (50 and 150, respectively) to see the impact of patch's resolution. The comparison result is shown in Table V. We can observe that the attack effectiveness of adversarial patches is slightly swayed by its resolution for most cases. In contrast, a more vital adversarial patch is acquired for attacking Swin Transformer by reducing the patch's resolution. During the rest of the experiments, we adopt 50×50 as the patches' resolution because we believe that the simpler the adversarial patches, the less loss of attack efficacy during the physical–digital transformation.

C. Proportionally Scaled Validation Experiments in Physical Domain

In this part, we report the attack effectiveness of adversarial patches elaborated by our AP-PA method in physical scenarios.

First, we place the adversarial patches acquired by a camera on the targets of an aerial image from the public dataset, and Figs. 12 and 13 show the predicted probabilities of several targets corresponding to the ground-truth label before and after conducting attacks in real scenarios. The results demonstrate that the predicted probabilities and IOUs of different targets have dropped significantly, and the maximum reduction is 1.00, which means the targets cannot be detected at all. This illustrates that the adversarial patches generated digitally by the proposed AP-PA method can maintain a stable attack efficacy when applied to realistic physical scenarios.

Second, we use aircraft models to simulate aerial detection in real scenarios with different angles and distances, and the experimental results are shown in Fig. 14. It illustrates that

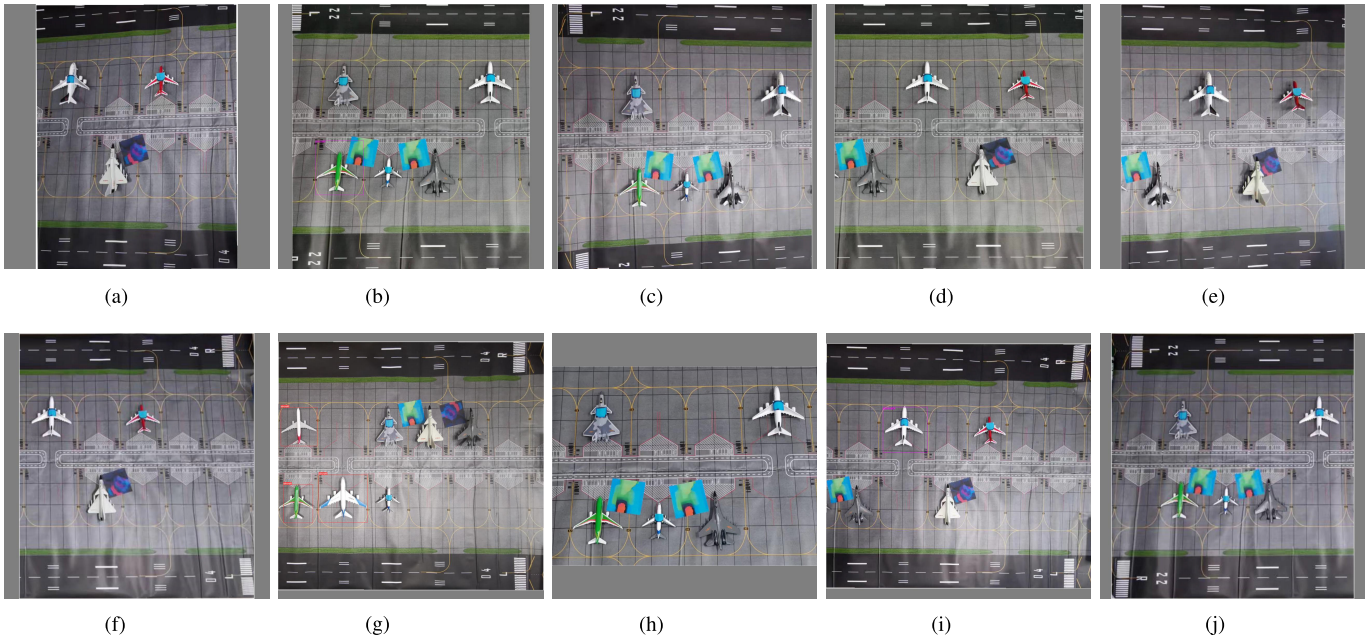


Fig. 14. Visual examples of attack effectiveness in the proportionally scaled scenarios with different physical conditions, including angles and distances, where A and D represent the angle and distance, respectively. (a) A1/D1. (b) A2/D1. (c) A3/D1. (d) A2/D1. (e) A3/D1. (f) A2/D1. (g) A2/D2. (h) A2/D3. (i) A2/D2. (j) A2/D2.

