DeMPAA: Deployable Multi-Mini-Patch Adversarial Attack for Remote Sensing Image Classification

Jun-Jie Huang, *Member, IEEE*, Ziyue Wang, Tianrui Liu, Wenhan Luo, *Senior Member, IEEE*, Zihan Chen, Wentao Zhao, and Meng Wang, *Fellow, IEEE*

***Abstract*— Deep neural networks (DNNs) have demonstrated excellent performance in image classification, yet remain vulnerable to adversarial attacks. Generating deployable adversarial patches (AdvPatchs) represents a promising approach to safeguard critical facilities against DNN-based classifiers used for remote sensing images (RSIs). While existing AdvPatch attack methods are designed for natural images, they typically generate a single and large patch which is impractically oversize for RSI applications. In this article, we propose a deployable multi-mini-patch adversarial attack (DeMPAA) method for RSI classification task, which deploys multiple small AdvPatchs on key locations considering both the feasibility and the effectiveness. The proposed DeMPAA method formulates the problem as a constrained optimization problem that jointly optimizes patch locations and AdvPatchs. The proposed DeMPAA method takes a searching and optimization strategy to tackle it. The DeMPAA framework consists of a feasible and effective map generation (FEMG) module and a patch generation (PG) module. The FEMG module generates a location map to guide the AdvPatch location sampling by excluding the infeasible locations and considering the location effectiveness. In the PG module, a probability-guided random sampling (PRSamp)-based patch location selection method is used to search better locations, and then we optimize the AdvPatchs using gradient descent with respect to an adversarial classification (AdvC) loss and an imperceptibility loss. Extensive experimental results conducted on aerial image dataset (AID) show that the proposed DeMPAA method achieves 94.80% attacking success rate (ASR) against ResNet50 using 16 small patches, which significantly outperforms other AdvPatch methods.**

***Index Terms*— Adversarial patch (AdvPatch) attack, classification, remote sensing image (RSI).**

I. Introduction

DEEP neural networks (DNNs) have achieved superior performance on remote sensing image (RSI) classification [1], [2], [3], and hence enable automatic classification on large-scale RSI. Meanwhile, DNN-based classification also poses a great security concern when the targets in remote sensing scenarios are critical facilities that need to be protected from recognition. Considering that DNNs are vulnerable to adversarial attacks, adversarial attack methods can be of great value in protecting such critical facilities from DNN-based RSI classification. The adversarial attack methods generate adversarial examples by adding adversarial perturbation which is optimized with respect to the adversarial loss to the benign image [4], [5], [6]. The generated adversarial examples can then fool the DNN classifier, leading to an erroneous predicted class label.

Adversarial attack methods have been investigated for RSI classification. Xu et al. [9] systematically analyze the threat of adversarial attack on DNN-based RSI classification and show that adding subtle adversarial perturbation to RSI can lead to misclassification with high confidence. Chen et al. [10] conduct a comprehensive investigation to the effect of adversarial examples on the RSI classification task and reveal the universality and severity of the adversarial example problem. Xu and Ghamisi [7] propose a Mixup-Attack method for black-box adversarial attack to seek the common vulnerabilities of different DNNs which is termed as universal adversarial examples in remote sensing (UAE-RS). The above RSI adversarial attack methods [7], [9], [10] mainly investigate digital adversarial attack approaches to deceive the DNN-based RSI classification models. These methods impose adversarial perturbations on every pixel of an RSI, which is impractical in the real world to deploy the dense and additive adversarial noise everywhere.

To obtain deployable adversarial attack for RS scenes, the number of modifications made to the original RSI should be limited to an acceptable amount. The sparse adversarial attack methods [11], [12], [13], [14], [15] have been proposed to restrict the adversarial perturbation to be sparse, aiming to improve the imperceptibility of the adversarial examples. The sparse adversarial attack approaches seem to be a promising option for deployable adversarial attack for RSI. However, perturbing even 0.1% of pixels on an RSI still requires to change the pixel values on hundreds of locations. Therefore, it is still challenging to physically implement the sparse adversarial attack methods on RSI applications.

The adversarial patch (AdvPatch) attack approach [8] generates adversarial examples by applying an optimized AdvPatch on the image to fool the DNN-based image classifier. In comparison to other aforementioned adversarial attack methods, the AdvPatch attack only requires deploying a single AdvPatch

Manuscript received 5 January 2024; revised 1 April 2024; accepted 20 April 2024. Date of publication 6 May 2024; date of current version 21 May 2024. This work was supported in part by the National Natural Science Foundation of China under Project 62201600, Project 62201604, and Project U1811462; in part by the National University of Defense Technology (NUDT) Innovation Science Foundation under Grant 23-ZZCXKXKY-07; and in part by NUDT Research Project under Grant ZK22-56. *(Jun-Jie Huang and Ziyue Wang contributed equally to this work.) (Corresponding authors: Tianrui Liu; Wentao Zhao.)*

Jun-Jie Huang, Ziyue Wang, Tianrui Liu, Zihan Chen, and Wentao Zhao are with the College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: trliu@nudt.edu.cn; wtzhao@nudt.edu.cn).

Wenhan Luo is with the Division of Emerging Interdisciplinary Areas (EMIA), The Hong Kong University of Science and Technology, Hong Kong, China (e-mail: whluo@ust.hk).

Meng Wang is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230002, China (e-mail: eric.mengwang@gmail.com).

Digital Object Identifier 10.1109/TGRS.2024.3397354

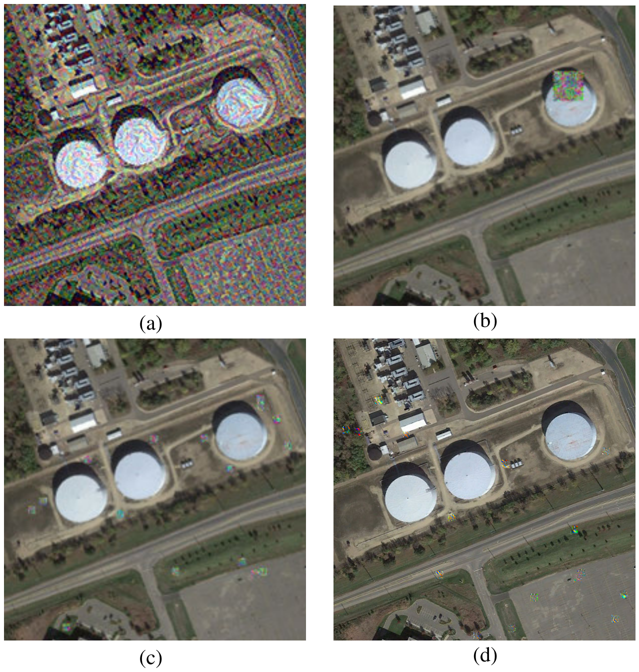


Fig. 1. Visualization of adversarial examples generated by different adversarial attack methods. (a) UAE-RS [7] generates dense and notable adversarial perturbation on every pixel of the input image and (b) AdvPatch [8] pastes a single and large AdvPatch on the input image. The size of this patch is around 1% of the image size and the location of this patch is partially on the oil tank which is difficult to be deployed in practice. (c) Proposed DeMPAA realizes deployable adversarial attack with 16 small AdvPatchs selected according to a feasible and effective map. (d) Our DeMPAA-IP generates more imperceptible AdvPatchs relying on additional imperceptible loss.

instead of perturbing pixel values on hundreds of independent locations. This characteristic increases its practicality for real-world implementation. Since the proposal of AdvPatch by Brown et al. [8], it has been practically applied in many real scenarios, including face recognition [16], [17], [18], autonomous driving [19], [20], pedestrian detection [21], [22], [23], RSI object detection [24], [25], etc.

Despite the success of adversarial attack methods, it is still challenging to achieve deployable adversarial attack on the RSI classification task. The deployability of the model should balance between the size of feasible region and the number of patches. The size of feasible reign for deployment directly relates to the attacking capability, while the number of patches relates to the difficulty for deployment. Fig. 1 shows a visual comparison of different adversarial attack methods on an exemplar RSI. In Fig. 1(a), the UAE-RS method [7] generates universal adversarial examples with dense noise on every image pixel, which is impractical to be deployed in a real scene. In Fig. 1(b), the AdvPatch method [8] generates AdvPatchs that are of 1% size of the image. Such patch size for RSI corresponds to a physical size of over 30 $\times$ 30 m^2, which is too large to be easily deployed in the real scene. Besides, the generated patch is partially located on the oil tank, which is difficult to deploy in practice. There are also adversarial attack methods which generate adversarial examples by perturbing a few number of pixels such as l_0-RS [13]. For the l_0-RS [13] method, 1000 pixels need to be perturbed. Although those 1000 perturbed pixels are imperceptible on the RSI, such a large number of perturbation points can hardly be deployed in practice.

