Generating Adversarial Examples Against Remote Sensing Scene Classification via Feature Approximation

Rui Zhu, Shiping Ma, Jiawei Lian, *Graduate Student Member, IEEE*, Linyuan He, and Shaohui Mei, *Senior Member, IEEE*

***Abstract*—The existence of adversarial examples highlights the vulnerability of deep neural networks, which can change the recognition results by adding well-designed perturbations to the original image. It brings a great challenge to the remote sensing images (RSI) scene classification. RSI scene classification primarily relies on the spatial and texture feature information of images, making attacks in the feature domain more effective. In this study, we introduce the feature approximation (FA) strategy, which generates adversarial examples by approximating clean image features to virtual images that are designed to not belong to any category. Our research aims to attack image classification models that are trained with RSI and discover the common vulnerabilities of these models. Specifically, we benchmark the FA attack using both featureless images and images generated via data augmentation methods. We then extend the FA attack to multimodel FA (MFA), improving the transferability of the attack. Finally, we show that the FA strategy is also effective for targeted attacks by approximating the input clean image features to the target category image features. Extensive experiments on the remote sensing classification datasets UC Merced and AID demonstrate the effectiveness of the methods in this article. The FA attack exhibits remarkable attack performance. Furthermore, the proposed MFA attack outperforms the success rate achieved by existing advanced targetless black-box attacks by an average of over 15%. The FA attack also performs better compared to multiple existing targeted white-box attacks.**

***Index Terms*—Adversarial examples, feature approximation (FA), remote sensing, scene classification.**

I. Introduction

REMOTE sensing technologies have undergone significant advancements in recent years, resulting in diverse acquisition methods for remote sensing images (RSI) and increased availability of such imagery. This has greatly contributed to significant advancements in remote sensing studies [1], [2], [3], [4], such as object detection [5], [6], [7], scene classification [8], [9], [10], and object tracking [5], [11], [12], [13]. The application of deep neural networks (DNNs) has demonstrated greater potential compared to traditional image processing methods, and therefore, the implementation of these tasks largely relies on DNNs.

However, in recent studies, it has been found that DNNs are susceptible to some intentional or unintentional perturbations. Adversarial examples can be created by adding subtle perturbations to the original image [14], [15], and DNNs have little resistance to attacks from adversarial examples. This phenomenon questions the security of neural networks. Since DNNs has been widely used in both Earth sciences and remote sensing [16], [17], [18], it is essential to study adversarial examples in remote sensing. By studying digital attacks, we can find the vulnerabilities of DNNs, and research based on this can improve the robustness of DNNs [19]. It can also provide research benchmarks for researchers afterward and indirectly enrich the amount of data in RSI.

Czaja et al. [20] conducted pioneering research and discovered the presence of adversarial examples in classification tasks of satellite RSI. They demonstrated that by introducing adversarial perturbations in a small part of the RSI, it is possible to deceive DNNs. Xu and Ghamisi [21] first investigated universal adversarial examples in RSI and formed an adversarial example dataset called UAE-RS. Zhang et al. [22] proposed a generalized adversarial patch generation method for multiscale objects in implemented the attack in the physical world.

Currently, white-box attacks are the primary method used in research on remote sensing and Earth sciences [21], [23], [24]. These approaches presuppose that the parameters of the target model are known, and the researchers can carry out an attack with these parameters. However, these methods are highly idealized. In practice, the relevant information about the target model are unknown or we only have access to a small portion of the information. Under these circumstances, studying black-box attacks or gray-box attacks is more reliable. A black-box attack refers to an attack where the attacker possesses no prior information about the target model [25]. On the other hand, there is also an attack that is somewhere between the black-box and the white-box, we call it gray-box attack [26]. In the gray-box attack, the attacker has a certain degree of knowledge about the target system, and may have some, but not all, knowledge about the system architecture, some of the code, or configuration information. As mentioned earlier, traditional adversarial

Manuscript received 14 March 2024; revised 22 April 2024; accepted 7 May 2024. Date of publication 13 May 2024; date of current version 30 May 2024. *(Corresponding author: Linyuan He.)*

Rui Zhu, Shiping Ma, and Linyuan He are with the College of Aeronautical Engineering, Air Force Engineering University, Xi'an 710038, China (e-mail: zr15735946178@163.com; mashiping@126.com; hal1983@163.com).

Jiawei Lian and Shaohui Mei 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).

Digital Object Identifier 10.1109/JSTARS.2024.3399780

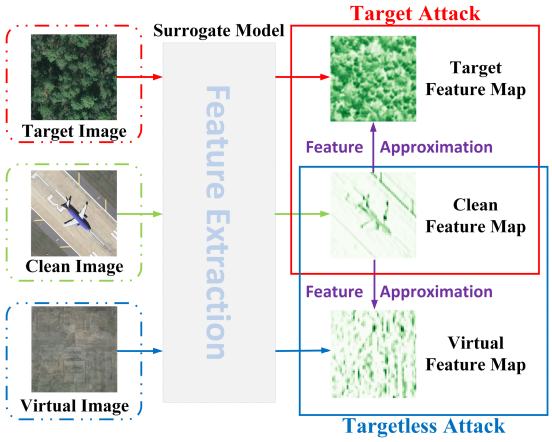


Fig. 1. Illustration of the FA attack is used for both targetless attack and targeted attack. Red lines indicate targeted attack process and blue lines indicate targetless attack process.

example generation methods are typically designed to attack specific DNNs, while these methods often fail to deceive other models, especially those that differ significantly from the original model. Furthermore, once the principles behind these white-box attack methods are understood, corresponding and effective defense mechanisms can be developed, rendering the attack methods ineffective. For example, defense distillation methods [27] have been successful in countering JSMA [28] and FGSM [15] algorithms, but subsequently, the C&W [29] adversarial attack algorithm managed to bypass defense distillation models.

The task of RSI classification is mainly based on the spatial and texture feature of the images. Therefore, for digital attacks, as long as the spatial texture feature of the image is changed, the image classification model can be successfully fooled. And the key to achieving black-box attacks lies in identifying common vulnerabilities between different network models. Yosinski et al. [30] found that different models have similar feature representations, and share common vulnerabilities. Based on this, Xu and Ghamisi [21] proposed a black-box attack, specifically using models for which complete knowledge is available as a springboard for attacks. After learning enough knowledge, the unknown model can be attacked. The method enhanced the transferability of the attack, however, it did not perform well in terms of the overall effectiveness of the attack. Inspired by the above-mentioned research works, we propose the feature approximation (FA) attack and benchmark its attack capability using multiple virtual images. Based on this, we incorporate the idea of multimodel fusion and propose the multimodel feature approximation (MFA) attack, dramatically improving the transferability of targetless attack. In addition, we found the effectiveness of FA for targeted attacks and realized targeted white-box attacks with a high success rate (SR). The flowchart of FA attack to realize targeted and targetless attacks is shown in Fig. 1.

This article presents several key contributions, which are summarized as follows.

1) We propose the FA attack, which approximates the input clean image features to a virtual image that do not belong to any category. We benchmark it using various featureless and featured images to verify its attack capability.
2) We combine the idea of multimodel fusion with FA to propose an MFA attack so that the generated adversarial examples consider the features of multiple models, effectively improving the transferability of targetless attack.
3) We apply the FA attack to targeted white-box attacks by randomly selecting a target category image as a virtual image. We conduct experiments on four models to achieve attacks with a higher SR than other methods.

The rest of this article is organized as follows. The research relating to this topic is reviewed in Section II. Section III describes the implementation details of the proposed adversarial attack methods. The results of the experiments and datasets used in this study are presented in Section IV. Finally, Section V concludes this article.

II. Related Work

A. Adversarial Attacks

1) L-BFGS: Szegedy et al. [14] proposed the box-constrained L-BFGS algorithm, which is the first adversarial attack method. Modeling the process of generating adversarial examples as an optimization problem to be handled. Due to the difficulty in solving the optimization problem, it is transformed into a box-constrained form. The ultimate mission is to find a minimal input perturbation within the constrained input space that is imperceptible. The minimization formula for the adversarial examples is

$$\arg\min_{\boldsymbol{r}} c \cdot ||\boldsymbol{r}||_2 + J\left(\boldsymbol{x}', \boldsymbol{\theta}, t\right), \text{s.t.} : \boldsymbol{x}' \in [0,1]^n \quad (1)$$

where all elements of the input image $\boldsymbol{x}$ are regularized in $[0,1]^n$, n denotes the pixels of $\boldsymbol{x}$, $J(\boldsymbol{x}', \boldsymbol{\theta}, t)$ is the final loss function of the target model, $\boldsymbol{\theta}$ indicates the parameters of the model, t is the misclassification label of the target. $\boldsymbol{r}$ refers to the perturbation, $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{r}$. For objectives with parameter $c > 0$ or more there is no guarantee that they are adversarial examples. The above-mentioned optimization process is performed in an iterative form.

2) Fast Gradient Sign Method (FGSM): Goodfellow et al. [15] proposed the FGSM. By overlaying the original image with the inverse gradient of the loss function, this technique produces effective adversarial examples. They also provide a rationalization for the existence of adversarial examples. Before them, DNNs were thought to be nonlinear, but they thought adversarial examples existed mainly because of the linear nature of these models high-dimensional spaces. Assuming an image $\boldsymbol{x}$, whose true label is y, we can calculate the perturbation by

$$\delta = \varepsilon \cdot \text{sign}(\bigtriangledown_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)) \quad (2)$$

where ε denotes the perturbation strength, $\text{sign}(\cdot)$ refers to the symbolic function, $\bigtriangledown_{\boldsymbol{x}} J(\cdot)$ denotes the computation of the gradient of the loss function.