TABLE VI
ATTACK PERFORMANCE ANALYSIS OF DIFFERENT LOSS CONFIGURATIONS OF AP-PA

L_{obj}	L_{tv}	L_{nps}	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	Avg
×	–	–	1.00	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	0.99
✓	×	×	0.99	0.92	0.47	0.99	1.00	0.98	0.98	0.99	1.00	0.99	1.00	0.96	0.99	0.60	0.92
✓	×	✓	0.99	0.63	0.41	1.00	0.83	0.91	0.88	0.96	1.00	0.51	1.00	0.94	0.99	0.62	0.83
✓	✓	×	0.22	0.22	**0.13**	0.19	**0.00**	0.46	0.21	0.96	0.96	**0.10**	0.98	0.36	0.97	**0.00**	0.41
✓	✓	✓	**0.14**	**0.00**	0.40	**0.08**	0.12	**0.05**	**0.11**	**0.21**	**0.72**	**0.10**	**0.53**	**0.24**	**0.06**	**0.00**	**0.20**

The best attack performance are highlighted in **bold**.
T and Avg represent target and average, respectively.

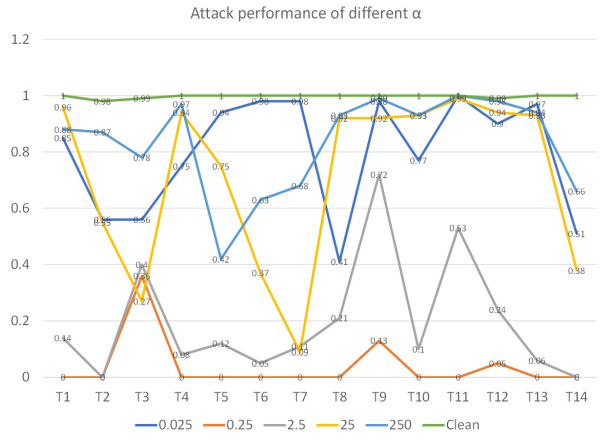
the proposed AP-PA can be robust enough to achieve strong attack effectiveness for most targets with different physical conditions, such as image acquire angles and distances, in real proportional scaled scenarios.

D. Discussions

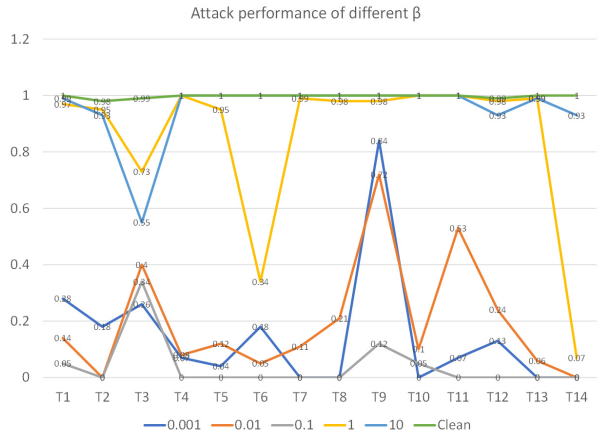
In this part, we perform experiments on ablation studies and hyperparameter settings, respectively.

1) Ablation Studies: The objective function contains three parts, adversarial objectiveness loss (L_{obj}), nonprintability score (L_{nps}), and total variation (L_{tv}). Various forms of ablation studies have been conducted in the physical world to verify the effect of each component in the proposed loss function. Specifically, we first craft the adversarial patches corresponding to different loss configurations. Then, the adversarial patches are printed and captured by the printer and camera, respectively. Finally, the adversarial patches are pasted on the aircraft to verify the physical attack performance. The detailed experimental results are shown in Table VI, in which the prediction result of the clean image is used as the baseline (second row) to compare the prediction confidence drop. It is observed that total variation loss plays a key role in physical attack; nonprintability score can also improve the attack effect to a certain extent, so the combination of both is adopted to get a better attack performance (12 of the 14 targets had the best attack performance and the lowest average prediction confidence) in the physical world. All these results demonstrate the superiority of the proposed AP-PA for the physical attack against aerial detection.