From the above, we can see that the existing adversarial attack methods still face significant challenges when applied to RSI applications. For one thing, the existing AdvPatch methods have primarily been developed for natural images and often require a relatively large patch to ensure a high success rate in attacking the classifier. However, the practical deployment of AdvPatchs in RSI classification requires minimizing their physical size, placing a constraint on the size of the patch. This presents a dilemma in RSI applications where a small patch may not be conducive to achieving a high attacking success rate (ASR), whereas a large patch might be challenging to deploy in real-world settings. Furthermore, the application of the AdvPatch is limited to specific regions within a scene due to practical constraints. For instance, it is difficult to deploy the patch on areas such as trees, cars, or bodies of water. To the best of our knowledge, there is currently a lack of research focus on exploring physically realizable AdvPatch attack methods specifically for RSI classification.

In this article, we propose a simple and effective deployable multi-mini-patch adversarial attack (DeMPAA) method for RSI classification to tackle the aforementioned problems. Our goal is to design a robust and practically deployable adversarial attack method for RSI classification. To achieve this, we propose a novel DeMPAA method which selects a small number of physically deployable key locations on RSI to deploy multiple small AdvPatchs. This ensures that each individual AdvPatch is small enough to be physically deployed in practical locations, while maintaining a high ASR. Specifically, a feasible and effective map generation (FEMG) module is used to determine a patch location map with reference to a feasibility map guided by a feasible region selection network (FRSNet) and an effectiveness map guided by the backpropagated loss. The patch generation (PG) module uses a probability-guided random sampling (PRSamp) method to sample n key patch locations for deploying the AdvPatchs and then optimizes the n AdvPatchs with respect to an adversarial classification (AdvC) loss and an imperceptibility loss. Relying on the cooperation of the FEMG module and the PG module, the proposed DeMPAA method can work with a searching and optimizing strategy to select both practically deployable and effective patch locations for robust and imperceptible adversarial attacks.

The contribution of this work is mainly threefold.

1) We propose a novel DeMPAA method for RSI classification. Instead of using a single large AdvPatch, we propose to search and optimize multiple small AdvPatchs for robust and deployable adversarial attack.
2) In the proposed DeMPAA method, an FEMG module determines a patch location map considering both the feasibility and the effectiveness, and a PG module then samples n key patch locations and optimizes the AdvPatchs with respect to AdvC loss and imperceptible loss.
3) From extensive experimental results, the proposed DeMPAA method achieves a significantly higher ASR when compared with other methods, and an imperceptible version of the proposed method, i.e., DeMPAA-IP,

generates even more visually imperceptible AdvPatchs to be practically feasible for attacking RSI scenes.

The rest of this article is organized as follows: Section II briefly reviews the related work in adversarial attack methods. Section III first formulates the optimization problem, and then introduces the proposed DeMPAA method in detail. Section IV shows comparison results on two commonly used RS datasets and presents ablation studies and discussions to further investigate the properties of the proposed method. Finally, Section V draws conclusions.

II. Related Work

A. Digital Adversarial Attack

Most existing adversarial attacks generate adversarial examples in the digital domain to mislead the DNN classifier [4], [5], [6], [26], [27], [28]. Given a benign image $\boldsymbol{x}$ and the corresponding ground-truth label y, they aim to mislead the classifier $f_{\boldsymbol{\theta}}(\cdot)$ by adding adversarial perturbation on $\boldsymbol{x}$. The adversarial attack methods can be divided into targeted and untargeted attack methods. The targeted adversarial attack method generates adversarial example $\boldsymbol{x}_{\text{adv}}$ with an identified target class label, and the untargeted adversarial attack methods aim to mislead to the classifier to any other wrong class label. The fast gradient sign method (FGSM) [4] generates adversarial examples by adding adversarial perturbation in the direction of the sign of gradient, while the projected gradient descent (PGD) method [26] takes an iterative manner with a smaller step size to achieve a more refined generation of adversarial examples. Deepfool method [5] aims to find the classification boundary hyperplane and move $\boldsymbol{x}$ to the hyperplane to fool the classifier with minimum perturbations. C&W method [27] formulates the adversarial attack as a constrained optimization problem to find minimum perturbations. Luo et al. [28] introduce a constraint on low-frequency subbands between benign and adversarial images, which encourages to generate more imperceptible adversarial examples. Chen et al. [6] propose to generate adversarial examples via invertible neural networks by both adding and dropping semantic information which makes the distortions more imperceptible for human perception. Though taking different strategies, the digital adversarial attack methods require perturbing almost all the pixel values on the image, and therefore can face difficulty in transferring the adversarial perturbations to the physical domain.

B. Sparse Adversarial Attack

To improve the imperceptibility of adversarial attacks, sparse attack methods are introduced to change as few pixels as possible by restricting the perturbation with an l_0-norm [11], [12], [13], [14], [15]. Modas et al. [11] propose a geometry-inspired sparse attack method and approximate the decision boundary as an affine hyperplane to compute the sparse perturbations. Croce and Hein [12] propose to craft adversarial examples by minimizing l_0-norm between the adversarial image and the original image and adding perturbations in region of high variation. Croce et al. [13] propose a random search strategy to optimize the sparse perturbation, termed as l_0-RS. By designing specific sampling distributions, the l_0-RS method only needs to change 0.1%–0.3% pixels to fool the classifiers. He et al. [15] propose to generate transferable sparse adversarial attack by generating the locations and values of the adversarial perturbation with two decoder networks. Zhu et al. [14] propose a homotopy algorithm to generate sparse perturbations and restrict the perturbation bound in one unified framework. Although sparse adversarial attack methods significantly reduce the number of perturbations, it is still challenging to deploy hundreds of perturbations in the physical world.

C. AdvPatch Attack

The AdvPatch attack approach is widely applied for physical attacks. An adversarial example can be obtained by pasting an AdvPatch on the image or on the object. Therefore, it is more feasible to implement in real scene. In the seminal work, Brown et al. [8] propose the AdvPatch method to learn a universal patch which can be pasted on all the images to fool the deep classifiers. AdvPatch uses an AdvPatch of 5% image size which is acceptable for natural images but is too large to attack RSI in real scene. Wei et al. [18] propose to simultaneously optimize the AdvPatch and its position based on reinforcement learning for attacking the face recognition network. Evtimov et al. [19] use the AdvPatch to attack the autonomous driving system by searching a suitable position to paste special stickers on traffic signs. Hu et al. [23] aim to attack pedestrian detectors by optimizing a naturalistic patch within the latent manifold of a pretrained generative adversarial network. Fu et al. [29] prove the effectiveness of AdvPatch attack on vision transformers and select the location with the guidance of the corresponding saliency map termed as patch fool (PFool). Li and Ji [30] propose an end-to-end differentiable AdvPatch attack method termed as generative dynamic patch attack (GDPA) which uses a generator to produce the patch pattern and decides the location with reduced inference time. The existing AdvPatch methods mainly investigate to optimize a single AdvPatch and decide its position to generate adversarial images, since it is in general acceptable to paste a large patch on the natural image.

D. Adversarial Attack on RSI

There are adversarial attack methods designed for RSI applications. For the RSI image classification task, Burnel et al. [31] generate untargeted natural adversarial examples based on Wasserstein generative adversarial networks [32] and achieve high transferability over different DNN classification models. Xu and Ghamisi [7] use a surrogate model to extract the shallow feature of clean images and mix-up images and generate universal adversarial examples for RSI classification by adding perturbations to the benign images. These works are extensions of the digital adversarial attack methods on the RSI classification task; however, these methods are difficult to be impracticable in the physical world. There are recent works that propose to apply AdvPatch attack on the RSI object detection task. Zhang et al. [24] find that the size of objects in

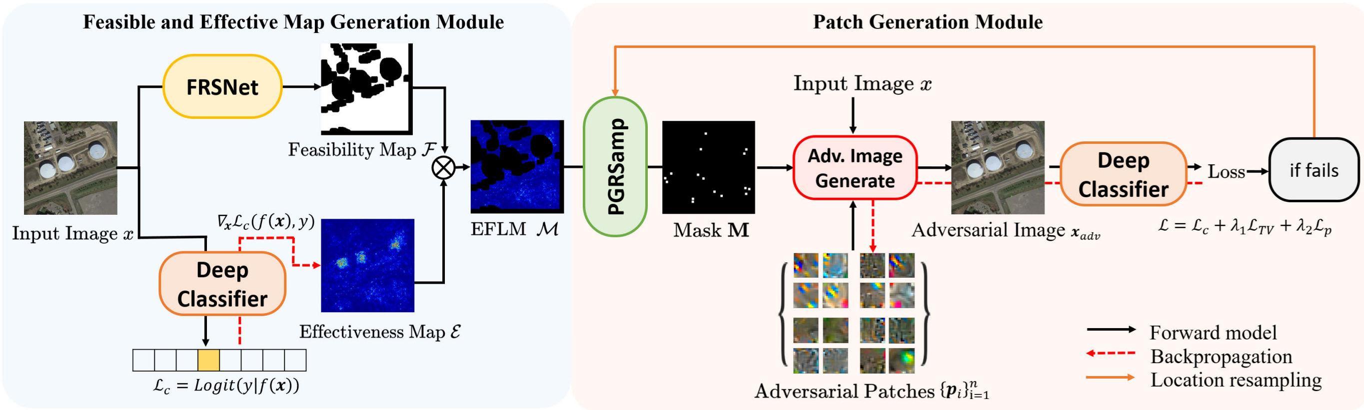


Fig. 2. Overview of the proposed DeMPAA method for RSI classification task. It consists of an FEMG module and a PG module. In the FEMG module, an effectiveness map $\mathcal{E}$ and a feasibility map $\mathcal{F}$ are generated to provide the feasible and effective location map $\mathcal{M}$ to determine the AdvPatch locations considering both the feasibility and the effectiveness. In the PG module, according to $\mathcal{M}$, a PRSamp based patch location selection method is proposed to generate a mask $\mathbf{M}$ with n sampled patch locations. A gradient descent with backpropagation is then used to update the AdvPatchs $\{\mathbf{p}_i\}$ with the given mask. And if attack fails, a new set of patch locations will be resampled with respect to $\mathcal{M}$.