Kurakin et al. [31] proposed the I-FGSM attack. Unlike the FGSM, which applies the gradient computation and updates only once, the I-FGSM increases the adversarial nature by iteratively

applying the perturbations of the FGSM. The iterative formula for the adversarial examples is as follows:

$$\boldsymbol{x}_{\text{adv}}^{t+1} = \text{clip}\left(\boldsymbol{x}_{\text{adv}}^{t} + \alpha \cdot \text{sign}\left(\nabla_{\boldsymbol{x}_{\text{adv}}} J\left(\boldsymbol{\theta}, \boldsymbol{x}_{\text{adv}}^{t}, y\right)\right)\right) \quad (3)$$

where clip denotes the clipping of the perturbation size to keep it within a finite range and α denotes the step size, when $t = 0$, $\boldsymbol{x}_{\text{adv}}^{t}$ is the clean image of the original input.

3) C&W: Carlini and Wagner [29] proposed the C&W attack algorithm, which successfully attacked the defense distillation model that was the most advanced at that time. The algorithm views the adversarial examples as a variable and transforms such problems into constraint minimization problems, which is defined as

$$\arg\min_{\boldsymbol{x}_{\text{adv}}} \left\|\boldsymbol{x}_{\text{adv}} - \boldsymbol{x}\right\|_{\infty} - \mu \cdot J\left(\boldsymbol{\theta}, \boldsymbol{x}_{\text{adv}}, y\right) \quad (4)$$

where μ refers to the weight coefficient.

B. Black-Box Attacks

Currently, most research on adversarial attacks are white-box attacks [15], [32], [33]. White-box attack methods generally have a high SR, but their practical utility is limited because it is challenging to obtain any information from the target model in real-world scenarios. Researchers have proposed black-box attack [34], [35], [36] methods in response to this, which assume that the information of the target model are unknown or only partially known. On the contrary, black-box attacks are generally more challenging but also more practical compared to white-box attacks. There are three main black-box attacks: Query-based attacks, transfer-based attacks, and decision-boundary-based attacks [37]. In this study, we primarily utilized the transfer-based attack approach.

1) Query-Based Attacks: It mainly uses querying the input–output information of the target model to get its approximate gradient by constantly approximating the attacked model numerically, and later using the estimated gradient to counter the attack. Chen et al. [38] proposed the zero-order-optimization attack, which is the first black-box attack based on gradient estimation. Su et al. [39] investigated the single-pixel attack and introduced a differential evolutionary lookup strategy to improve the attacking ability. Bai et al. [40] proposed a black-box attack based on a neural process that reduces the number of queries while improving the attack performance against samples.

2) Transfer-Based Attacks: The attack logic is extended by white-box attacks. As a general method, there will usually be a white-box model used as a surrogate model for the attack, and the adversarial examples will generally be migration-aggressive to other models as well. Papernot et al. [41] investigated the local surrogate model attack (LSMA) which is the earliest black-box attack. In this attack scenario, the attacker is granted access to a portion of the original training data and can also query and access the classification models. This information allows the attacker to craft specific attacks using the training data and test the effectiveness of these attacks by querying the model. The idea of integration was incorporated into the LSMA algorithm by Liu et al. [42]. They select multiple models at once and combine their loss values to carry out an attack. Shi et al. [43] inspired by the MI-FGSM attack approach, proposed the Curls&Whey black-box attack approach, which improves the adversarial examples transferability. Xu and Ghamisi [21] used the information of known models to discover the vulnerabilities of unknown models, and generated generic adversarial samples with good migration.

3) Decision-Boundary-Based Attacks: This method neither relies on an surrogate model nor requires a confidence score vector. It represents a more restricted adversarial scenario where only using the category labels from the output of the black-box classifier can achieve a successful attack. It is more in line with real-world scenarios, but the attack is more difficult and usually requires more queries. The pioneering decision-boundary-based attack boundary attack was introduced by Brendel et al. [44]. The attack relies only on the category labels output by the classification model and does not require information, such as gradient or confidence scores. Cheng et al. [45] proposed the opt-attack, which solves the problem that a boundary attack requires supermultiple model queries and cannot guarantee convergence improves the attack query efficiency. Li et al. [46] discovered that queries can be generated by introducing perturbations to the image and proposed query-efficient boundary-based blackbox attack, which greatly reduces the queries number.

C. Data Augmentation

Data augmentation refers to the process of generating additional training images by applying many variations to the original images, thereby increasing their diversity [47]. These variations do not drastically alter the features of the original image, but rather adjust the positioning of the featured components of the image accordingly, altering their distribution. The discovery of data augmentation techniques has greatly enhanced the model generalization ability and data robustness. Data augmentation methods mainly include traditional data augmentation and deep learning-based data augmentation [48]. This article focuses on using traditional data augmentation methods.

1) Traditional Data Augmentation: These methods are techniques commonly utilized in traditional computer vision tasks to enhance the data in several ways. They are categorized into single-sample transformation and multisample fusion [48]. The former refers to transforming and expanding a single sample to generate more diverse training data. These methods are usually used when the dataset is small or the sample is insufficient to generate additional samples by transforming operations on a single sample to enrich the variety and size of datasets. The specific measures are rotating, flipping, cropping, translating, scale changing, brightness adjusting, noise addition, and other operations. The latter refers to the method of extracting multiple images and ultimately fusing them into a new image to expand the dataset and improve the training effect. Zhang et al. [49] proposed the Mixup data augmentation method, in which two images with different categories are proportionally superimposed to form a new sample, and the labeling categories of the new sample are also proportionally composed of the original two labels. Yun et al. [50] proposed the CutMix method, whose idea is to splice the images of different categories after

Algorithm 1: Targetless Black-Box Attack MFA.

Input:
-The image $\boldsymbol{x}$ to be attacked and the true label y.
-Parameters of the surrogate model $\boldsymbol{\theta}_{si}$, feature extraction function g_{si}, prediction function p_{si}.
-Virtual image $\widetilde{\boldsymbol{x}}$ generated by the data enhansement methods.

$g_0 \leftarrow 0,\ \boldsymbol{x}_{adv}^0 \leftarrow \boldsymbol{x},\ I \leftarrow 5,\ \alpha \leftarrow 1,\ \beta \leftarrow 0.1.$
for t in $range(0, I)$ **do**
  Calculate the mix loss $\mathcal{L}_{Mixs}(\boldsymbol{\theta}_s, \boldsymbol{x}, \widetilde{\boldsymbol{x}})$ and predicted loss $\mathcal{L}_{CEs}(\boldsymbol{\theta}, \boldsymbol{x}, y)$ through Eqs. (11) and (12).
  Calculate the overall loss $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y)$ through Eq. (13).
  The momentum term m_{t+1} is updated through (9) and the adversarial example $\boldsymbol{x}_{adv}^t$ is updated through Eq. (10).
end for
return $\boldsymbol{x}_{adv}^t$

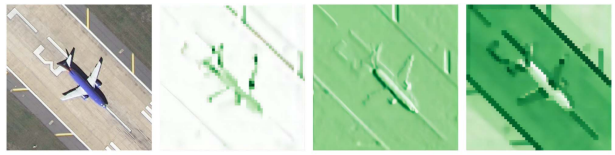
Algorithm 2: Targeted White-Box Attack FA.

Input:
-The attacked image $\boldsymbol{x}$ and the target category image $\boldsymbol{T}$, and the true label y and the target label t.
-Parameters of the surrogate model $\boldsymbol{\theta}_{si}$, feature extraction function g_{si}, prediction function p_{si}.

$g_0 \leftarrow 0, \boldsymbol{x}_{adv}^0 \leftarrow \boldsymbol{x}, I \leftarrow 5, \alpha \leftarrow 1, \beta \leftarrow 0.1, \gamma \leftarrow 0.4.$
for t in $range(0, I)$ **do**
  Calculate the mix loss $\mathcal{L}_{Mix\_T}(\boldsymbol{\theta}_s, \boldsymbol{x}, \widetilde{T})$, true pre-dicted loss $\mathcal{L}_{CE}(\boldsymbol{\theta}, \boldsymbol{x}, y)$ and target prediction loss $\mathcal{L}_T(\boldsymbol{\theta}, \boldsymbol{x}, t)$ through Eqs. (14), (6) and (15).
  Calculate the overall loss $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y)$ through Eq. (16).
  The momentum term m_{t+1} is updated through Eq. (9) and the adversarial example $\boldsymbol{x}_{adv}^t$ is updated through Eq. (10).
end for
return $\boldsymbol{x}_{adv}^t$

intercepting them, which improves the Mixup method and makes the generated images more natural. Harris et al. [51] proposed the FMix method, which fourier samples the low-frequency images to obtain binary templates, and uses this as the basis for the image interpolation, which outperforms both the Mixup and the CutMix methods.

2) Deep Learning-Based Data Augmentation: These methods aim to increase the diversity and complexity of training by expanding the training dataset in the same way as traditional methods. Still, the two are entirely different in terms of their implementation methods. The former performs random transformations and deformations on the images to make the training more complex and diverse. It does not require labeling information, also known as unsupervised data augmentation. Goodfellow et al. [52] proposed the generative adversarial networks. The network consists of a generator, which tries to generate samples that resemble actual samples, and a discriminator, which distinguishes between generated and actual samples. Adversarial training of generators and discriminators allows for generating samples with a high degree of diversity. Cubuk et al. [53] proposed the AutoAugment method, which automatically finds the best data augmentation strategy through a search algorithm, including rotations, clipping, scaling, flipping, etc., as well as other more complex distortions and deformations.



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

Fig. 2. Original input image and feature maps extracted by three surrogate models respectively. (a) Origin. (b) Alexnet. (c) ResNet. (d) DenseNet.

