Local Aggregative Attack on SAR Image Classification Models

Meng Du ✉, Da-ping Bi, Ming-yang Du, Zi-long Wu, and Xin-song Xu
College of Electronic Engineering, National University of Defense Technology, Hefei, People's Republic of China
✉**Correspondence**
Meng Du, College of Electronic Engineering, National University of Defense Technology, Hefei 230031, People's Republic of China.
Email:dumeng_nudt@163.com

Convolutional neural networks (CNN) have been widely used in the field of synthetic aperture radar (SAR) image classification for their high classification accuracy. However, because CNNs learn a fairly discontinuous input-output mapping, they are vulnerable to adversarial examples. Unlike most existing attack manners that fool CNN models with complex global perturbations, this study provides an idea for generating more dexterous adversarial perturbations. It demonstrates that minor local perturbations are also effective for attacking. We propose a new attack method called local aggregative attack (LAA), which is a black-box method based on probability label information, to reduce the range and amplitude of adversarial perturbations. Our attack introduces the differential evolution (DE) algorithm to search for the optimal perturbations and applies the maximum between-class variance method (OTSU algorithm) to accomplish pixel-level labelling of the target and background areas, enabling attackers to generate adversarial examples of SAR images (AESIs) by adding small-scale perturbations to specific areas. Meanwhile, the structural dissimilarity (DSSIM) metric optimises the cost function to limit image distortion and improve attack stealthiness. Experiments show that our method achieves a high attack success rate against these CNN-based classifiers, and the generated AESIs are equipped with non-negligible transferability between different models.

Introduction: Owing to the fantastic feature extraction ability of convolutional neural networks (CNN), CNN-based SAR image classification models achieve higher precision classification than traditional supervised classification methods [1]. However, when there is a distribution drift between the training and test data, these classifiers are severely disturbed; in other words, a subtle adversarial perturbation may result in significant prediction errors [2]. Adversarial vulnerabilities make CNN models face enormous security challenges in real-world deployment.

In recent years, the proposed adversarial attack methods can be mainly divided into two categories: white-box attacks and black-box attacks. In the white-box condition, attackers can obtain the internal information of the victim models to generate adversarial examples, such as gradient-based attacks [2, 3], and saliency map-based attacks [4], etc. On the contrary, attackers cannot access the inside of models in a black-box scenario, so they can only finish attacks by querying the models' outputs, or by the transferability of adversarial examples, such as probability label-based attacks [5], decision-based attacks [6], transferability-based attacks [7], etc. Moreover, some studies have applied adversarial attacks to the real world. For example, the work in [8] successfully attacked Face++ (an advanced face recognition system) with just a pair of glasses. Thus, researching adversarial examples is necessary for both attackers and defenders.

Despite the fact that researchers have done a tremendous amount of work on adversarial examples of natural images, only a few studies have been done on AESIs. Li et al. [9] generated AESIs through fast gradient sign method (FGSM) [2] and its improved methods. Yin et al. [10] generated universal adversarial examples for remote sensing images. By adding global perturbations to the original images, the above approaches achieved high attack success rates but required large-scale and precise modifications to the SAR imaging results, rendering AESIs physically impractical. Some studies also proposed using local perturbations to fool classifiers. The work in [11] introduced U-Net to generate perturbations that cover only the target area. The sparse attack [12] focused on the sparse properties of SAR images, significantly reducing the number of perturbed pixels. Nevertheless, the current local perturbation still has the shortcomings of an extensive distribution range and a complex structure, and its physical feasibility needs further improvement.

In this study, we generate adversarial examples for CNN-based SAR image classifiers and explore the transferability [13] of AESIs between different models. To our knowledge, we are the first to fool CNNs by adding minor local adversarial perturbations to specified image regions. The main contributions are as follows.

1) A black-box attack method based on differential evolution (DE) [14] is proposed to generate adversarial examples in a scenario where the attacker can only access the models' probability label information. We call the proposed method local aggregative attack (LAA), which can fool the classifiers with small-scale simple perturbations and enhance the physical feasibility of AESIs.