RSI varies, and therefore, they propose to generate a universal AdvPatch that can adapt to multiscale objects. They formulate a joint optimization problem to attack as many objects as possible and use a scale factor to adapt to objects with various sizes. Lian et al. [25] propose adaptive-patch-based physical attack (AP-PA) method for aerial detection to place the AdvPatch both on the object and outside the object. To the best of our knowledge, there are few works that investigated deployable adversarial attack for RSI classification. The major challenging is how to achieve high adversarial ASR while making the adversarial attack feasible for deployment.

III. Proposed Method

In this article, we propose a novel DeMPAA for RSI classification with the objective to generate robust and deployable adversarial examples. The core idea is instead of generating a single large AdvPatch which is difficult to deploy due to its large physical size, we propose to generate multiple small AdvPatchs on key locations.

A. Problem Formulation

Given a benign RSI $\boldsymbol{x}$, the objective of this work is to paste multiple AdvPatchs on $\boldsymbol{x}$ to mislead the deep image classification network of RSI. We leverage a mask image $\mathbf{M}$ and a patch image $\mathbf{P}$ which are of the same size of the benign image $\boldsymbol{x}$ to represent our adversarial image. The region to paste the AdvPatchs is denoted by a binary mask image $\mathbf{M}$ where the patch region is with value 1 and otherwise. The AdvPatchs $\{\mathbf{p}_i\}$ are of size $s \times s$. The adversarial image $\boldsymbol{x}_{\text{adv}}(\mathbf{M}, \mathbf{P})$ can thus be expressed as

$$\boldsymbol{x}_{\text{adv}}(\mathbf{M}, \mathbf{P}) = (\mathbf{1} - \mathbf{M}) \odot \boldsymbol{x} + \mathbf{M} \odot \mathbf{P} \tag{1}$$

where $\mathbf{1}$ is an all-ones matrix, and $\odot$ denotes the Hadamard product, $\mathbf{M}$ and $\mathbf{P}$ are both with the same size of the image $\boldsymbol{x}$.

In this article, we mainly focus on untargeted adversarial attack, i.e., to minimize the probability that the generated adversarial image is classified to the correct class label, that is, to misguide deep image classification networks to predict any of the wrong class labels. The optimization objective for DeMPAA can then be expressed as

$$\arg \min_{\{\mathbf{M} \in \mathcal{F}, \mathbf{P} \in [0,255]\}} \mathcal{L}(f(\boldsymbol{x}_{\text{adv}}(\mathbf{M}, \mathbf{P}), y)) \tag{2}$$

where y is the ground-truth class label, $f(\cdot)$ denotes the target deep classifier, $\mathcal{F}$ represents a feasibility set of patch locations, and $\mathcal{L}$ denotes the loss function. For AdvPatchs to be effective in the physical world, it is crucial that the value of $\mathbf{P}$ should be fall within the range of $[0, 255]$.

From (2), we can see that the objective function involves the optimization of both the patch locations $\mathbf{M}$ and the AdvPatchs $\mathbf{P}$, where the AdvPatchs $\mathbf{P}$ depend on the patch locations $\mathbf{M}$. Such a bilevel optimization problem is generally difficult to optimize. In this article, we propose to solve this problem with a searching and optimization strategy. That is, we first use an FEMG module to generate a feasible and effective location map which is used to guide the selection of the AdvPatch locations, and then use a PG module to search the locations of the AdvPatchs $\mathbf{M}$ and to optimize the AdvPatchs $\mathbf{P}$. The two modules will be introduced in detail in Sections III-C and III-D, respectively.

B. Overview of DeMPAA

Fig. 2 gives an overview of the proposed DeMPAA for RSI classification. It consists of an FEMG module and a PG module. In the FEMG module, an effectiveness map $\mathcal{E}$ and a feasibility map $\mathcal{F}$ are generated to provide the location map $\mathcal{M}$ which is used to determine the AdvPatch locations considering both the feasibility and the effectiveness. Specifically, the effectiveness map $\mathcal{E}$ is obtained by feeding the benign image $\boldsymbol{x}$ into the target classifier using a guided backpropagation, and the feasibility map $\mathcal{F}$ is obtained via an FRSNet. The black area on the feasibility map $\mathcal{F}$ represents the area unsuitable to deploy the AdvPatchs. Then, the feasible and effective location map $\mathcal{M}$ is calculated by elementwise multiplication of $\mathcal{E}$ and $\mathcal{F}$. In the PG module, we calculate the selecting probability P with respect to $\mathcal{M}$ and sample n locations to settle the mask and use a gradient descent with backpropagation to update the AdvPatchs with the given patch locations.

In case that an attack fails, we use a resampling strategy to obtain a new set of patch locations with respect to $\mathcal{M}$.

C. FEMG Module

We propose the FEMG module to guide the optimization of the mask $\mathbf{M}$ in (1). The optimization takes both the feasibility of the AdvPatch locations and the effectiveness of adversarial attack into consideration, relying on a physical feasibility map $\mathcal{F}$ and an attack effectiveness map $\mathcal{E}$, respectively.

1) Feasibility Map: The feasibility map $\mathcal{F}$ is generated by an FRSNet which follows the pipeline of object-contextual representation (OCR) [33]. It helps exclude the locations that are unsuitable to deploy the AdvPatchs in the real-world locations, such as on ships, cars, trees, or bodies of water. The FRSNet is trained on the dense labeling RS dataset (DLRSD) [34] with 17 classes.[1] Among them, six of the classes, i.e., *bare soil*, *dock*, *field*, *grass*, *pavement*, and *sand*, are selected as the feasible locations which can deploy AdvPatchs. The pixel values with feasible class labels are assigned to 1, and 0 otherwise to generate the feasibility map $\mathcal{F}$. A dilation operation according to the patch size $s \times s$ is applied to the semantic map to avoid overlapping with the boundary. The feasibility map $\mathcal{F}$ can be expressed as

$$\mathcal{F} = \mathbf{1}_{s\times s} \circledast \Pi_{\mathbb{B}}(g(\boldsymbol{x})) \tag{3}$$

where $g(\cdot)$ denotes the OCR network, which outputs pixelwise semantic labels, and $\Pi_{\mathbb{B}}(\cdot)$ is a binary projection operator which assigns the feasible locations to 1, and 0 otherwise, $\mathbf{1}_{s\times s}$ represents an all-one convolution kernel with size $s \times s$, and $\circledast$ represents the convolutional operator.

2) Effectiveness Map: We generate an effectiveness map $\mathcal{E}$ to indicate the contribution of the image region to the classification of the image. Specifically, we calculate the gradient of the loss function $\mathcal{L}_c$ for the input image $\boldsymbol{x}$. The magnitude of the gradient on each pixel indicates the potential effectiveness contributing to the generation of the adversarial image. The effectiveness map $\mathcal{E}$ can then be obtained by

$$\mathcal{E} = \mathbf{1}_{s\times s} \circledast |\nabla_{\boldsymbol{x}} \mathcal{L}_c(f(\boldsymbol{x}), y)| \tag{4}$$

where $\mathcal{L}_c$ denotes the AdvC loss function, which can be expressed as $\mathcal{L}_c(f(\boldsymbol{x}), y) = \text{Logit}(y \mid f(\boldsymbol{x}))$ denoting the logit output of $f(\boldsymbol{x})$ with respect to the target class y, and $\mathbf{1}_{s\times s}$ denotes an all-one convolution kernel which is used to calculate the sum of gradient values within the patch.

3) Feasible and Effective Location Map: To obtain a feasible and effective location map, denoted as $\mathcal{M}$, the impact of each pixel is evaluated and unsuitable locations are avoided. This is achieved through an elementwise product operation between the feasibility map $\mathcal{F}$ and the effectiveness map $\mathcal{E}$, i.e., $\mathcal{M} = \mathcal{F} \otimes \mathcal{E}$. The black area in $\mathcal{M}$ is the region where patches cannot be placed, and the brighter the nonblack area, the easier it will affect the classifier. We perform a probabilistic sampling with reference to the magnitude of $\mathcal{M}$, and the specific method will be introduced in Section III-D.

[1] Airplane, bare soil, buildings, cars, chaparral, court, dock, field, grass, mobile home, pavement, sand, sea, ship, tanks, trees, and water.

D. PG Module

Given the feasible and effective location map $\mathcal{M}$, we use a PG module to select the locations and optimize the multiple AdvPatchs.