III. Methodology

Modifying the pixel data of images is a common adversarial attack method, which can cause the surrogate model to produce false predictions [39]. However, the impact of this attack on the victim model may be limited because of the differences between different neural networks [25]. To enhance the transferability of adversarial attack, it is possible to consider attacking the shallow features of the surrogate model. Shallow features share similar representations in different networks and preserve detailed spatial information of images [30]. Therefore, they may also contain more similar vulnerabilities. Feature maps extracted by various models are shown in Fig. 2.

Based on this, we propose the FA strategy to generate adversarial examples. The idea is to approximate the input clean image features to a virtual image by minimizing the distance between them. Since this virtual image is carefully designed by us and does not belong to any category, this method can realize the adversarial attack. We first benchmark FA attack by varying the virtual image to determine its effect on the capability of FA attack. To enhance the attack transferability, we propose the MFA attack, where three models with different architectures are chosen as surrogate models and learned simultaneously so that the adversarial examples take into account multimodel features. In addition, we found the effectiveness of FA attack in realizing the targeted attack by using the image of the target category as the virtual image. The targeted attack is done by making the input image features approximate to the target image. This section details the proposed methodology.

A. FA Benchmark

First, we examine the impact of various virtual images on FA attack to form a benchmark for relevant researchers.

1) Generate Virtual Image: Xu and Ghamisi [21] studied the impact of Mixup and CutMix data augmentation methods, based on which we perform an extended study. This test is split into two parts. In the first part, we use six featureless images as the

(a) (b) (c)

Fig. 3. Virtual images generated by three data augmentation methods respectively. (a) FMix. (b) Mosaic. (c) AugMix.

virtual images to approximate the features of the input clean images to the featureless images for attack. The featureless images are pure color images of black, white, red, green, and blue and random Gaussian noise images. In the second part, we use three multisample data augmentation methods to generate virtual images for testing, they are FMix [51], Mosaic [54], and AugMix [55].

FMix [51] data augmentation method: Two categories are randomly selected in the dataset, one image is taken from each category. The mask image is binarized according to the high and low-frequency regions of the image, then the pixels are combined in a weighted manner using this mask. For the mask image we use the example mask in text.

Mosaic [54] data augmentation method: Four categories are randomly selected in the dataset, one image is extracted from each category. The four images are randomly flipped, scaled, color gamut transformed, and other operations are performed. The final images are placed according to the order of the first image on the top-left, the second image on the top-right, the third image on the bottom-left, and the last image on the bottom-right. The final image should be the same scale of size as the original image.

AugMix [55] data augmentation method: The original method is to process one image, we use multiple images superimposed here. Four categories are randomly selected from the dataset, and for each category we select one image. The first image is not subjected to any operation. The last three images are subjected to translation, rotation, and color gamut transformation operations, after which the last three images are superimposed according to the random weights, but make sure the weights sum up to 1. The combined image formed is superimposed with the first image according to the weights of 0.5 and 0.5 to create the virtual image we need. The virtual images formed by the three methods are shown in Fig. 3.

2) Design Loss Function: For the adversarial examples to have close features to our well-designed virtual image, we need to minimize the difference in distribution between them. We design the mixture loss function as

$$\mathcal{L}_{\text{Mix}}\left(\boldsymbol{\theta}_s, \boldsymbol{x}, \tilde{\boldsymbol{x}}\right) = -\sum_{r=1}^{n_r}\sum_{c=1}^{n_c}\sum_{k=1}^{n_k} g_s(\boldsymbol{x})^{(r,c,k)} \log \frac{g_s(\boldsymbol{x})^{(r,c,k)}}{g_s(\tilde{\boldsymbol{x}})^{(r,c,k)}} \tag{5}$$

where n_r, n_c, n_k denotes the number of rows, columns, and channels of the feature map, respectively. $\boldsymbol{\theta}_s$ refers to the parameters of the surrogate model, and $\boldsymbol{x}$ is the clean image, $\tilde{\boldsymbol{x}}$ for the well-designed virtual image. $g_s(\boldsymbol{x})$ denotes the mapping function for extracting shallow features. The first pooling layer of the model is used to extract the shallow features of the image. We only need to minimize the distances between the virtual image $\tilde{\boldsymbol{x}}$ and the clean image $\boldsymbol{x}$ to achieve our goal. We utilize KL-divergence to calculate the distance.

In addition, we need to do an auxiliary attack against the predictions of model, which we compute using the cross-entropy loss

$$\mathcal{L}_{\text{CE}}\left(\boldsymbol{\theta}_s, \boldsymbol{x}, y\right) = -\sum_{i=1}^{n_j} y^{(i)} \log q_s(\boldsymbol{x})^{(i)} \tag{6}$$

where n_j denotes the number of categories in the classification mission, and y is the one-hot encoding of the true label of the input image $\boldsymbol{x}$. $q_s(\boldsymbol{x})$ s denotes the prediction function of the model for the input image $\boldsymbol{x}$.

The final loss function $\mathcal{L}$ is the combination of $\mathcal{L}_{\text{Mix}}$ and $\mathcal{L}_{\text{CE}}$

$$\mathcal{L}\left(\boldsymbol{\theta}_s, \boldsymbol{x}, y\right) = \mathcal{L}_{\text{Mix}}\left(\boldsymbol{\theta}_s, \boldsymbol{x}, \tilde{\boldsymbol{x}}\right) + \beta \cdot \mathcal{L}_{\text{CE}}\left(\boldsymbol{\theta}_s, \boldsymbol{x}, y\right) \tag{7}$$

where β is the weight parameter of the cross-entropy loss.

We can attain the adversarial example by adding the gradient of the final loss to the input image

$$\boldsymbol{x}_{\text{adv}}^{t+1} = \text{clip}\left(\boldsymbol{x}_{\text{adv}}^{t} + \alpha \cdot \frac{\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{\theta}_s, \boldsymbol{x}_{\text{adv}}^{t}, y)}{\|\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{\theta}_s, \boldsymbol{x}_{\text{adv}}^{t}, y)\|_{\infty}}\right) \tag{8}$$

where $\text{clip}(\cdot)$ denotes the example is clipped to keep it within a certain range. $\boldsymbol{x}_{\text{adv}}^{t}$ denotes the adversarial example formed in the tth iteration. When $t = 0$, adversarial example is the input clean image and α is the step size.

We introduce the momentum attack [56] to enhance the attack transferability, which can stabilize the updating direction of the gradient and eliminate the worst local maxima. We set the momentum term of the tth iteration to be $\boldsymbol{m}_t$. When $t = 0, \boldsymbol{m}_t = 0$. After that, using the gradient direction of the velocity vector to update $\boldsymbol{m}_t$

$$\boldsymbol{m}_{t+1} = \boldsymbol{m}_t + \frac{\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{\theta}_s, \boldsymbol{x}_{\text{adv}}^{t}, y)}{||\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{\theta}_s, \boldsymbol{x}_{\text{adv}}^{t}, y)||_1}. \tag{9}$$

Use this as a basis for updating the adversarial example

$$\boldsymbol{x}_{\text{adv}}^{t+1} = \text{clip}\left(\boldsymbol{x}_{\text{adv}}^{t} + \alpha \cdot \frac{\boldsymbol{m}_{t+1}}{||\boldsymbol{m}_{t+1}||_{\infty}}\right). \tag{10}$$

B. Targetless Black-Box Attack

We combine the idea of FA with multimodel fusion to propose the MFA attack. Fig. 4 shows the flowchart of the targetless black-box attack MFA.

1) Generate Virtual Image: For comparison, we use the virtual images of the Mixup and Mixcut attacks as our virtual images. We refer to these approaches as MFA+Mixup and MFA+Mixcut.

The Mixup attack generates the virtual image in an easy-to-understand way, one image is extracted from each of the ten different categories of the training set, and then the ten images are superimposed, and the transparency of each image is set to 0.1 to form a mixed image. The virtual image of the Mixcut attack was stitched together from ten different categories of images, and one-tenth of the image is cut from each image, and different

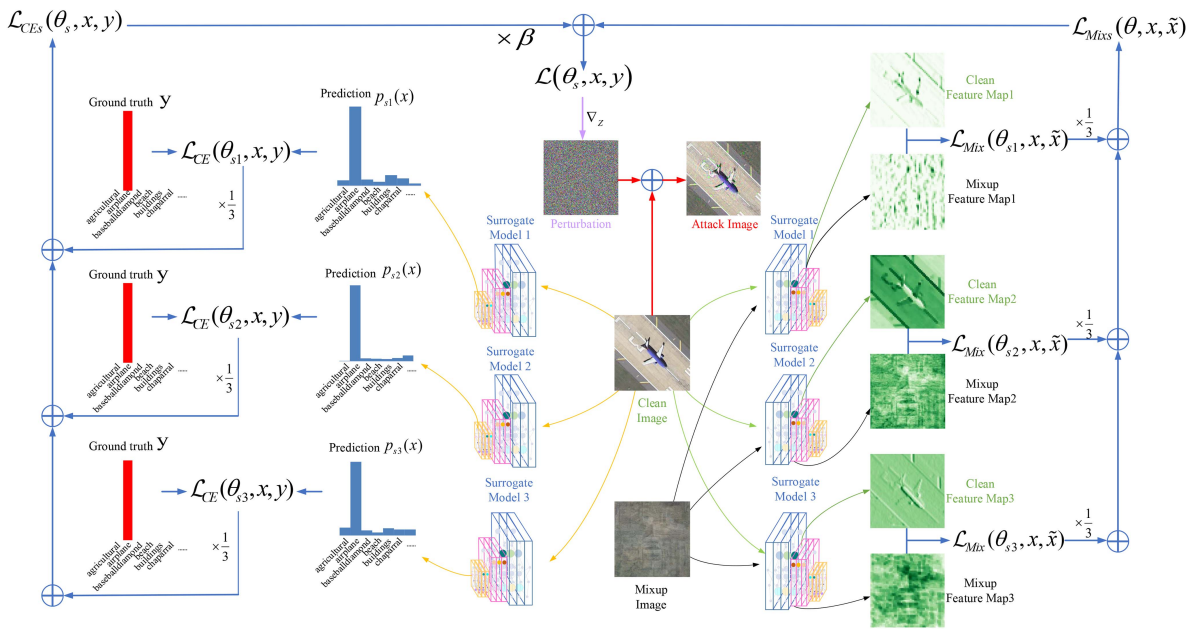


Fig. 4. Flowchart of targetless black-box attack MFA. The green and black lines indicate the feature extraction process, the orange lines indicate the model prediction process, the blue lines indicate the loss computation (the left-hand side is the prediction loss, and the right-hand side is the feature loss), the purple lines indicate the computation of perturbation, and the red lines indicate the final generation of the adversarial example.

positions of each image are selected depending on the order of the cuts. The width of the image is kept constant when cutting, and the height is intercepted as one-tenth of the height of the image, and the size of the formed hybrid image is consistent with the original image.