2) We apply the maximum between-class variance method (OTSU) [15] to segment a SAR image into the target area and the background area. In this manner, attackers can place a high-stress level on the target area containing critical semantic information for image classification, which can improve the effectiveness, i.e., attack success rate, compared to other global perturbation methods.

3) The structural dissimilarity (DSSIM) [16] metric is added to form the cost function such that it balances the constraint relationship between attack effectiveness and stealthiness. Compared with traditional distance metrics, such as mean squared error (MSE) and peak signal-to-noise ratio (PSNR), DSSIM is an image quality assessment metric more in line with human visual perception, which can effectively limit image distortion and improve attack stealthiness.

Problem definition: We suppose that f is a well-trained k-class SAR image classifier and $I \in [0,255]^{w \times h}$ is an original input image. The reason we don't consider image channels here is that SAR images are usually 8-bit gray-scale images, so the number of channels defaults to 1. The output of f is a k-dimensional confidence vector denoted as $f(I)=[f_1(I), f_2(I), \cdots, f_k(I)]$, where each element $f_i(I)$ is the confidence of I belongs to class i. Let $C_p = \arg\max_i [f_i(I)]$, C_t represent the predicted and true classes of I. The ultimate goal of the attacker is to find an adversarial example I_{adv} formed by original image I and adversarial perturbation δ such that $C_t \neq C_p$, i.e., reducing the confidence level of C_t. It can be described as the following optimisation problem:

$$\min_{I_{adv}} f_{C_t}(I_{adv}), \qquad \text{s.t. } \|I_{adv} - I\|_F \leq M \tag{1}$$

where $f_{C_t}(I_{adv})$ is the confidence of adversarial example belongs to the true class, the difference between I_{adv} and I is constrained by the Frobenius norm (F-norm), and M controls the limitation of the maximum modification.

Methodology: In this section, we detail our method in three parts, and the framework of our algorithm is given at the end of this section.

Proposed method: A novel function $G(x, y, r, a)$ is designed to generate the adversarial perturbation δ, which takes as input a centre coordinate (x, y), a radius r and an amplitude a. The output of G is the adversarial perturbation δ as follows:

$$\begin{aligned} G(x, y, r, a)^{(m,n)} &= \begin{cases} a & \text{if } \sqrt{(m-x)^2 + (n-y)^2} \leq r \\ 0 & \text{otherwise} \end{cases} \\ &= \delta^{(m,n)} \qquad (\delta \in \mathbb{R}^{w \times h}) \end{aligned} \tag{2}$$

where $\delta \in \mathbb{R}^{w \times h}$ has the same size as I, (m, n) is the coordinate of the pixel in δ. In brief, the function G is to make the pixels in the range of r around the perturbation centre (x, y) equal to the same amplitude a, while the other pixels in δ are 0. To guarantee that I_{adv} is still an 8-bit grey-scale image, we need a clipping operation to restrict its grey-scale values to the interval of [0,255], as follows:

$$I_{adv} = Clip_{[0,255]}(I + \delta) \tag{3}$$

Obviously, the objective function value $f_{C_t}[Clip(I+\delta)]$ in equation (1) (i.e. the confidence level of the true class) is inversely proportional to r and a. Although a perturbation with a wider scale and larger amplitude can attack the model more easily, the blind pursuit of attack effects inevitably leads to excessive distortion of adversarial examples and damage the stealthiness of the attack. Therefore, a distance metric, called DSSIM, is introduced to limit image distortion caused by excessive perturbation, which is an objective image quality assessment metric that measures the dissimilarity of two images, as follows:

$$DSSIM(u,v) = \frac{1 - SSIM(u,v)}{2} \tag{4}$$

We can see that DSSIM is a sub-metric derived from structural similarity (SSIM). In contrast, the latter is an image quality assessment metric to measure the similarity of two images and can be calculated by the following equation:

$$SSIM(u,v) = \frac{(2\mu_u\mu_v + C_1)(2\sigma_{uv} + C_2)}{(\mu_u^2 + \mu_v^2 + C_1)(\sigma_u^2 + \sigma_v^2 + C_2)} \tag{5}$$

where μ_u, μ_v and σ_u, σ_v are the mean and standard deviation of the corresponding image pixel values, σ_{uv} is the covariance of two comparison images, C_1 and C_2 are the constants used to keep the metric stable. SSIM ranges from -1 to 1, while DSSIM ranges from 0 to 1. It means two images are identical when SSIM is equal to 1 or DSSIM is equal to 0. For more details, please refer to the literature [16].