1) Patch Location Sampling: We propose a PRSamp based patch location selection method to select the multiple small AdvPatch locations. In this work, we generate n small AdvPatchs rather than a single large one. These n AdvPatchs work in a mutual cooperative manner so that the optimization problem here is a combinatorial one and is difficult to be solved. A naive sampling strategy which directly selects the top-n gradient based on the $\mathcal{M}$ map cannot ensure the optimum solution. The proposed PRSamp method treats the values of $\mathcal{M}$ as guidance and selects patch locations based on probabilities. That is, the location with greater value has a higher probability to be selected. The PRSamp method uses softmax with temperature [35] to soft the contribution and calculate the probability as follows:

$$p_{i,j} = \frac{\exp\left(\mathcal{M}_{i,j}/t\right)}{\sum_{u,v} \exp\left(\mathcal{M}_{u,v}/t\right)} \tag{5}$$

where (i, j) denotes the top left location of patch, $\mathcal{M}_{i,j}$ denotes the magnitude at (i, j), $p_{i,j}$ denotes the probability to select (i, j), and t represents the temperature hyperparameter. We discuss the selection of t in detail in Section IV-D3.

2) Patch Optimization: Given the mask $\mathbf{M}$, the AdvPatchs $\mathbf{P}$ are further optimized to fool the deep classifiers. $\mathbf{P}$ can be optimized with respect to (2) with $\mathbf{M}$ being fixed. The values of patches $\mathbf{P}$ are initialized by adding additive white Gaussian noise (AWGN) $\boldsymbol{n} \sim \mathcal{N}(0, \sigma^2)$ with mean zero and variance σ^2 to the RSI patch values. Then gradient descent with backpropagation is used to update $\mathbf{P}$.

3) Resampling Strategy: Since a single random sampling attempt may not ensure successful attack, we propose a patch location resampling strategy. The patch locations will be resampled with respect to (5) if the previously sampled locations do not lead to a successful attack. With this patch location resampling strategy, the proposed DeMPAA method can achieve improved ASR with better patch locations. The adversarial attack will be considered as a successful attack and terminated if the output confidence of the ground-truth label P_y is lower than a threshold $T = 10\%$. Otherwise, a new set of patch locations will be resampled with respect to the location map $\mathcal{M}$.

E. Loss Functions

In this article, we adopt two categories of loss terms to guide the PG. The first one is an AdvC loss, and the second one is an imperceptible loss which includes a total variation (TV) loss and a perceptual color (PerC) loss.

1) AdvC Loss: The AdvC loss is used to guide the generation of adversarial images with high ASR. The logit output refers to the vector before applying the final softmax function when evaluating the CE loss. Following [36], the logit output with respect to the ground-truth label y is used to measure the output score of $f(\cdot)$. The AdvC loss can be expressed as

$$\mathcal{L}_c = \text{Logit}(y \mid f(\boldsymbol{x}_{\text{adv}}(\mathbf{M}, \mathbf{P}))). \tag{6}$$

2) Imperceptible Loss: The visual imperceptibility of the AdvPatchs is essential to protect the privacy of the critical facilities. The TV loss [37] and the PerC loss [38] are set to impose imperceptibility constraint on the generated AdvPatchs.

The TV loss encourages smoothness on the generated AdvPatchs and reduces high-frequency components. It can be expressed as

$$\mathcal{L}_{\text{TV}} = \sum_{i,j} \sqrt{\left(\boldsymbol{x}_{i,j-1} - \boldsymbol{x}_{i,j}\right)^2 + \left(\boldsymbol{x}_{i+1,j} - \boldsymbol{x}_{i,j}\right)^2}. \tag{7}$$

The PerC loss is set to improve the color imperceptibility of the AdvPatchs. Instead of measuring distance in the original RGB color space, PerC measures the color distance in the CIELCH space which is better aligned with human visual perception. Specifically, the PerC loss can be expressed as

$$\mathcal{L}_p = \left(\frac{\Delta L}{S_L}\right)^2 + \left(\frac{\Delta C}{S_C}\right)^2 + \left(\frac{\Delta H}{S_H}\right)^2 + \Delta R \tag{8}$$

where $\Delta L, \Delta C, \Delta H$ are the distance between pixel values of the L (light) channel, C (chroma) channel, and H (hue) channel in CIELCH space, $\Delta R = R_T(\Delta C/S_C)(\Delta H/S_H)$, and S_L, S_C, S_H, R_T are constants.

Therefore, the total loss function can be expressed as

$$\mathcal{L} = \mathcal{L}_c + \lambda_1 \mathcal{L}_{\text{TV}} + \lambda_2 \mathcal{L}_p \tag{9}$$

where λ_1 and λ_2 are regularization parameters to balance the imperceptibility and ASR. In the following of the article, let us denote the proposed DeMPAA method learned with the imperceptible loss as DeMPAA-IP.

The proposed DeMPAA method is summarized in Algorithm 1. Given a benign image $\boldsymbol{x}$ with the ground-truth label y, we first generate a feasible and effective map $\mathcal{M}$, then sample n patch locations guided probability p, and finally optimize the n AdvPatchs with respect to the AdvC loss and the imperceptible loss using gradient descent with backpropagation.

IV. Experiments

In this section, we perform extensive experiments to verify the effectiveness of the proposed DeMPAA method. We first describe the experimental settings, and then compare with the state-of-the-art methods and present ablation studies and discussions.

A. Experimental Settings

1) Dataset: The aerial image dataset (AID) [39] and the RSI classification dataset created by Northwestern Polytechnical University (NWPU-RESISC) [40] are used to evaluate the proposed method. AID has 10 000 images with 30 classes, and all the images are of the resolution of 600 × 600. The NWPU-RESISC dataset has 31 500 images of 45 classes, and the image size is 256 × 256. The dataset has been randomly split into a training set and a testing set with a ratio of 7:3.

Algorithm 1 DeMPAA

Input : benign image $\boldsymbol{x}$, classifier $f(\cdot)$, ground-truth label y, confidence threshold T, number of resampling attempts K, number of patches n, patch size $s \times s$, maximum iterations N;
Output: Adversarial image $\boldsymbol{x}_{adv}$;
Generate Effectiveness Map $\mathcal{E} \leftarrow \mathbf{1}_{s\times s} \circledast |\nabla_{\boldsymbol{x}} \mathcal{L}_c(f(\boldsymbol{x}), y)|$;
Generate Feasibility Map $\mathcal{F} \leftarrow \mathbf{1}_{s\times s} \circledast \Pi_{\mathbb{B}}(g(\boldsymbol{x}))$;
Generate Feasible and Effective Location Map $\mathcal{M} \leftarrow \mathcal{F} \otimes \mathcal{E}$;
Calculate probability $p_{i,j} \leftarrow \frac{\exp(\mathcal{M}_{i,j}/t)}{\sum_{u,v}\exp(\mathcal{M}_{u,v}/t)}$;
for $j = 0 \rightarrow K-1$ **do**
  Randomly initialize n patches $\mathbf{P}$;
  Sample mask $\mathbf{M}$ with respect to probability p;
  for $l = 0 \rightarrow N-1$ **do**
    Update adversarial image $\boldsymbol{x}_{adv}(\mathbf{M},\mathbf{P}) \leftarrow (\mathbf{1}-\mathbf{M}) \odot \boldsymbol{x} + \mathbf{M} \odot \mathbf{P}$;
    Update the loss function: $\mathcal{L}_c \leftarrow \text{Logit}(y \mid f(\boldsymbol{x}_{adv}(\mathbf{M},\mathbf{P})))$;
    Update the output confidence of y: $P_y \leftarrow \text{Softmax}(y \mid f(\boldsymbol{x}_{adv}(\mathbf{M},\mathbf{P})))$;
    if $P_y > T$ **then**
      Update $\mathbf{P}$ by gradient descent with backpropagation;
    else
      Break;
    end
  end
  if $\boldsymbol{x}_{adv}$ *is adversarial* **then**
    Break;
  end
end
return: $\boldsymbol{x}_{adv}$.

2) Classification Model: For AID, the pretrained ResNet50[2] [41] is used as the target classifier, which achieves 3.83% top-1 error by further fine-tuning with the training dataset. We have also fine-tuned ResNet34, ResNet101 [41], and DenseNet121 [42] with 4.90%, 4.43%, and 3.80% top-1 errors, respectively, to evaluate the performance of the proposed DeMPAA method. For NWPU-RESISC, we keep the default settings, and the classifiers ResNet34, ResNet50, ResNet101, and DenseNet121 are fine-tuned with 6.17%, 5.31%, 5.48%, and 4.77% top-1 errors, respectively.

3) Evaluation Metrics: We use the ASR in percentage to evaluate the attacking performance. The learned perceptual image patch similarity (LPIPS) [43] is used to evaluate the perceptual quality of the generated adversarial images. The average processing time is used to evaluate the efficiency of different methods. During testing, the images which cannot be correctly classified are discarded.