2) Design Loss Function: We define the mixture loss function as

$$\mathcal{L}_{\text{Mixs}}(\boldsymbol{\theta}, \boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{1}{n_{\text{Models}}} \sum_{i=1}^{n_{\text{Models}}} \mathcal{L}_{\text{Mix}}(\boldsymbol{\theta}_{si}, \boldsymbol{x}, \tilde{\boldsymbol{x}}) \tag{11}$$

where n_{Models} denotes the number of surrogate models, $\tilde{\boldsymbol{x}}_i$ denotes the image features extracted by the ith model.

Each iteration involves calculating the distance between the input and virtual image features extracted by different models and approximating them simultaneously.

In addition, we use the cross-entropy loss between the clean image prediction logits values of the multimodel and the true labels to carry out ancillary attack. The cross-entropy loss function is

$$\mathcal{L}_{\text{CEs}}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \frac{1}{n_{\text{Models}}} \sum_{i=1}^{n_{\text{Models}}} \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}_{si}, \boldsymbol{x}, y). \tag{12}$$

The final loss function $\mathcal{L}$ is the combination of $\mathcal{L}_{\text{Mixs}}$ and $\mathcal{L}_{\text{CEs}}$

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \mathcal{L}_{\text{Mixs}}(\boldsymbol{\theta}, \boldsymbol{x}, \tilde{\boldsymbol{x}}) + \beta \cdot \mathcal{L}_{\text{CEs}}(\boldsymbol{\theta}, \boldsymbol{x}, y). \tag{13}$$

The adversarial example is generated by adding the gradient of final loss function to the original clean image. In addition, we also use the momentum attack to stabilize the gradient direction.

Algorithm 1 shows the implementation details of the targetless black-box attack MFA.

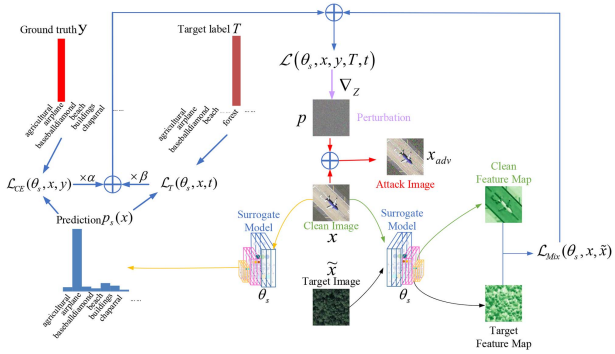


Fig. 5. Flowchart of targeted white-box attack FA. Using one model as the surrogate model. Compared to the targetless attack, the targeted attack incorporates the true label of the target as the forward direction of the prediction iteration and the true label of the original image as the reverse direction of the prediction iteration.

C. Targeted White-Box Attack

We also found the effectivity of FA attack for targeted attack. Fig. 5 shows the flowchart of the targeted white-box attack FA.

1) Generate Virtual Image: To realize the targeted attack, we need to approximate the features of the clean image $\boldsymbol{x}$ to the target image $\boldsymbol{T}$. Therefore, we randomly select a target category image as the virtual image.

2) Design Loss Function: We define the mixture loss function as

$$\mathcal{L}_{\text{Mix\_}T}(\boldsymbol{\theta}_s, \boldsymbol{x}, \boldsymbol{T}) = \mathcal{L}_{\text{Mix}}(\boldsymbol{\theta}_s, \boldsymbol{x}, \tilde{\boldsymbol{x}})\big|_{\tilde{\boldsymbol{x}}=\tilde{\boldsymbol{T}}}. \tag{14}$$

The cross-entropy loss function is divided into two parts. For the first part, we maximize the cross-entropy loss $\mathcal{L}_{\text{CE}}$ between the predicted logits values of the clean images and the true labels as before. The next part, we minimize the cross-entropy loss $\mathcal{L}_T$

between the predicted logits values of the clean images and the target class labels. The target cross-entropy loss function is

$$\mathcal{L}_T\left(\boldsymbol{\theta}_s, \boldsymbol{x}, t\right) = -\left.\mathcal{L}_{\mathrm{CE}}\left(\boldsymbol{\theta}_s, \boldsymbol{x}, y\right)\right|_{y=t} \tag{15}$$

where t denotes the one-hot encoding of the target category label. The final loss function $\mathcal{L}$ is

$$\mathcal{L}(\boldsymbol{\theta}_s, \boldsymbol{x}, y, \boldsymbol{T}, t) = \mathcal{L}_{\mathrm{Mix\_}T} + \beta \cdot \mathcal{L}_{\mathrm{CE}} + \gamma \cdot \mathcal{L}_T \tag{16}$$

where β is the weight of the true cross-entropy loss function and γ is the weight of the target cross-entropy loss function.

The adversarial example is generated by adding the gradient of final loss function to the original clean image. In addition, we also use the momentum attack to stabilize the gradient direction.

Algorithm 2 shows the implementation details of the targeted white-box attack FA.

In order to attack the entire dataset, we need to traverse all the images in the dataset, each with a different adversarial perturbation. The larger the dataset, the longer the algorithm takes to run.

IV. Experiments

A. Dataset

We used two RSI datasets for scene classification, UC Merced (UCM)[1] and AID.[2]

1) UC Merced Dataset: The UCM [62] dataset comprises a high-resolution RSI encompassing 21 distinct scene categories. It includes a total of 2100 images, with 100 images per category. The dataset is derived from large images in the USGS National Map Urban Areas image set and represents urban areas across different regions in the United States. Each image in the UCM dataset has a resolution of 256×256 pixels and is supplied in a lossless compressed format.

2) AID Dataset: The AID [63] dataset is a recently compiled collection of aerial images by Google Earth. It encompasses 30 distinct scene categories, with varying image counts for each category, ranging from a minimum of 220 to a maximum of 400. It comprises 10 000 images, each with a resolution of 600 × 600 pixels, and saved in PNG format.

B. Experimental Settings

For comparison experiments, we use six adversarial attacks, which are FGSM [15], I-FGSM [31], C&W [29], PGD [57], Mixup [21], and Mixcut attack [21]. We conduct experiments on two remote sensing datasets, UCM and AID. For each dataset, we selected 30% images as the training set, 30% images as the validation set and the other 40% images as the test set. We select AlexNet [58], ResNet18 [59], DenseNet121 [60], and RegNetX-400MF [61] four models with different architectures as surrogate models. As in Xu and Ghamisi's [21] work, the iteration step size α is 1. For methods that require iterations such as I-FGSM, C&W, PGD, Mixup attack, and Mixcut attack, we uniformly set the iteration number I to 5. We use the first pooling layer of the model to define the feature extraction function $g_s(\boldsymbol{x})$. For the evaluation metric, the attack SR were used

$$\mathrm{SR} = \frac{n_{\mathrm{wrong}}}{n_{\mathrm{total}}} \tag{17}$$

where n_{total} denotes the total number of examples involved in the test, and n_{wrong} denotes the number of misclassified examples. Higher SR indicates the method is more effective.

This research was conducted through the Pytorch platform, using one NVIDIA GeForce RTX 3060 (12 GB) GPU.

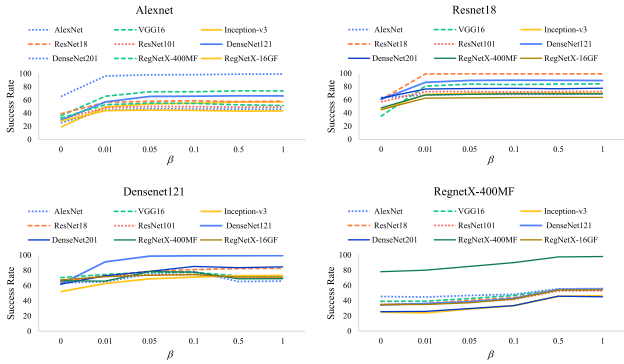


Fig. 6. Impact of β on SR of targetless attack in different surrogate models is obtained by choosing four surrogate models to attack nine models.

C. Adversarial Attacks

1) FA Benchmark: Tables I and II present the detailed results of the attack using featureless virtual images and virtual images generated by three data augmentation methods. We can observe that the attack results of red, green, and blue images are poor compared to the black and white ones because the direction of FA is different in the three dimensions of the image, which may lead to the averaging of features. The reason that the attack results of black and white images are not much different from each other is that the calculation of KL-divergence considers the absolute distance. For the black image, the features are overall approximated to the low, while the white image is precisely the opposite, which leads to a slight difference in the attack results. Therefore, we conducted relevant experiments on black, red, green, and blue virtual images only in two datasets for the AlexNet surrogate model. For the rest of the surrogate models, we selected white images and random Gaussian noise images as the virtual images.