2) Hyperparameter Settings: The hyperparameters α and β are used to balance the digital attack potency and the physical attack potency of the adversarial patches. Suppose the physical adaptation constraints (nonprintability score and total variation) are stronger. In that case, the attack efficacy constraint (adversarial objectiveness loss) will be weakened. The attack effectiveness of the adversarial patch in the real world will be less lost, and vice versa. Therefore, these two hyperparameters can be adjusted flexibly according to the focus of attack requirements. Similarly, we conduct experiments to verify the physical attack performance in various hyperparameter settings. The detailed experimental results are shown in Fig. 15, in which the prediction result of the clean image is also used as the baseline to compare the prediction



(a)



(b)

Fig. 15. Attack performance of adversarial patches with different hyperparameter settings. (a) Comparisons of different α. (b) Comparisons of different β.

confidence drop of different parameter settings. Since the loss mainly comes from objectiveness confidence, nonprintability score and total variation are only a small part. We conduct experiments with different α (ranged in [0.025, 0.25, 2.5, 25, 250]) and β (ranged in [0.001, 0.01, 0.1, 1, 10]), respectively. We can observe that the hyperparameter settings play a key role in physical attack, and we can choose the proper hyperparameter values based on the above experimental results.

V. Conclusion and Future Work

In this article, we proposed the AP-PA adversarial attack algorithm, a physically practicable attack based on negative patches for both the white-box and black-box settings in real physical scenarios. We perform attacks based on adversarial patches in both the digital and physical domains. However, attacking aerial detectors poses a more significant challenge than attacking image classifiers, significantly broadening attacks from the digital environment to real physical scenarios, which requires the adversarial patch to be robust enough to survive real-world distortions due to some uncontrollable physical dynamics, such as different viewing distances, object scales, and lighting conditions. To solve the above problems, we devised the AP-PA method to generate adversarial patches to hide objects from aerial detectors in the physical world. To reach this, aircraft is chosen as the target object for experiments, and different target objects can be derived. During the training process, the weights and biases of the targeted aerial detectors should be fixed. Instead, the pixel values of the adversarial patch should be updated iteratively, which means we are "training" a patch instead of a model. Furthermore, a new loss is devised to consider more available information on detected objects to optimize the adversarial patches, which can significantly improve the patch's attack efficacy and optimization efficiency. In addition, most of the existing adversarial attack methods focus on digital attacks and individual object detectors. So we also establish one of the first comprehensive, coherent, and rigorous benchmarks to evaluate the attack robustness of adversarial patches on aerial detection tasks in both the digital and physical domains. Extensive experiments on aerial detection in both the white-box and black-box settings demonstrated the robust attack efficacy and transferability of our proposed AP-PA.

In contrast to imperceptible perturbations, patch-based attacks are more accessible to fool aerial detectors in real physical scenarios due to the adversarial perturbations concentrated on a small area, which imaging devices can easily capture with less loss and distortion of attack efficacy. This article focuses on improving malicious patches' generating efficiency and attack efficacy and comprehensively evaluating adversarial patches. In future work, we would like to camouflage adversarial patches and improve the attack effectiveness against robust detectors, such as YOLOv5 and Swin Transformer. In addition, other directions where more research should be conducted are to search for the patch's optimal position and shape.

References

[1] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial–spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.

[2] G. Zhang et al., "Spectral variability augmented sparse unmixing of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.

[3] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3D convolutional autoencoder for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6808–6820, Sep. 2019.

[4] G. Cheng et al., "Anchor-free oriented proposal generator for object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.

[5] S. Mei, X. Chen, Y. Zhang, J. Li, and A. Plaza, "Accelerating convolutional neural network-based hyperspectral image classification by step activation quantization," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2021.

[6] S. Mei, X. Li, X. Liu, H. Cai, and Q. Du, "Hyperspectral image classification using attention-based bidirectional long short-term memory network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2022.

[7] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2014.

[8] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[9] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," 2017, *arXiv:1712.09665*.