4) Settings for DeMPAA: We set the patch number as $n = 16$, and the maximum number of resampling attempts

[2]https://download.pytorch.org/models

TABLE I

ASR, PERCEPTUAL QUALITY, AND AVERAGE PROCESSING TIME OF DIFFERENT ADVPATCH ATTACK METHODS AGAINST FOUR DIFFERENT CLASSIFIERS EVALUATED ON AID AND NWPU-RESISC. (THE BEST AND THE SECOND BEST RESULTS IN EACH COLUMN ARE IN BOLD AND UNDERLINED)

Metrics	Methods	AID					NWPU-RESISC				
		ResNet34	ResNet50	ResNet101	DenseNet121	Average	ResNet34	ResNet50	ResNet101	DenseNet121	Average
ASR (%) ↑	AdvPatch [8]	69.12	71.48	66.32	61.15	67.02	71.25	50.51	73.43	69.24	66.11
	GDPA [30]	66.32	63.10	48.80	40.61	54.71	51.36	34.79	47.62	43.78	44.39
	PFool [29]	65.40	90.20	66.47	70.79	73.22	73.24	54.94	77.78	68.09	68.51
	DeMPAA-IP*	87.46	85.29	83.45	79.61	83.95	90.12	75.95	88.82	85.20	85.02
	DeMPAA*	**93.92**	**96.61**	**93.97**	**89.89**	**93.60**	**94.47**	**79.25**	**92.10**	**90.91**	**89.18**
LPIPS ↓	AdvPatch [8]	0.0597	0.0596	0.0596	0.0602	0.0598	0.1598	0.1673	0.1524	0.1669	0.1616
	GDPA [30]	0.0915	0.0915	0.0916	0.0976	0.0931	0.2419	0.2396	0.2454	0.2437	0.2427
	PFool [29]	0.0531	0.0525	0.0537	0.0594	0.0547	0.1398	0.1457	0.1469	0.1434	0.1440
	DeMPAA-IP*	**0.0425**	**0.0431**	**0.0440**	**0.0420**	**0.0429**	**0.1038**	**0.1048**	**0.1039**	**0.1091**	**0.1054**
	DeMPAA*	0.0537	0.0557	0.0558	0.0569	0.0555	0.1413	0.1558	0.1436	0.1498	0.1476
Time (s) ↓	AdvPatch [8]	12.5	12.1	12.7	13.2	12.6	12.8	13.9	13.1	14.8	13.7
	GDPA [30]	9.6	9.6	**10.1**	**10.3**	9.9	9.7	9.9	10.0	**10.1**	9.9
	PFool [29]	10.7	6.3	10.8	11.8	9.9	7.9	**8.1**	8.3	10.4	8.7
	DeMPAA-IP*	10.1	12.8	18.8	20.4	15.5	10.0	19.9	19.2	21.1	17.6
	DeMPAA*	**3.4**	**5.1**	**10.1**	11.7	**7.6**	**3.3**	9.1	**5.0**	11.9	**7.3**

* w/o FRSNet for fair comparison.

TABLE II

ABLATION STUDY OF THE EFFECT OF DIFFERENT COMPONENTS IN DeMPAA METHOD. (W/O PRSAMP MEANS RANDOMLY SELECTING THE PATCH LOCATIONS WITHOUT ANY GUIDANCE)

Settings			Metrics	Patch Number						
FRSNet	PRSamp	Resampling		1	2	4	8	16	24	32
✗	✓	✓	ASR(%)↑	79.05	87.08	92.23	93.42	96.61	97.15	97.22
			Time(s)↓	11.2	7.2	6.6	5.5	5.1	5.0	5.0
✓	✗	✓	ASR(%)↑	74.49	81.84	87.79	89.74	92.34	93.12	93.87
			Time(s)↓	11.9	10.2	7.6	6.6	6.6	6.5	6.4
✓	✓	✗	ASR(%)↑	69.72	77.62	84.49	89.01	91.37	92.01	92.48
			Time(s)↓	9.3	7.8	4.9	4.8	4.0	3.9	3.8
✓	✓	✓	ASR(%)↑	76.19	84.44	90.21	92.74	94.80	95.12	95.39
			Time(s)↓	10.5	9.4	6.5	6.4	6.1	6.1	6.0

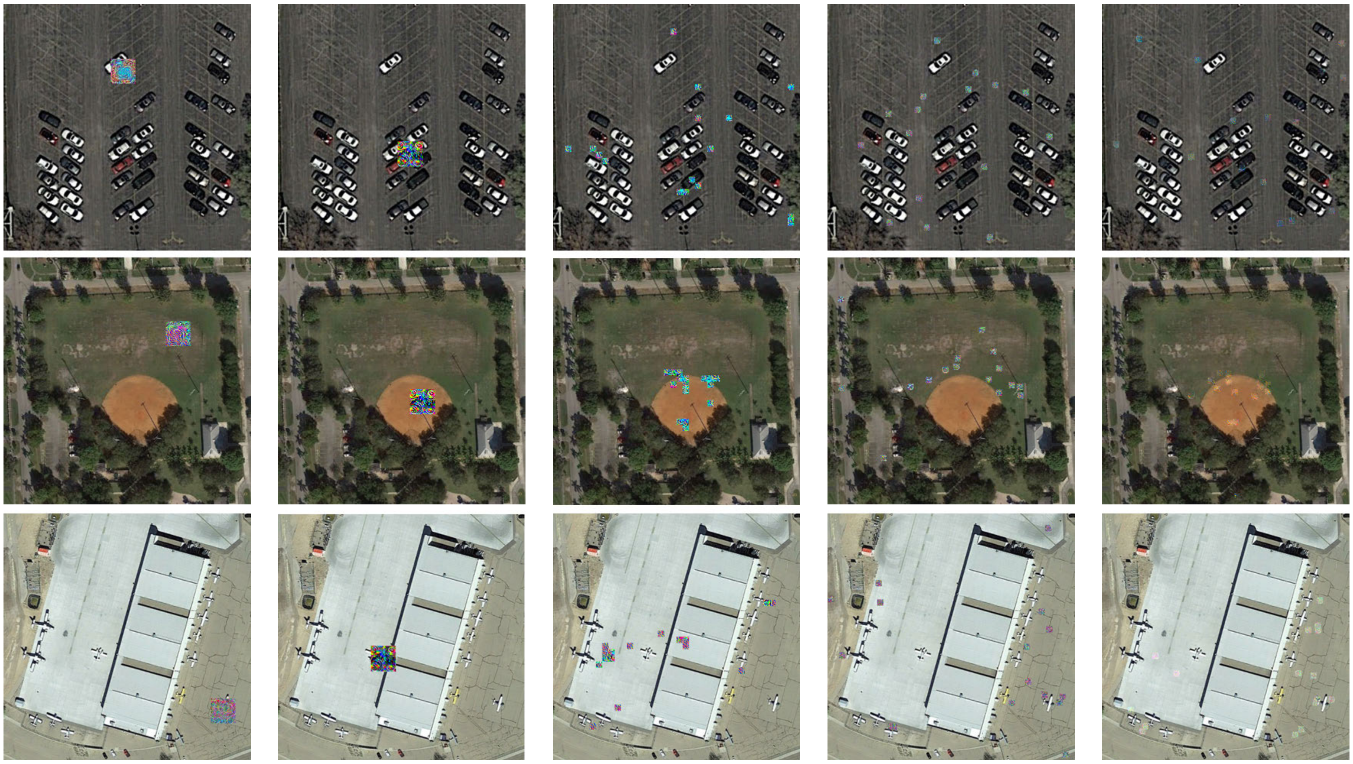
$K = 3$ by default. For DeMPAA, we initialize the AdvPatchs with random noise and set the regularization parameters λ_1, λ_2 to zeros to achieve higher ASR and faster convergence speed. For the imperceptible version DeMPAA-IP, we initialized the AdvPatchs by adding AWGN with standard deviation $\sigma^2 = 75$ to the original image patches and set the regularization parameters to $\lambda_1 = 0.0025$ and $\lambda_2 = 0.005$ to generate more imperceptible adversarial examples. The optimizer used in the PG module is Adam [44] with the initial learning rate 2/255 which is decayed every 200 iterations with decay rate 0.9. The maximum number of iterations is set to $N = 2000$, and the confidence threshold is set to $T = 10\%$ and used to ensure that the generated adversarial image is sufficiently misclassified. All the experiments are performed on a computer with a NVIDIA RTX 3090 GPU of 24-GB memory, and the average memory consumption during code running is 8.0 GB. The code of the proposed method will be publicly available.

B. Comparisons to SOTA Methods

To evaluate the effectiveness of the proposed method, we compare the proposed DeMPAA method with the SOTA AdvPatch attack methods, including AdvPatch [8], GDPA [30], and PFool [29]. For fair comparison, the total patch size is set to 1% of the image size.

Table I illustrates the ASR, LPIPS score, and the average processing time of different adversarial attack methods against commonly used deep classifiers evaluated on AID and NWPU-RESISC. Since the comparison methods can paste the AdvPatch on any position, for fair comparison, here we show the results of the proposed DeMPAA method without using FRSNet to exclude the unsuitable locations. For the results of the complete DeMPAA, refer to Table II.