The experiment results show that changing only the virtual image has a limited effect on the SR of the attacks, so our study can be used as a benchmark for the research of the related people afterward. The weighting factor β in the loss function controls the weighting between the feature loss and the prediction loss, both of which are related to the surrogate model we selected. So we can conclude that different surrogate models should be chosen with different β. According to the experimental results, we set the β of the AlexNet, DenseNet121, and ResNet18 models to 0.1 and the β of the RegNetX-400MF model to 0.5. Fig. 6 shows the experiment results, and we use the UCM dataset for the experiments.

[1][Online]. Available: http://weegee.vision.ucmerced.edu/datasets/landuse.html

[2][Online]. Available: https://captain-whu.github.io/AID/

TABLE I
BENCHMARK OF DIFFERENT VIRTUAL IMAGE IN FA ATTACK ON THE UCM DATASET

Surrogate Model	Method	AlexNet	ResNet18	DenseNet 121	RegNet X-400MF	VGG16	Inception -v3	ResNet 101	DenseNet 201	RegNet X-16GF
AlexNet	Mixup [21]	98.62	58.76	60.48	54.29	69.43	51.52	50.00	46.67	44.86
	Mixcut [21]	87.52	38.38	32.10	37.62	42.57	31.33	35.62	26.67	30.19
	White	99.43	58.29	63.05	56.48	73.71	57.24	52.48	46.48	44.19
	Black	97.62	58.10	61.14	50.57	71.52	55.62	51.52	44.57	43.33
	Noise	91.43	61.90	51.14	53.05	52.48	40.86	52.76	56.00	51.90
	Red	87.62	52.95	31.71	33.90	42.29	40.19	37.62	26.86	28.76
	Green	89.90	62.10	33.43	30.95	56.19	46.48	41.43	24.95	36.38
	Blue	89.33	55.14	35.62	36.67	42.00	41.24	40.67	29.43	33.43
	FMix [51]	97.71	58.10	64.86	54.76	72.48	54.38	50.19	45.71	42.38
	Mosaic [54]	98.48	58.76	65.71	55.14	73.43	55.05	49.90	46.29	44.78
	AugMix [55]	97.71	58.1	61.90	54.76	70.48	54.10	49.90	46.76	45.90
ResNet18	Mixup [21]	68.57	99.43	88.67	67.71	83.62	70.10	72.38	76.67	63.62
	Mixcut [21]	68.38	99.43	89.24	67.71	83.24	69.62	72.19	75.76	63.71
	White	68.19	99.43	88.95	67.43	83.14	70.10	72.38	76.86	63.14
	Noise	68.00	99.52	87.43	67.43	80.76	68.95	72.86	75.90	63.14
	FMix [51]	68.10	99.51	89.33	68.19	83.14	70.48	71.90	77.05	63.62
	Mosaic [54]	68.57	99.36	89.24	68.19	82.95	69.71	72.29	76.86	63.60
	AugMix [55]	67.81	99.43	89.14	67.69	83.71	69.71	71.62	76.67	62.95
DenseNet121	Mixup [21]	64.76	82.67	99.33	69.14	77.62	72.76	70.29	85.52	71.71
	Mixcut [21]	65.43	82.48	99.33	68.29	78.10	72.86	70.00	85.05	71.24
	White	65.14	81.71	99.33	68.38	78.10	73.24	69.24	85.52	71.14
	Noise	65.62	82.00	99.33	69.24	77.43	73.24	70.00	85.24	71.71
	FMix [51]	65.05	82.86	99.24	68.67	77.14	72.95	70.38	85.04	71.33
	Mosaic [54]	65.52	82.38	99.33	69.05	77.71	73.14	70.28	85.62	71.52
	AugMix [55]	65.05	82.67	99.33	69.33	77.33	73.43	70.10	85.05	71.43
RegNetX-400MF	Mixup [21]	51.24	56.38	58.28	96.19	56.19	49.71	54.76	49.90	56.19
	Mixcut [21]	51.14	50.10	52.10	96.67	49.71	41.71	49.81	44.29	49.71
	White	53.24	53.05	58.95	93.81	53.33	49.24	50.10	47.05	54.10
	Noise	53.42	56.19	61.43	92.67	53.33	49.14	51.71	53.14	56.10
	FMix [51]	54.76	56.57	58.29	96.76	55.33	48.76	52.86	50.38	56.48
	Mosaic [54]	50.95	51.81	52.86	96.38	51.71	42.00	50.00	43.05	50.10
	AugMix [55]	54.00	56.38	57.14	97.14	55.71	48.19	53.43	48.57	55.33

Note: The results are shown as SR (%).

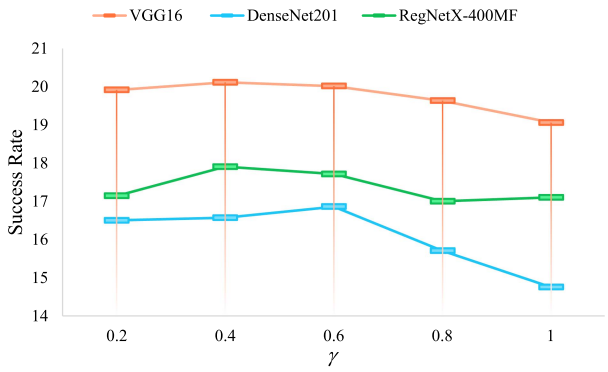


Fig. 7. Impact of γ on the SR of targeted attacks, the Alexnet model was chosen as a surrogate model to attack three models, VGG16, DenseNet201, and RegNetX-400MF.

2) Targetless Black-Box Attack MFA: We fused three models, AlexNet, DenseNet121, and ResNet18, as surrogate models to perform targetless black-box attack, and the detailed results are shown in Tables III and IV. Due to the validation in Xu and Ghamisi's [21] work showing that the performance of Mixup and Mixcut attacks is significantly superior to classical adversarial algorithms, such as PGD and C&W, we only compare Mixup and Mixcut algorithms. Example images and generated adversarial examples from the two datasets are shown in Figs. 8 and 9. Comparing the results with single surrogate model attacks, we can see that the black-box attack performance of MFA attack has shown an overall improvement, while the white-box attack performance has been slightly weakened, but to a lesser extent. The weakening of the effectiveness of the white-box attack is a contradiction between global optimization and local optimization. Specifically, when only one model is used as the surrogate model, the weight of the loss function of the model is 1, and when three models are used, the weight of the loss function of each model becomes 0.33, so that in the case of the same number of iterations, the latter will be slower than the former for the update of a single model, and the effect will be weakened. In addition, because of the different architectures of multiple models, when targeting a certain pixel point, the update direction of one model may conflict with another model, and after the neutralization of one positive and one negative, it will cause this iteration to lose its significance at that pixel point, leading to the weakening of the effect of the attack. The attack efficacy of the three fused surrogate models have an SR above 97%, and most white-box attacks in the AID dataset have an SR above 99%. That means our method maintains the performance of the white-box attacks. Overall, the SR of the MFA+Mixup attack improved by more than 15% compared to Mixup attack, and the SR of the MFA+Mixcut attack improved by more than 18% compared to Mixcut attack. The black-box attack SR overall improved by more than 15%, showing a more significant improvement. However, due to the fusion of multiple

TABLE II
BENCHMARK OF DIFFERENT VIRTUAL IMAGE IN FA ATTACK ON THE AID DATASET

Surrogate Model	Method	AlexNet	ResNet18	DenseNet 121	RegNet X-400MF	VGG16	Inception -v3	ResNet 101	DenseNet 201	RegNet X-16GF
AlexNet	Mixup [21]	100	84.06	73.14	73.64	80.56	70.90	70.70	80.12	77.60
	Mixcut [21]	100	83.78	71.68	73.96	80.94	70.48	71.68	79.80	77.72
	White	100	83.96	73.44	73.14	80.40	70.42	70.42	80.54	76.96
	Black	100	84.12	73.02	73.50	80.32	70.16	70.64	80.40	77.00
	Noise	100	85.24	72.90	73.86	82.14	71.50	72.44	80.16	79.18
	Red	99.98	85.2	72.78	73.24	81.86	70.36	72.12	78.98	72.50
	Green	98.52	80.46	73.22	71.18	77.58	68.24	71.58	77.58	75.28
	Blue	99.96	85.12	71.90	72.86	81.62	70.14	72.56	79.24	77.62
	FMix [51]	100	82.04	72.42	72.54	84.96	70.50	70.66	78.50	77.70
	Mosaic [54]	100	84.88	72.02	77.58	83.46	69.94	77.20	79.32	78.86
	AugMix [55]	100	84.42	73.48	74.52	81.34	71.24	71.40	80.56	78.08
ResNet18	Mixup [21]	86.22	96.66	83.52	81.42	87.72	78.04	87.08	82.94	86.04
	Mixcut [21]	78.96	99.86	89.38	81.20	85.84	79.30	84.56	92.04	86.60
	White	78.94	99.86	89.26	81.28	85.68	78.88	84.42	92.22	86.38
	Noise	79.62	99.88	89.28	81.24	85.52	79.78	84.80	91.96	86.42
	FMix [51]	78.72	99.88	90.02	81.36	85.78	78.86	84.58	92.12	86.66
	Mosaic [54]	78.76	99.82	89.62	81.50	85.86	79.24	84.40	92.22	86.62
	AugMix [55]	78.88	99.88	89.72	81.62	85.92	78.80	84.60	92.20	86.90
DenseNet121	Mixup [21]	82.1	92.30	99.44	80.42	82.86	77.04	84.00	91.02	87.34
	Mixcut [21]	80.42	92.78	100	80.50	82.44	77.30	82.40	91.98	86.98
	White	80.12	93.02	100	80.44	82.84	77.64	82.24	92.08	86.96
	Noise	80.32	93.04	100	80.36	82.58	77.44	82.16	91.98	86.78
	FMix [51]	80.24	92.96	100	80.16	82.64	77.51	82.16	91.96	86.90
	Mosaic [54]	80.3	92.78	100	80.58	82.70	77.26	82.00	92.12	87.08
	AugMix [55]	80.32	92.84	100	80.68	82.76	77.00	81.90	92.06	86.98
RegNetX-400MF	Mixup [21]	66.26	80.16	71.32	99.88	80.76	61.96	72.60	80.54	84.62
	Mixcut [21]	64.66	76.64	68.44	99.84	79.72	58.72	69.20	76.28	80.60
	White	65.08	75.12	62.58	91.80	79.72	51.82	65.22	71.52	78.10
	Noise	66.44	78.88	63.54	91.96	81.64	52.36	67.16	73.28	81.48
	FMix [51]	65.58	78.26	70.04	99.86	80.46	58.26	69.90	77.88	83.04
	Mosaic [54]	66.38	80.14	71.84	99.90	80.98	61.36	71.40	79.54	83.76
	AugMix [55]	66.24	79.74	71.66	99.92	81.00	61.06	71.58	80.12	84.24

Note: The results are shown as SR (%).