[10] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1528–1540.

[11] Y. Dong et al., "Efficient decision-based black-box adversarial attacks on face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7714–7722.

[12] X. Wei, Y. Guo, and J. Yu, "Adversarial sticker: A stealthy attack method in the physical world," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 23, 2022, doi: 10.1109/TPAMI.2022.3176760.

[13] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1369–1378.

[14] D. Song et al., "Physical adversarial examples for object detectors," in *Proc. 12th USENIX Workshop Offensive Technol. (WOOT)*, 2018, pp. 1–10.

[15] S. Thys, W. V. Ranst, and T. Goedeme, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 49–55.

[16] Z. Wang, S. Zheng, M. Song, Q. Wang, A. Rahimpour, and H. Qi, "AdvPattern: Physical-world attacks on deep person re-identification via adversarially transformable patterns," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8341–8350.

[17] B. Chen et al., "Adversarial examples generation for deep product quantization networks on image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 5, 2022, doi: 10.1109/TPAMI.2022.3165024.

[18] A. Du et al., "Physical adversarial attacks on an aerial imagery object detector," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1796–1806.

[19] M. Lu, Q. Li, L. Chen, and H. Li, "Scale-adaptive adversarial patch attack for remote sensing image aircraft detection," *Remote Sens.*, vol. 13, no. 20, p. 4078, Oct. 2021.

[20] F. Liu, C. Zhang, and H. Zhang, "Towards transferable unrestricted adversarial examples with minimum changes," 2022, *arXiv:2201.01102*.

[21] Y. Shi, Y. Han, Q. Hu, Y. Yang, and Q. Tian, "Query-efficient black-box adversarial attack with customized iteration and sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 25, 2022, doi: 10.1109/TPAMI.2022.3169802.

[22] C. Ma, L. Chen, and J.-H. Yong, "Simulating unknown target models for query-efficient black-box attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11835–11844.

[23] K. Mahmood, R. Mahmood, and M. van Dijk, "On the robustness of vision transformers to adversarial examples," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7838–7847.

[24] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 125–136.

[25] G. Cheng, X. Sun, K. Li, L. Guo, and J. Han, "Perturbation-seeking generative adversarial networks: A defense framework for remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.

[26] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[27] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[28] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. London, U.K.: Chapman & Hall, 2018, pp. 99–112.

[29] Y. Dong et al., "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.

[30] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[31] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 284–293.

[32] K. Eykholt et al., "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1625–1634.

[33] R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin, and Y. Yang, "Adversarial camouflage: Hiding physical-world attacks with natural styles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1000–1008.

[34] A. Zolfi, M. Kravchik, Y. Elovici, and A. Shabtai, "The translucent patch: A physical and universal attack on object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15232–15241.

[35] K. Eykholt et al., "Note on attacking object detectors with adversarial stickers," 2017, *arXiv:1712.08062*.

[36] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "DARTS: Deceiving autonomous cars with toxic signs," 2018, *arXiv:1802.06430*.

[37] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "A general framework for adversarial examples with objectives," *ACM Trans. Privacy Secur.*, vol. 22, no. 3, pp. 1–30, Jul. 2019.

[38] Z.-A. Zhu, Y.-Z. Lu, and C.-K. Chiang, "Generating adversarial examples by makeup attacks on face recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2516–2520.

[39] D.-L. Nguyen, S. S. Arora, Y. Wu, and H. Yang, "Adversarial light projection attacks on face recognition systems: A feasibility study," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 814–815.

[40] S. Komkov and A. Petiushko, "AdvHat: Real-world adversarial attack on ArcFace face ID system," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 819–826.

[41] E. Kaziakhmedov, K. Kireev, G. Melnikov, M. Pautov, and A. Petiushko, "Real-world attack on MTCNN face detection system," in *Proc. Int. Multi-Conf. Eng., Comput. Inf. Sci. (SIBIRCON)*, Oct. 2019, pp. 0422–0427.

[42] M. Pautov, G. Melnikov, E. Kaziakhmedov, K. Kireev, and A. Petiushko, "On adversarial patches: Real-world attack on ArcFace-100 face recognition system," in *Proc. Int. Multi-Conf. Eng., Comput. Inf. Sci. (SIBIRCON)*, Oct. 2019, pp. 391–396.