From Table I, we can observe a similar trend of the results on two datasets. From the average results, the proposed DeMPAA method achieves the highest ASR and fast inference speed against all the classifiers, and DeMPAA-IP achieves the best imperceptibility in terms of LPIPS. The AdvPatch [8] method optimizes a single AdvPatch with a randomly selected location which leads to around 25% lower ASR compared with DeMPAA against all the classifiers evaluated on AID. This result validates the effectiveness of using multiple smaller AdvPatchs. The GDPA [30] method uses a generator to generate the AdvPatch instead of gradient iteration approach and achieves a faster processing speed against ResNet101



(a) (b) (c) (d) (e)

Fig. 3. Three exemplar visualization results of different AdvPatch attack methods evaluated on the AID dataset. The total patch size is set to 1% of image size in all the methods. (a) AdvPatch [8]. (b) GDPA [30]. (c) PFool [29]. (d) DeMPAA (ours). (e) DeMPAA-IP (ours).

and DenseNet121. However, GDPA achieves the lowest ASR among all the methods. The PFool [29] method divides the benign image into a fixed number of blocks by the size of patches, and then uses a saliency map to select the blocks and place the patches which restricts the flexibility of patch locations. Compared with PFool, the proposed DeMPAA method achieves 28.5%, 6.4%, 27.5%, and 19.1% higher ASR on AID against ResNet34, ResNet50, ResNet101, and DenseNet121, respectively. This result also indicates that the proposed location selection method using effectiveness map information as a probabilistic guidance is effective. The DeMPAA-IP also achieves the second best ASR (except against ResNet50) and the best perceptual quality of the generated adversarial examples. Against all the classifiers in AID, DeMPAA-IP achieves around 0.01 lower LPIPS score than PFool.

Fig. 3 shows the visualization results of different AdvPatch attack methods on AID against ResNet50. We can see that the adversarial examples generated by AdvPatch [8] and GDPA [30] are with a large and visible AdvPatch, the PFool [29] method are with a number of visible AdvPatchs on grid, whereas the proposed DeMPAA method generates AdvPatchs which are deployed on feasible locations, and DeMPAA-IP, i.e., the proposed method with imperceptible loss further improves the imperceptibility for human perception.

C. Ablation Studies

To investigate the properties of DeMPAA method, we conduct ablation studies to investigate the effect of different components on the overall performance.

1) Network Structure: Table II shows the ablation study of the proposed DeMPAA method with respect to the proposed FRSNet for feasibility map generation, the proposed PRSamp-based patch location selection method, and the patch location resampling strategy. From Table II, we can see that the FRSNet would slightly reduce ASR (around 2%), but it can help the generated AdvPatchs easier to be deployed in real scene. The proposed location selection method PRSamp can improve ASR and reduce processing time compared with randomly selecting the patch locations without any guidance. And the resampling strategy leads to an improved ASR which also slightly increases the processing time.

It is also interesting to note that by increasing the number of patches, DeMPAA not only achieves a higher ASR but also generates adversarial examples with a reduced processing time. This validates that the proposed DeMPAA leads to both improved effectiveness and efficiency. It can be noted that ASR consistently rises as the number of patches increases, but the improvements tend to diminish when the number of patches exceeds 16. If we keep increasing the number of patches, DeMPAA will be closer to a sparse attack, with a higher ASR, but harder to be deployed for the excessive patch numbers. Considering practical deployment difficulties and ASRs, we set patch number $n = 16$ by default.

2) Imperceptibility: In DeMPAA-IP, the TV loss imposes a smoothness constraint on the generated AdvPatchs, and the PerC loss can further constrain the color of the AdvPatchs to be more similar to the background. The visualizations of examples with and without imperceptible loss are shown in Fig. 4. We can see that with TV loss and PerC loss,

Fig. 4. Visualizations of the generated adversarial examples by (first row) DeMPAA and (second row) DeMPAA-IP with additional imperceptible loss.

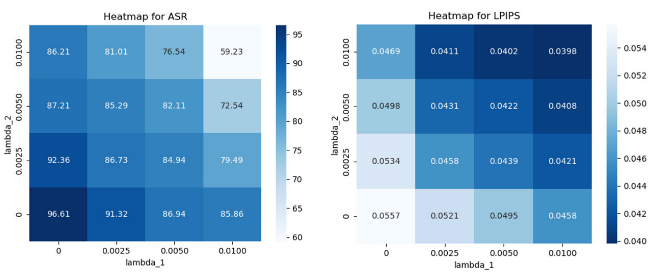


Fig. 5. Ablation study on weight choices for TV loss and PerC loss with respect to (left) ASR and (right) LPIPS.

TABLE III

ASR, Perceptual Quality, and Average Processing Time of DeMPAA-IP With Different Noise Levels for Initialization

σ^2	0	25	50	75	100
ASR(%)↑	69.29	73.34	79.25	83.24	88.34
LPIPS ↓	0.0325	0.0305	0.0386	0.0411	0.0479
Time(s)↓	30.7	27.1	26.7	15.1	11.2

the AdvPatchs will be less perceptible and more similar to the background.

Fig. 5 shows the ablation study on weights of TV loss and PerC loss with respect to the ASR and LPIPS. We can see that both the TV loss and PerC loss contribute to the imperceptibility of the generated AdvPatchs. After comprehensively evaluating different combinations of λ_1 and λ_2, we set their default value to 0.0025 and 0.005, respectively, which achieves a good balance between ASR and imperceptibility.

Table III shows the impact of the strength of the added AWGN on the attacking performance and imperceptibility. We can observe that with an increase in noise level for initializing the AdvPatchs, the generated adversarial examples will be less perceptible, but the attack will be more difficult to converge. To generate less perceptible adversarial examples and keep an acceptable ASR, we set $\sigma^2 = 75$ as the default setting for DeMPAA-IP.

D. Discussions

1) Comparison With Previous Work: In our previous work [45], we propose a class activation map to guide random sampling (APRSamp)-based patch location selection method to select the effective location of patches. It uses Grad-CAM [46] to locate the image region which has the greatest contribution to the classification. Table IV shows comparison of the effectiveness and efficiency of the APRSamp and PRSamp methods. We can observe that PRSamp generates adversarial examples with a faster speed than APRSamp. When taking FRSNet into consideration, APRSamp achieves lower ASR than PRSamp. The reason is that APRSamp prefers to paste AdvPatchs on the locations which attract much more attentions of the classifier. However, these areas are usually infeasible locations to deploy AdvPatchs (e.g., aircraft in *airport*, cars in *parking*) which will be excluded by FRSNet. Fig. 6 shows the generated adversarial examples by the proposed DeMPPA w/o and w/FRSNet. We can see that DeMPPA with FRSNet is able to past adversarial patches on the feasible locations for deployment. Fig. 7 shows the visualization of the gradient map and heat map of benign images with different scene categories; we can observe that the values of heat map are denser than gradient map. Besides, the areas that attract more attention may be excluded by FRSNet with a high probability.

TABLE IV

Adversarial Attack Performance of Different Location Selection Methods When Using Different Numbers of AdvPatchs

Method	Metrics	Patch number				
		1	2	4	8	16
APRSamp	ASR (%)↑	74.36	82.56	88.39	91.13	93.01
	Time (s)↓	11.9	10.1	9.7	8.3	7.4
PRSamp	ASR (%)↑	76.19	84.44	90.21	92.74	94.80
	Time (s)↓	10.5	9.4	6.5	6.4	6.1
APRSamp*	ASR (%)↑	79.42	84.19	90.05	94.18	96.98
	Time (s)↓	13.1	12.4	10.2	8.0	6.4
PRSamp*	ASR (%)↑	79.05	87.08	92.23	93.42	96.61
	Time (s)↓	11.2	7.2	6.6	5.5	5.1

* w/o FRSNet.

TABLE V

Cross-Model Transferability of DeMPAA. The First Column Indicates the Target Classifier for Generating Adversarial Examples. And the Numbers in the Table Indicate the Fooling Rate (%) Against Other Classifiers

Models	ResNet34	ResNet50	ResNet101	DenseNet121
ResNet34	–	8.98	7.57	5.64
ResNet50	10.36	–	5.82	4.10
ResNet101	10.29	7.78	–	4.80
DenseNet121	15.71	10.66	9.13	–

2) Transferability of DeMPAA: We also evaluate the cross-model transferability of the proposed DeMPAA method, and the results are shown in Table V. Specifically, we test the proposed DeMPAA on four models, including ResNet34, ResNet50, ResNet101, and DensNet121. We can observe that the adversarial examples generated by DeMPAA have low transferability cross models, especially when transferring to more complex model (only around 5% fooling rate transferring to DenseNet121 model). The possible reason is that our gradient-based optimization method makes the generated adversarial examples overfit to the target model and common gradient-based attack methods face the same problem.

Fig. 6. Visualization of the generated adversarial examples by DeMPAA (first row) w/o FRSNet and (second row) w/FRSNet. We use patch number $n = 4$ for illustration. Without using FRSNet, most AdvPatchs are pasted on unsuitable areas.