TABLE III
SR (%) OF DIFFERENT UNTARGETED BLACK-BOX ATTACKS ON THE UCM DATASET

Surrogate Model	Method	AlexNet	ResNet18	DenseNet121	RegNetX-400MF	VGG16	Inception-v3	ResNet101	DenseNet201	RegNetX-16GF
AlexNet [58]	Mixup [21]	97.62	58.76	60.48	54.29	69.43	51.52	50.00	46.67	44.86
	Mixcut [21]	87.52	38.38	32.10	37.62	42.57	31.33	35.62	26.67	30.19
	PGD [57]	79.71	24.19	26.48	25.14	31.33	26.67	25.48	17.33	22.08
	C&W [29]	91.71	34.10	24.00	26.97	33.62	23.14	27.52	17.71	15.61
ResNet18 [59]	Mixup [21]	68.57	99.43	88.67	67.71	83.62	70.10	72.38	76.67	63.62
	Mixcut [21]	68.38	99.43	89.24	67.71	83.24	69.62	72.19	75.76	63.71
	PGD [57]	43.05	100.00	45.14	46.32	54.10	41.33	35.91	39.05	40.12
	C&W [29]	45.05	99.09	45.52	49.78	41.71	39.06	42.95	39.54	38.65
DenseNet121 [60]	Mixup [21]	64.76	82.67	99.33	69.14	77.62	72.76	70.29	85.52	71.71
	Mixcut [21]	65.43	82.48	99.33	68.29	78.10	72.86	70.00	85.05	71.24
	PGD [57]	36.86	39.62	89.81	42.36	38.57	31.91	32.00	41.71	35.86
	C&W [29]	38.76	51.81	98.95	39.69	37.52	28.38	38.05	52.38	37.09
RegNetX-400MF [61]	Mixup [21]	51.24	56.38	58.28	96.19	56.19	49.71	54.76	49.90	56.19
	Mixcut [21]	51.14	50.10	52.10	96.67	49.71	41.71	49.81	44.29	49.71
	PGD [57]	26.13	30.72	25.44	88.73	21.02	22.46	31.32	22.05	40.69
	C&W [29]	28.66	28.51	25.67	98.12	26.43	20.15	29.71	23.37	42.56
A+R+D	MFA+Mixup(ours)	98.29	98.57	98.67	78.19	88.76	81.05	78.48	88.19	75.24
	MFA+Mixcut(ours)	97.24	97.52	97.14	74.19	87.05	78.48	76.10	85.81	71.71

Note: 1) The white-box attack results are displayed in green font, while the black-box attack results are displayed in black font. 2) The data section is filled with a gradient of red, white, and blue colors, with darker shades of red indicating stronger attack capability and darker shades of blue indicating weaker attack capability. On the same row, the smaller the variation of red color, the better the attack transferability. 3) The gray area represents the surrogate model used. A denotes the AlexNet model, R denotes the ResNet18 model, and D denotes the DenseNet121 model. 4) The yellow area indicates the attack methods and the orange area represents the attacked models.

models, the computational effort for training is also elevated, and the training time is roughly two times higher compared to a single model.

The β setting of this method is the same as the FA attack.

In addition, we also examined how the type and quantity of fusion models affect the effectiveness of attacks. Table V shows the specific quantitative results. Based on the experiment results, we can observe that fusing AlexNet, DenseNet121, and ResNet18 models as the surrogate models is the most effective. When ResNet18 and RegNetX-400MF models work together, the effect worsens because the architecture of the two models is similar. The fusion will produce the phenomenon of overfitting, resulting in the deterioration of the attack effectiveness.

TABLE IV
SR (%) OF DIFFERENT UNTARGETED BLACK-BOX ATTACKS ON THE AID DATASET

Surrogate Model	Method	AlexNet	ResNet18	DenseNet121	RegNetX-400MF	VGG16	Inception-v3	ResNet101	DenseNet201	RegNetX-16GF
AlexNet [58]	Mixup [21]	100.00	84.06	73.14	73.64	80.56	70.90	70.70	80.12	77.60
	Mixcut [21]	100.00	83.78	71.68	73.96	80.94	70.48	71.68	79.80	77.72
	PGD [57]	99.02	37.68	36.80	40.94	38.60	34.97	31.88	32.46	46.56
	C&W [29]	100.00	52.06	39.80	43.05	54.00	50.60	44.48	45.68	44.59
ResNet18 [59]	Mixup [21]	86.22	96.86	83.52	81.42	87.72	78.04	87.08	82.94	86.04
	Mixcut [21]	78.96	99.86	89.38	81.20	85.84	79.30	84.56	92.04	86.60
	PGD [57]	44.64	86.77	46.98	53.94	51.32	42.04	47.40	51.76	59.61
	C&W [29]	50.12	100.00	66.34	55.46	60.56	53.28	63.56	66.84	57.45
DenseNet121 [60]	Mixup [21]	82.10	92.30	99.42	80.42	82.86	77.04	84.00	91.02	87.34
	Mixcut [21]	80.42	92.78	100.00	80.50	82.44	77.30	82.40	91.98	86.98
	PGD [57]	42.44	57.46	88.02	44.12	46.10	41.22	47.22	60.12	35.62
	C&W [29]	44.26	67.32	100.00	43.05	51.40	45.86	55.08	69.40	38.93
RegNetX-400MF [61]	Mixup [21]	66.26	80.16	72.32	99.96	80.76	61.96	72.60	80.54	84.62
	Mixcut [21]	64.66	76.64	68.44	99.84	79.72	58.72	69.20	76.28	80.60
	PGD [57]	26.12	32.33	30.45	95.61	46.34	40.07	30.47	40.63	51.02
	C&W [29]	35.12	32.14	28.44	93.43	42.85	37.52	25.89	42.15	55.93
A+R+D	MFA+Mixup(ours)	99.84	99.46	99.52	88.34	90.14	89.52	90.68	95.70	93.20
	MFA+Mixcut(ours)	99.90	99.28	99.50	96.50	90.24	88.38	90.66	95.40	91.82

Note: 1) The white-box attack results are displayed in green font, while the black-box attack results are displayed in black font. 2) The data section is filled with a gradient of red, white, and blue colors, with darker shades of red indicating stronger attack capability and darker shades of blue indicating weaker attack capability. On the same row, the smaller the variation of red color, the better the attack transferability. 3) The gray area represents the surrogate model used. A denotes the AlexNet model, R denotes the ResNet18 model, and D denotes the DenseNet121 model. 4) The yellow area indicates the attack methods and the orange area represents the attacked models.

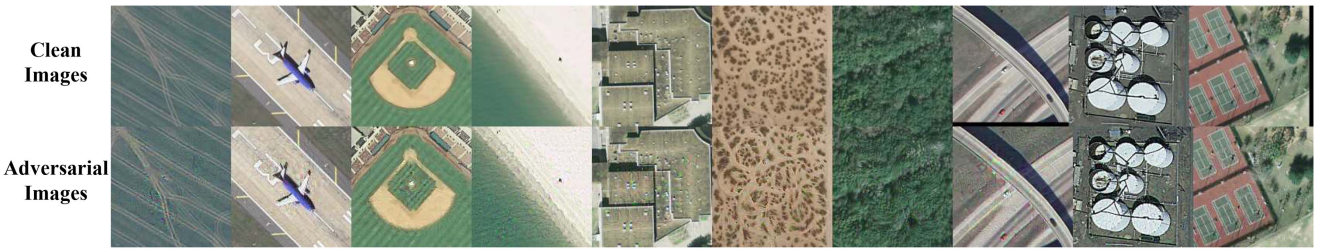


Fig. 8. Top images in each section showcase examples from the UCM dataset. The bottom images in each section display adversarial images generated using MFA+Mixup attack method.

Fig. 9. Top images in each section showcase examples from the AID dataset. The bottom images in each section display adversarial images generated using MFA+Mixup attack method.