[43] Z. Wu, S.-N. Lim, L. S. Davis, and T. Goldstein, "Making an invisibility cloak: Real world adversarial attacks on object detectors," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 1–17.

[44] K. Xu et al., "Adversarial t-shirt! evading person detectors in a physical world," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 665–681.

[45] S. Vellaichamy et al., "DetectorDetective: Investigating the effects of adversarial examples on object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 21484–21491.

[46] Z. Cai et al., "Zero-query transfer attacks on context-aware object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15024–15034.

[47] L. Chen, Z. Xu, Q. Li, J. Peng, S. Wang, and H. Li, "An empirical study of adversarial examples on remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7419–7433, Sep. 2021.

[48] J.-C. Burnel, K. Fatras, R. Flamary, and N. Courty, "Generating natural adversarial remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[49] Y. Xu, B. Du, and L. Zhang, "Assessing the threat of adversarial examples on deep neural networks for remote sensing scene classification: Attacks and defenses," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1604–1617, Feb. 2021.

[50] Y. Xu and P. Ghamisi, "Universal adversarial examples in remote sensing: Methodology and benchmark," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.

[51] W. Czaja, N. Fendley, M. Pekala, C. Ratto, and I.-J. Wang, "Adversarial examples in remote sensing," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 408–411.

[52] R. den Hollander et al., "Adversarial patch camouflage against aerial detection," *Proc. SPIE*, vol. 11543, pp. 77–86, Sep. 2020.

[53] A. Du et al., "Adversarial attacks against a satellite-borne multispectral cloud detector," 2021, *arXiv:2112.01723*.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.

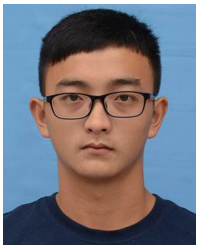
[55] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[56] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[57] J. Glenn et al. *YOLOv5*. Accessed: Jul. 21, 2021. [Online]. Available: https://github.com/ultralytics/yolov5

[58] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[59] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
[60] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted Windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
[61] G.-S. Xia et al., "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.
[62] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.



Jiawei Lian (Graduate Student Member, IEEE) received the B.S. degree in automation from Jiangxi University of Science and Technology, Ganzhou, China, in 2019, and the M.S. degree in control engineering from Northwestern Polytechnical University, Xi'an, China, in 2022, where he is pursuing the Ph.D. degree in information and communication engineering with the School of Electronics and Information.

His research interests include adversarial robustness and computer vision.

Shaohui Mei (Senior Member, IEEE) received the B.S. degree in electronics and information engineering and the Ph.D. degree in signal and information processing from Northwestern Polytechnical University, Xi'an, China, in 2005 and 2011, respectively.

He is a Professor with the School of Electronics and Information, Northwestern Polytechnical University. He was a Visiting Student with The University of Sydney, Sydney, NSW, Australia, from October 2007 to October 2008. His research interests include hyperspectral remote sensing image processing and applications, intelligent signal and information acquisition and processing, video processing, and pattern recognition.

Dr. Mei serves as a Topical Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATION AND REMOTE SENSING (JSTARS), and the Guest Editor for several remote sensing journals. He was a recipient of the Excellent Doctoral Dissertation Award of Shaanxi Province in 2014, the Best Paper Award of the IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) 2017, the Best Reviewer of the IEEE JSTARS in 2019, and the IEEE TGRS in 2022. He also served as the Registration Chair for the IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP) 2014.

Shun Zhang (Member, IEEE) received the B.Eng. degree in electronic engineering from Xi'an Jiaotong University, Xi'an, China, in 2009, and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, in 2016.

He is currently an Associate Professor with the School of Electronic and Information, Northwestern Polytechnical University, Xi'an. His research interests include machine learning, computer vision, and human–computer interaction, focusing on object detection, visual tracking, and person reidentification.

Mingyang Ma (Graduate Student Member, IEEE) received the B.S. degree in communication engineering and the Ph.D. degree in communication and information system from Northwestern Polytechnical University, Xi'an, China, in 2015 and 2021, respectively.

His main research interests include video summarization and image processing.