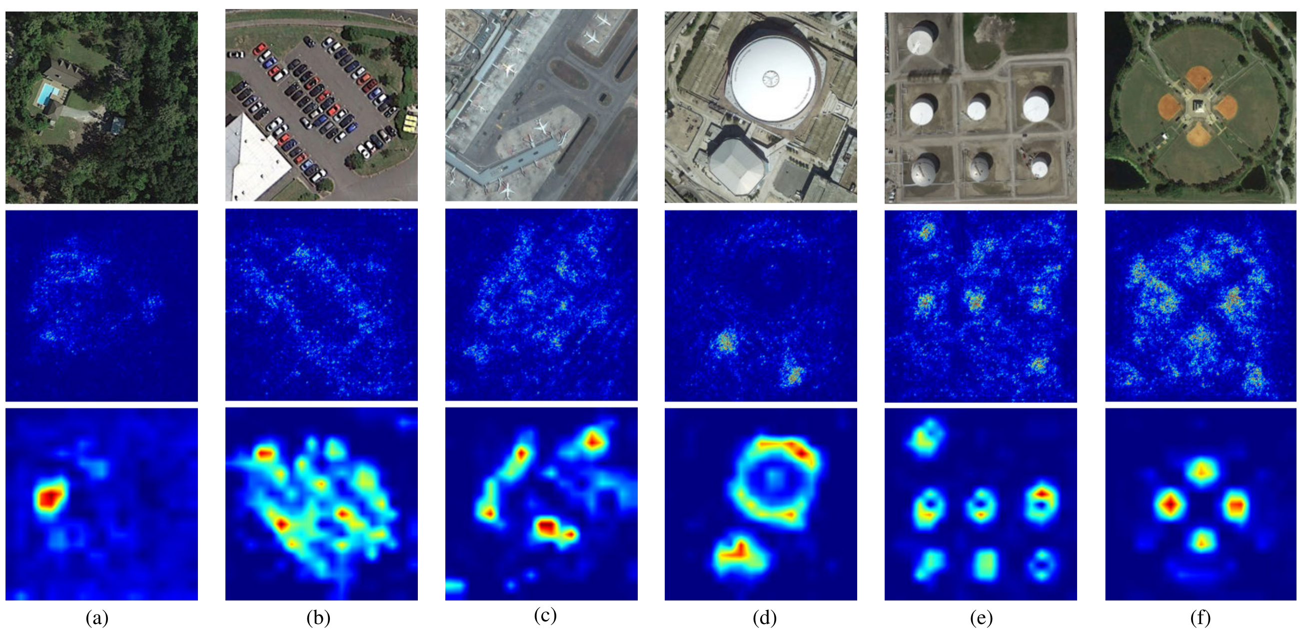


Fig. 7. Gradient of the loss function for (second row) the input image and (third row) the heat map generated by Grad-CAM of benign images with different scene categories. (a) Sparse residential. (b) Parking. (c) Airport. (d) Center. (e) Storage tanks. (f) Baseball field.

To improve the cross-model transferability of the proposed method, the most common method is model ensemble. In detail, the average gradients are used to guide the location selection and PG. And the termination condition is modified to that the output confidences of all the methods are lower than the threshold $T = 10\%$. The results of DeMPAA with model ensemble are shown in Table VI. The values in the diagonal of the table indicate the transferability from the ensemble model to the target black-box model. We can see that the adversarial examples achieve an acceptable transferability (over 40%) on models with similar structures. But when the structure of the target model is different from the ensemble model (from ResNet to DenseNet), there is still much room for improvement.

TABLE VI

CROSS-MODEL TRANSFERABILITY OF DeMPAA WITH MODEL ENSEMBLE METHOD. THE FIRST COLUMN INDICATES THE LEAVE-ONE-OUT ENSEMBLE MODEL AND "-" MODEL DENOTES THE SPECIFIC MODEL IS EXCLUDED FROM THE ENSEMBLE MODEL. THE NUMBERS IN THE TABLE INDICATE THE FOOLING RATE (%) AGAINST THE CORRESPONDING CLASSIFIER

Models	ResNet34	ResNet50	ResNet101	DenseNet121
- ResNet34	54.05	88.89	86.01	75.91
- ResNet50	85.51	44.63	97.10	79.71
- ResNet101	92.48	87.61	40.71	67.70
- DenseNet121	90.61	85.39	89.77	19.87

3) Discuss About Temperature t: In the PRSamp method described in Section III-D, temperature t is an essential parameter to soft the distribution. Table VII shows the results of the PRSamp method with different temperature settings. When $t = 1$, (5) becomes a softmax function. By increasing the temperature parameter, we can observe an improved ASR and a reduced processing time. When t is larger than 10, the ASR begins to decline. Therefore, we choose $t = 10$ as our default setting.

4) Attention Transfer: The Grad-CAM [46] method can be used to visualize the heat map of the class activation of an input image. In this section, we use Grad-CAM to visualize the heat map transformation between the benign images and adversarial examples. The visualization results are

Fig. 8. Visualization of the heat maps of the adversarial examples generated by DeMPAA with different numbers of AdvPatchs n. With the increasing number of patches, the attention transfers from the objects to the AdvPatchs and becomes more scattered. (a) Benign image. (b) $n = 4$. (c) $n = 8$. (d) $n = 16$.

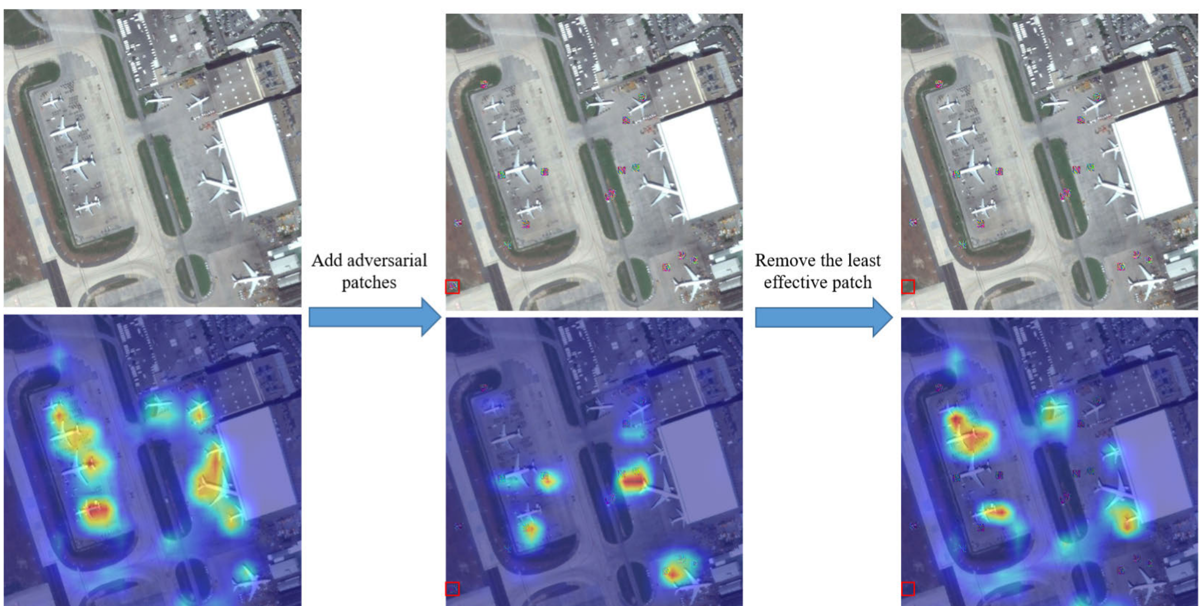


Fig. 9. Analysis on the effect of the AdvPatch with the least attention. The first row represents the benign image, the adversarial image, and the patch removed adversarial image, respectively. The second row denotes the corresponding heat map and the confidence of top-1 classification. (a) Airport (100%). (b) Resort (76%). (c) Airport (68%).

TABLE VII

ASR AND THE AVERAGE PROCESSING TIME OF DIFFERENT t IN (5) TO CALCULATE THE PROBABILITY IN PRSAMP METHOD

Temperature	1	5	10	15	20
ASR (%)↑	90.28	92.48	94.80	94.12	94.04
Time (s)↓	7.4	6.4	6.1	6.2	6.2

shown in Fig. 8. The heat maps of two benign images with the ground-truth label *storage tanks* and *airport* are shown in the first column in Fig. 8. We can see that the classifier mainly focuses on the characteristics related to the class labels, i.e., storage tanks and aircraft. In Fig. 8(b), we show the heat map of adversarial images with $n = 4$ AdvPatchs and can find that most of the attention has now been attracted to the deployed AdvPatchs. In Fig. 8(c) and (d), the number of AdvPatchs increases, and the heat map becomes more scattered. As we conjecture, this would be the reason why the proposed DeMPAA method has a higher ASR when the number of AdvPatchs increases.

In Fig. 8, we find that although most AdvPatchs attract attention from the classifier, there are still certain AdvPatchs attracting less attention. Therefore, we have further investigated whether these AdvPatchs with less attention are necessary or not. We remove the least effective AdvPatch [marked in red box in Fig. 9(b)]. The result shows that after removing the least effective AdvPatch, the classification result changes from *Resort* with 76% confidence to *Airport* with 68% confidence and the heat map shows that the attention transfers back to the aircraft. This indicates that different AdvPatchs are collaborated to form an adversarial example and all contribute to the success of an adversarial attack.

V. Conclusion and Future Works

In this article, we propose a novel DeMPAA method for RSI classification which uses multiple small and less perceptible AdvPatchs to achieve physically feasible adversarial attack. The proposed DeMPAA consists of two main modules. The first module is FEMG module, which uses an FRSNet to generate a map with feasible and effective regions for sampling patch locations while excluding potentially unsuitable regions. The second module is the PG module, which uses a PRSamp-based patch location selection method to select patch locations and performs optimization of the AdvPatchs using gradient descent. An imperceptible version DeMPAA-IP is also proposed to generate less perceptible adversarial examples using the TV and PerC loss. Extensive experimental results on the AID and NWPU-RESISC demonstrate that DeMPAA not only achieves a higher ASR but also accelerates the attacking process.