TABLE V
SR (%) OF OF FA AND MFA ATTACKS WITH DIFFERENT SURROGATE MODELS ON THE UCM DATASET

Surrogate Model	Method	AlexNet	ResNet18	DenseNet 121	RegNet X-400MF	VGG16	Inception -v3	ResNet 101	DenseNet 201	RegNet X-16GF
AlexNet [58]	Mixup	97.62	58.76	60.48	54.29	69.43	51.52	50.00	46.67	44.86
ResNet18 [59]	Mixup	68.57	99.43	88.67	67.71	83.62	70.10	72.38	76.67	63.62
DenseNet121 [60]	Mixup	64.76	82.67	99.33	69.14	77.62	72.76	70.29	85.52	71.71
RegNetX-400MF [61]	Mixup	51.24	56.38	58.28	96.19	56.19	49.71	54.76	49.90	56.19
A+R1	MFA+Mixup(ours)	98.10	98.48	87.81	68.86	84.86	72.67	70.67	75.90	60.86
A+R1+D	MFA+Mixup(ours)	98.29	98.57	98.67	78.19	88.76	81.05	78.48	88.19	75.24
A+R1+R2	MFA+Mixup(ours)	94.10	90.86	77.14	87.71	75.05	66.38	66.29	67.62	61.33
A+D+R2	MFA+Mixup(ours)	93.81	73.52	95.62	87.14	74.57	66.57	65.43	80.95	73.14
R1+D+R2	MFA+Mixup(ours)	60.86	91.05	91.50	87.71	73.33	66.38	68.00	75.90	66.48
A+R1+D+R2	MFA+Mixup(ours)	94.29	92.48	92.57	88.10	80.57	74.38	73.24	79.05	68.67

Note: 1) The white-box attack results are displayed in green font, while the black-box attack results are displayed in black font. 2) The data section is filled with a gradient of red, white, and blue colors, with darker shades of red indicating stronger attack capability and darker shades of blue indicating weaker attack capability. On the same row, the smaller the variation of red color, the better the attack transferability. 3) The gray area represents the surrogate model used. A for the AlexNet model, R1 for the ResNet18 model, D for the DenseNet121 model, and R2 for the RegNetX-400MF model. 4) The yellow area indicates the attack methods and the orange area represents the attacked models.

Fig. 10. Top images in each section showcase examples from the UCM dataset. The bottom images in each section display adversarial images generated using FA attack method. The target category is forests.

Fig. 11. Top images in each section showcase examples from the AID dataset. The bottom images in each section display adversarial images generated using FA attack method. The target category is forests.

TABLE VI
SR (%) OF DIFFERENT TARGET ADVERSARIAL ATTACK METHODS ON THE UCM DATASET

Target	Method	AlexNet	ResNet18	DenseNet 121	RegNet X-400MF
Forest	FGSM [15]	15.81	15.52	8.00	9.14
	I-FGSM [31]	96.48	98.29	85.43	94.28
	PGD [57]	96.29	98.38	85.33	94.48
	C&W [29]	91.71	90.57	64.38	77.71
	FA(ours)	**99.05**	**99.90**	**97.81**	**98.67**
Agricu ltural	FGSM [15]	20.95	8.95	5.62	6.67
	I-FGSM [31]	99.71	**100**	86.76	96.57
	PGD [57]	98.86	98.76	71.14	88.48
	C&W [29]	99.81	99.90	87.81	96.57
	FA(ours)	**100**	**100**	**98.67**	**99.81**
Airplane	FGSM [15]	19.81	22.67	10.76	8.76
	I-FGSM [31]	91.62	**100**	97.62	98.67
	PGD [57]	88.00	98.57	95.24	95.90
	C&W [29]	92.00	99.62	97.52	98.76
	FA(ours)	**95.24**	98.71	**98.86**	**99.33**
Baseball diamond	FGSM [15]	10.29	26.00	17.62	13.14
	I-FGSM [31]	95.71	99.90	96.48	99.33
	PGD [57]	89.81	99.05	93.14	97.71
	C&W [29]	95.71	99.81	96.38	99.24
	FA(ours)	**99.14**	**100**	**99.05**	**99.43**
Beach	FGSM [15]	7.52	9.14	6.76	4.67
	I-FGSM [31]	85.43	96.67	91.05	87.81
	PGD [57]	73.05	91.24	81.24	56.19
	C&W [29]	84.76	96.67	91.14	86.29
	FA(ours)	**99.71**	**98.95**	**98.95**	**98.57**

Note: The best results are highlighted in **bold**.

TABLE VII
SR (%) OF DIFFERENT TARGET ADVERSARIAL ATTACK METHODS ON THE AID DATASET

Target	Method	AlexNet	ResNet18	DenseNet 121	RegNet X-400MF
Forest	FGSM [15]	6.50	9.98	5.04	6.78
	I-FGSM [31]	88.50	93.34	60.24	75.82
	PGD [57]	85.72	99.02	63.44	92.60
	C&W [29]	85.62	99.24	63.68	93.02
	FA(ours)	**98.56**	**100**	**91.02**	**99.48**
Airport	FGSM [15]	13.18	36.84	37.96	13.24
	I-FGSM [31]	89.56	99.86	99.70	98.40
	PGD [57]	84.58	99.18	99.90	92.92
	C&W [29]	89.06	99.88	99.66	98.24
	FA(ours)	**98.60**	**100**	**100**	**99.92**
Parking	FGSM [15]	20.12	11.22	8.70	9.18
	I-FGSM [31]	99.86	97.74	58.70	88.90
	PGD [57]	99.90	99.82	74.54	97.42
	C&W [29]	99.88	96.78	74.38	97.66
	FA(ours)	**100**	**100**	**95.02**	**99.92**
School	FGSM [15]	31.08	21.98	16.72	23.40
	I-FGSM [31]	99.78	97.28	85.24	97.28
	PGD [57]	99.50	97.48	85.66	97.26
	C&W [29]	99.20	92.38	77.44	93.58
	FA(ours)	**99.58**	**99.88**	**92.56**	**99.26**
Stadium	FGSM [15]	11.66	13.88	8.94	11.32
	I-FGSM [31]	98.28	98.26	93.18	91.74
	PGD [57]	98.30	98.46	93.10	92.34
	C&W [29]	95.38	94.06	86.16	82.92
	FA(ours)	**98.84**	**99.48**	**93.96**	**95.46**

Note: The best results are highlighted in **bold**.

3) Targeted White-Box Attack FA: For the targeted white-box attack, four models, AlexNet, DenseNet121, ResNet18, and RegNetX-400MF were used as the surrogate models, and five classes are randomly selected as the target categories in the UCM and AID datasets, respectively. We decided on four attack methods, FGSM, I-FGSM, C&W, and PGD, as comparison methods.

Example images and generated adversarial examples from the two datasets are shown in Figs. 10 and 11. Tables VI and VII show the specific experimental results. Based on

TABLE VIII
IMPACT OF MFA ON TARGETED ATTACK TRANSFERABILITY

Dataset	Surrogate Model	Method	Alexnet	Resnet18	Densenet 121	RegNet X-400MF	VGG16	Inception -v3	ResNet 101	DenseNet 201	RegNet X-16GF
UCM	Alexnet [58]	FA	**99.24**	40.67	17.43	17.90	20.10	26.29	16.57	15.71	27.52
	Resnet18 [59]		48.76	**99.71**	47.14	20.32	26.86	44.57	36.68	51.24	33.89
	Densenet121 [60]		44.67	64.95	**97.81**	18.96	21.33	43.90	37.71	64.57	19.77
	A+R+D	MFA(ours)	99.14	99.33	97.52	**55.24**	**43.24**	**57.05**	**54.67**	**72.48**	**55.05**
AID	Alexnet [58]	FA	**98.86**	83.54	46.02	22.14	70.70	35.84	60.88	49.96	32.95
	Resnet18 [59]		37.08	**100.00**	44.14	25.86	51.90	30.42	58.72	40.64	32.67
	Densenet121 [60]		42.44	69.60	**98.90**	19.62	52.91	40.62	47.63	45.16	35.43
	A+R+D	MFA(ours)	98.58	99.82	95.71	**59.48**	**83.39**	**61.78**	**84.14**	**80.76**	**60.07**

Note: A for the AlexNet model, R for the ResNet18 model, D for the DenseNet121 model. Results are reported as SR (%). The best results are highlighted in **bold**.

TABLE IX
ATTACK SPEED (IMAGES/S) OF DIFFERENT ATTACK ALGORITHMS

Method	UCM			AID		
	Alexnet	Resnet18	Densenet121	Alexnet	Resnet18	Densenet121
I-FGSM [31]	**17.76**	**12.84**	3.62	**17.45**	11.98	3.31
C&W [29]	16.53	12.76	**3.75**	16.22	**12.03**	**3.76**
PGD [57]	16.61	12.62	3.47	16.26	10.52	3.33
Mixup [21]	6.12	3.92	1.22	5.94	3.88	1.21
FA(ours)	5.95	3.81	1.06	4.75	3.98	1.23
MFA(ours)	0.67			0.62		

Note: An NVIDIA GeForce RTX 3060 (12GB) GPU was used in this study. The best results in the table are shown in **bold**.

the tables, we can obverse that our method has a better results than the comparison methods because we designed three parts for the attack. One approach is to minimize the KL-divergence between the feature representations of the clean image and the target image. Another method involves maximizing the cross-entropy loss between the predicted logits values of the model and the true label. Finally, an alternative approach is to minimize the cross-entropy loss between the predicted logits values of the model and the target label. In addition, our findings suggest that the SR of the attack is closely linked to the model complexity, with more complex models exhibiting a lower SR of the attack.

We also conduct a simple test of attack transferability by applying the idea of multimodel fusion. We selected forest as the target category and compared it with the results of the AlexNet, ResNet18, and DenseNet121 models. We found that after the multimodel fusion, the transferability is vastly improved. Table VIII shows the specific results. It should be noted that for FA attacks, we use three models individually as surrogate models, while for MFA attacks, we use a cascade of three models as surrogate models. Therefore, for attacks on the single model, the FA attack capability is stronger than the MFA attack, but the MFA attack only reduces the performance by 1%, so we can consider it to maintain high white-box attack performance.

In this method, we fix the β to 0.1 and set the γ to 0.4 according to the experimental results. To ensure fairness, we also put the true loss weight and the target loss weight in the loss function according to $1:4$ in the four comparison methods. Fig. 7 illustrates the influence of varying γ on the attack.

The attack speed of the algorithm is also a very important indicator, so we have measured the attack time of the algorithm. The data are shown by the average number of attack images per second. The attack speed of different algorithms is shown in Table IX. From the table, it can be seen that the more complex the model, the slower the attack speed of the algorithm. And the more complex the algorithm, the slower the attack speed. Besides, the attack SR is inversely proportional to the attack speed.

V. CONCLUSION

In this study, our focus is to explore methods for adversarial attack specifically in remote sensing. First, we propose the FA attack to approximate the input clean image features to a virtual image that does not belong to any category. We benchmark its attack capability by using multiple virtual images, including six featureless images and three featured images generated with three data augmentation methods. On this basis, we propose the MFA attack for the targetless black-box attack, and the results of fusing three models as surrogate models for attacking in two remote sensing datasets show that our method can enhance the attack transferability and the aggressiveness of the adversarial examples. Finally, we apply the FA attack in targeted white-box attack to attack four models. Compare it with four advanced methods, our method exhibits a higher attack SR. In addition, we also conduct a simple test on the targeted attack transferability. Using the multimodel fusion, we found that the attack transferability is also greatly improved. Our experiments also found that the more complex the structure of the deep learning model, the higher the resistance to black-box attacks.

VI. DISCUSSION

There are still some directions that can be improved in this study as follows.

1) We found that the ability of FA attack is limited for attacking scenarios such as forest, parking lot, and other scenarios where spatial features are distributed more uniformly. And if this defect is improved, a higher SR and transferability of the attack can be obtained.
2) We observe that the effect produced by each iteration diminishes as the iteration proceeds. Relevant researchers can try to improve it.
3) The transferability of targeted attacks is also a research hotspot. This article mainly performs white-box attacks in targeted attacks without doing much research on transferability, which is also a direction that can be improved. We hope this study can help discover the vulnerabilities of deep learning models and provide some inspiration and enlightenment for related researchers.

ACKNOWLEDGMENT

The authors would like to express our gratitude to Professor Shawn Newsam for providing the UCM dataset, which has been made publicly available for this research. In addition, they extend their thanks to Professor Gui-Song Xia for generously sharing the AID dataset, which has also been made publicly available for this study.

REFERENCES

[1] S. K. Roy, A. Deria, D. Hong, B. Rasti, A. Plaza, and J. Chanussot, "Multimodal fusion transformer for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5515620.
[2] 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.
[3] S. Mei, Y. Geng, J. Hou, and Q. Du, "Learning hyperspectral images from RGB images via a coarse-to-fine CNN," *Sci. China Inf. Sci.*, vol. 65, pp. 1–14, 2022.
[4] B. Wei et al., "CAM-PC: A novel method for camouflaging point clouds to counter adversarial deception in remote sensing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 56–67, 2024.
[5] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 91–124, Mar. 2022.
[6] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.
[7] S. Xie, M. Zhou, C. Wang, and S. Huang, "CSPPartial-YOLO: A lightweight YOLO-based method for typical objects detection in remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 388–399, 2024.
[8] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, "Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3735–3756, 2020.
[9] 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, 2022, Art. no. 5509612.
[10] J. Wang, W. Li, M. Zhang, R. Tao, and J. Chanussot, "Remote sensing scene classification via multi-stage self-guided separation network," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4500416.
[11] K. Han, Y. Li, and B. Xia, "A cascade model-aware generative adversarial example detection method," *Tsinghua Sci. Technol.*, vol. 26, no. 6, pp. 800–812, 2021.
[12] Q. He, X. Sun, Z. Yan, B. Li, and K. Fu, "Multi-object tracking in satellite videos with graph-based multitask modeling," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5619513.
[13] Q. Jin, Y. Han, W. Wang, L. Tang, J. Li, and C. Deng, "An occlusion-aware tracker with local-global features modeling in UAV videos," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 5403–5415, 2024.
[14] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representations*, 2013.
[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2014.
[16] L. Chen, G. Zhu, Q. Li, and H. Li, "Adversarial example in remote sensing image recognition," 2019, *arXiv:1910.13222*.
[17] S. Mei, J. Lian, X. Wang, Y. Su, M. Ma, and L.-P. Chau, "A comprehensive study on the robustness of image classification and object detection in remote sensing: Surveying and benchmarking," 2023, *arXiv:2306.12111*.
[18] J. Lian, X. Wang, Y. Su, M. Ma, and S. Mei, "Contextual adversarial attack against aerial detection in the physical world," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2023, pp. 6632–6635.
[19] H. Yuan et al., "Unified physical-digital attack detection challenge," 2024, *arXiv:2404.06211*.
[20] 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.*, 2018, pp. 408–411.
[21] 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.
[22] Y. Zhang et al., "Adversarial patch attack on multi-scale object detection for UAV remote sensing images," *Remote Sens.*, vol. 14, no. 21, 2022, Art. no. 5298.
[23] X. Sun, G. Cheng, L. Pei, H. Li, and J. Han, "Threatening patch attacks on object detection in optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5609210.
[24] J. Lian, S. Mei, S. Zhang, and M. Ma, "Benchmarking adversarial patch against aerial detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5634616.
[25] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," *IEEE Access*, vol. 9, pp. 155161–155196, 2021.
[26] H. Wang, S. Wang, Z. Jin, Y. Wang, C. Chen, and M. Tistarelli, "Similarity-based gray-box adversarial attack against deep face recognition," in *Proc. IEEE 16th Int. Conf. Autom. Face Gesture Recognit.*, 2021, pp. 1–8.
[27] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. On Secur. Privacy*, 2016, pp. 582–597.
[28] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroSP)*, 2016, pp. 372–387.
[29] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
[30] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015, *arXiv:1506.06579*.
[31] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*.
[32] Y. Wang, J. Liu, X. Chang, R. J. Rodríguez, and J. Wang, "DI-AA: An interpretable white-box attack for fooling deep neural networks," *Inf. Sci.*, vol. 610, pp. 14–32, 2022.
[33] J. Lian, X. Wang, Y. Su, M. Ma, and S. Mei, "CBA: Contextual background attack against optical aerial detection in the physical world," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5606616.
[34] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2137–2146.
[35] N. Li and Z. Chen, "Toward visual distortion in black-box attacks," *IEEE Trans. Image Process.*, vol. 30, pp. 6156–6167, 2021.
[36] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on GAN," in *Proc. Int. Conf. Data Mining Big Data*, 2022, pp. 409–423.
[37] H. Liang, E. He, Y. Zhao, Z. Jia, and H. Li, "Adversarial attack and defense: A survey," *Electronics*, vol. 11, no. 8, pp. 1–19, 2022.
[38] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.
[39] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
[40] Y. Bai, Y. Wang, Y. Zeng, Y. Jiang, and S.-T. Xia, "Query efficient black-box adversarial attack on deep neural networks," *Pattern Recognit.*, vol. 133, 2023, Art. no. 109037.
[41] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.

[42] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016, *arXiv:1611.02770*.
[43] Y. Shi, S. Wang, and Y. Han, "Curls & whey: Boosting black-box adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6512–6520.
[44] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," 2017, *arXiv:1712.04248*.
[45] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," 2018, *arXiv:1807.04457*.
[46] H. Li, X. Xu, X. Zhang, S. Yang, and B. Li, "QEBA: Query-efficient boundary-based blackbox attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1218–1227.
[47] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *J. Comput. Graph. Statist.*, vol. 10, no. 1, pp. 1–50, 2001.
[48] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
[49] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.
[50] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6022–6031.
[51] E. Harris, A. Marcu, M. Painter, M. Niranjan, A. Prügel-Bennett, and J. Hare, "FMIX: Enhancing mixed sample data augmentation," 2020, *arXiv:2002.12047*.
[52] I. Goodfellow et al., "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 1–9, 2014.
[53] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," 2018, *arXiv:1805.09501*.
[54] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOV4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[55] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "AugMix: A simple data processing method to improve robustness and uncertainty," in *Proc. Int. Conf. Learn. Representations*, 2020.
[56] Y. Dong et al., "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.
[57] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[60] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
[61] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE/CVF IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10425–10433.
[62] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.
[63] 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.

Rui Zhu received the bachelor of engineering degree in electrical and information engineering in 2022 from Xi'an Air Force Engineering University, Xi'an, China, where he is currently working toward the master's degree in information and communication engineering with the Graduate School.

His research interests include remote sensing image processing, adversarial attack, and machine learning.

Shiping Ma received the Ph.D. degree in information and communication engineering from the Air Force Engineering University, Xi'an, China, in 2004.

He is currently a Professor with the College of Aeronautical Engineering, Air Force Engineering University. His research interests include image processing and computer vision.

Dr. Ma was a recipient of the Second Prize of Science and Technology Progress in Shaanxi Province.

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

His research interests include trustworthy machine learning and remote sensing.

Linyuan He received the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2018.

He is currently an Associate Professor with the School of Aeronautical Engineering, Air Force Engineering University, Xi'an, China. His research interests include image processing 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 currently a Professor with the School of Electronics and Information, Northwestern Polytechnical University. From 2007 to 2008, he was a Visiting Student with the University of Sydney, Camperdown, NSW, Australia. His research interests include hyperspectral remotesensing image processing and applications, intelligent signal and information acquisition and processing, video processing, and pattern recognition.

Dr. Mei was the recipient of the First prize of Natural Science Award of Shaanxi Province in 2022, 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) in 2017, the Best Reviewer of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS) in 2019, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS) in 2022. He is an Associate Editor for IEEE TGRS and IEEE JSTARS, Guest Editor for Remote Sensing, and the Reviewer for more than 30 international famous academic journals.