For future works, a promising direction is to delve into multipatch adversarial attack for black-box scenarios, which aligns closely with the practical applications. This would involve developing attack strategies that remain effective even when there is limited or no information about the target model. Furthermore, it is also essential to improve attacking robustness across a spectrum of environmental variables including fluctuations on lighting conditions, path locations, and diverse viewing perspectives.

References

[1] M. Rezaee, M. Mahdianpari, Y. Zhang, and B. Salehi, "Deep convolutional neural network for complex wetland classification using optical remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 9, pp. 3030–3039, Sep. 2018.

[2] A. Ma, Y. Wan, Y. Zhong, J. Wang, and L. Zhang, "SceneNet: Remote sensing scene classification deep learning network using multi-objective neural evolution architecture search," *ISPRS J. Photogramm. Remote Sens.*, vol. 172, pp. 171–188, Feb. 2021.

[3] X. Hu et al., "SPNet: Spectral patching end-to-end classification network for UAV-borne hyperspectral imagery with high spatial and spectral resolutions," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.

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

[5] 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.

[6] Z. Chen, Z. Wang, J. Huang, W. Zhao, X. Liu, and D. Guan, "Imperceptible adversarial attack via invertible neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 1, pp. 414–424.

[7] Y. Xu and P. Ghamisi, "Universal adversarial examples in remote sensing: Methodology and benchmark," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5619815.

[8] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," in *Proc. NIPS*, 2017, pp. 1–6.

[9] 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.

[10] 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.

[11] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, "SparseFool: A few pixels make a big difference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9087–9096.

[12] F. Croce and M. Hein, "Sparse and imperceivable adversarial attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 4724–4732.

[13] F. Croce, M. Andriushchenko, N. D. Singh, N. Flammarion, and M. Hein, "Sparse-RS: A versatile framework for query-efficient sparse black-box adversarial attacks," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 6, pp. 6437–6445.

[14] M. Zhu, T. Chen, and Z. Wang, "Sparse and imperceptible adversarial attack via a homotopy algorithm," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12868–12877.

[15] Z. He, W. Wang, J. Dong, and T. Tan, "Transferable sparse adversarial attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 14963–14972.

[16] 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.

[17] X. Wei, Y. Guo, and J. Yu, "Adversarial sticker: A stealthy attack method in the physical world," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2711–2725, Mar. 2023.

[18] X. Wei, Y. Guo, J. Yu, and B. Zhang, "Simultaneously optimizing perturbations and positions for black-box adversarial patch attacks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 9041–9054, Jul. 2023.

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

[20] 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.

[21] Z. Hu, S. Huang, X. Zhu, F. Sun, B. Zhang, and X. Hu, "Adversarial texture for fooling person detectors in the physical world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13307–13316.

[22] 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.

[23] Y.-C.-T. Hu, J.-C. Chen, B.-H. Kung, K.-L. Hua, and D. S. Tan, "Naturalistic physical adversarial patch for object detectors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7848–7857.

[24] Y. Zhang et al., "Adversarial patch attack on multi-scale object detection for UAV remote sensing images," *Remote Sens.*, vol. 14, no. 21, p. 5298, Oct. 2022.

[25] J. Lian, S. Mei, S. Zhang, and M. Ma, "Benchmarking adversarial patch against aerial detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.

[26] 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. (ICLR)*, Apr. 2018.

[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] C. Luo, Q. Lin, W. Xie, B. Wu, J. Xie, and L. Shen, "Frequency-driven imperceptible adversarial attack on semantic similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 15315–15324.

[29] Y. F. Fu et al., "Patch-fool: Are vision transformers always robust against adversarial perturbations?" in *Proc. ICLR*, 2021, pp. 1–18.

[30] X. Li and S. Ji, "Generative dynamic patch attack," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2021, pp. 1–21.

[31] 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, 2021.

[32] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[33] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 173–190.

[34] Z. Shao, K. Yang, and W. Zhou, "Performance evaluation of single-label and multi-label remote sensing image retrieval using a dense labeling dataset," *Remote Sens.*, vol. 10, no. 6, p. 964, Jun. 2018.

[35] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Comput. Sci.*, vol. 14, no. 7, pp. 38–39, 2015.

[36] Z. Zhao, Z. Liu, and M. Larson, "On success and simplicity: A second look at transferable targeted attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 6115–6128.

[37] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.

[38] Z. Zhao, Z. Liu, and M. Larson, "Towards large yet imperceptible adversarial image perturbations with perceptual color distance," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1039–1048.

[39] G.-S. Xia et al., "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[40] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

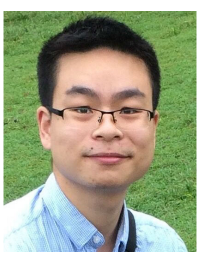
[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[43] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[44] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2014.

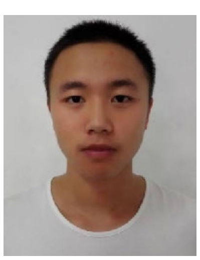
[45] Z. Wang et al., "Multi-patch adversarial attack for remote sensing image classification," in *Proc. 7th APWeb-WAIM Int. Joint Conf. Web Big Data*, 2023.

[46] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.



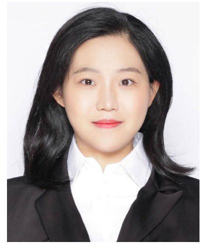
Jun-Jie Huang (Member, IEEE) received the B.Eng. degree (Hons.) in electronic engineering and the M.Phil. degree in electronic and information engineering from The Hong Kong Polytechnic University, Hong Kong, China, in 2013 and 2015, respectively, and the Ph.D. degree from Imperial College London (ICL), London, U.K., in 2019.

From 2019 to 2021, he was a Post-Doctoral Researcher with the Communications and Signal Processing (CSP) Group, Electrical and Electronic Engineering Department, ICL. He is an Associate Professor with the College of Computer Science and Technology, National University of Defense Technology (NUDT), Changsha, China. His research interests include model-based deep learning, computer vision, and signal processing.



Ziyue Wang received the bachelor's and master's degrees from the National University of Defense Technology (NUDT), Changsha, China, in 2017 and 2023, respectively.

His research interests include adversarial attack and remote sensing image.



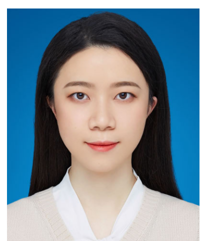
Tianrui Liu received the Ph.D. degree from Imperial College London (ICL), London, U.K., in 2019.

She is currently an Associate Professor with the College of Computer Science and Technology, National University of Defense Technology (NUDT), Changsha, China. Before that, she was a Research Associate with the BioMedIA Group, ICL. Her research interests include computer vision, machine learning, and medical image/video analysis.

Wenhan Luo (Senior Member, IEEE) received the B.E. degree from the Huazhong University of Science and Technology, Wuhan, China, 2009, the M.E. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, 2012, and the Ph.D. degree from Imperial College London, London, U.K., 2016.

He is an Associate Professor with The Hong Kong University of Science and Technology, Hong Kong. Prior to that, he was an Associate Professor with Sun Yat-sen University, Guangzhou, China, and a Research Scientist for Tencent and Amazon. He has published more than 70 papers in top conferences and leading journals, including ICML, CVPR, ICCV, ECCV, ACL, AAAI, ICLR, IEEE Transactions on Pattern Analysis and Machine Intelligence, *International Journal of Computer Vision*, and IEEE Transactions on Image Processing. His research interests include computer vision and machine learning, such as image/video synthesis, and image/video quality restoration.

Dr. Luo has been a reviewer, a senior PC member, and the guest editor of several prestigious journals and conferences.



Zihan Chen received the bachelor's degree from Lanzhou University, Lanzhou, China, in 2021, and the master's degree from the National University of Defense Technology (NUDT), Changsha, China, in 2023, where she is currently pursuing the Ph.D. degree.

Her research interests include computer vision and information hiding.

Wentao Zhao received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2009.

He is currently a Professor with NUDT. Since 2011, he has been a member of Council Committee of Postgraduate Entrance Examination of Computer Science and Technology, NUDT. He has edited one book entitled "*Database Principle and Technology*" and several technical articles, such as communications of the CCF, AAAI, IJCAI, and FAW. His research interests include network performance optimization, information processing, and machine learning.

Meng Wang (Fellow, IEEE) received the B.E. and Ph.D. degrees from the Special Class for the Gifted Young, Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China, in 2003 and 2008, respectively.

He is currently a Professor with Hefei University of Technology, Hefei, China. He has authored more than 200 book chapters, journals, and conference papers in his research areas. His research interests include multimedia content analysis, computer vision, and pattern recognition.

Dr. Wang was a recipient of the ACM SIGMM Rising Star Award 2014. He is an Associate Editor of IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Transactions on Neural Networks and Learning Systems.