Hence, we can balance the attack effectiveness and stealthiness by applying a linear weighted sum method to assign them corresponding weighting coefficients. Then, the problem shown in equation (1) can be transformed into a multi-objective optimisation problem, and the new objective function is as follows:

$$L(\delta) = (1-\omega)\cdot f_{C_t}[Clip(I+\delta)] + \omega \cdot DSSIM[I, Clip(I+\delta)] \tag{6}$$

where $\omega \in [0,1]$ is a constant that measures the relative importance of effectiveness and stealthiness.

Pixel-level labelling of SAR images: A SAR image usually consists of background clutter and a target. The detected objects in the target area have strong radar echoes and appear as discrete irregular bright spots. On the other hand, the background clutter area is primarily represented by low brightness dark spots because most of its echoes are redundant noise. There is a huge grey-scale difference between these two areas, and the features that strongly impact the prediction results are mainly concentrated in the target area [17]. Thus, focusing the adversarial perturbation on the target area strengthens the control over the perturbation location and improves our attack success rate.

We introduce the OTSU method [15] to extract target areas from SAR images, which is an unsupervised method of automatic threshold selection for image segmentation. OTSU is considered the best threshold selection algorithm for image segmentation and has been widely used in digital image processing for its simplicity and stability. The threshold selection principle of OTSU is to search for the optimal threshold $th_{opt} \in \mathbb{R}$, which can maximise the between-class variance of pixel segmentation results. The search process can be simplified to the following optimisation problem:

$$th_{opt} = \arg\min_{th\in\mathbb{R}} [p_t \cdot p_b \cdot (m_t - m_b)^2] \tag{7}$$

where $\mathbb{R}$ here is the closed interval of [0,255], p_t and p_b represent the probability of a pixel being classified into target and background by threshold th (greater than th and less than th, or vice versa), m_t and m_b represent the average grey-scale value of the target and background areas.

Finally, if the output of OTSU is O, an image size matrix contains only two elements, which label the target area as 1 and the background area as 0. In this way, the attacker can select the perturbed search area as desired.

Optimisation algorithm: After determining the cost function $L(\delta)$ and the search range of perturbation parameters, we need to find the optimal solution δ^* from the solution space quickly and efficiently. Nowadays, there are many optimisation algorithms, such as particle swarm optimisation algorithm (PSO), artificial bee colony algorithm (ABC), and adaptive simulated annealing algorithm (ASA), etc. In this study, we use the differential evolution (DE) algorithm [14], a population-based adaptive global optimisation algorithm, to search for δ^*. The specific optimisation process is as follows:

1) We first transform the perturbation δ into a candidate solution s holding four perturbation parameters (x, y, r, a) and initialize differential evolution by creating the population $S_0=\{s_1, \ldots, s_N\}$. N denotes the population size, which is set to be 80 here.

2) Next, for each parent $s_{i,g} \in S_g$, we generate a corresponding child $s_{i,g+1}$ by the following formula:

$$s_{i,g+1} = s_{r_1,g} + F \cdot (s_{r_2,g} - s_{r_3,g}) \tag{8}$$

where g is the current generation index and $r_1 \neq r_2 \neq r_3$ are the random indexes of candidate solutions. F is the mutation parameter set to be 0.5.

3) The last step is to evaluate the fitness of $s_{i,g+1}$ and $s_{i,g}$ by the cost function L. Regard the candidate solution with a smaller cost function value as the winner and keep it to the next generation.

We set the maximum number of generations to 100, and the optimisation stops early if there is a candidate solution that can make $C_t \neq C_p$. To speed up convergence, our method ignores the crossover process of the DE algorithm.

In summary, our attack is a probability label-based black-box method that optimises adversarial perturbations by repeatedly querying the confidence of the correct class until the attack is successful. The algorithm framework is shown in Figure 1. First, the coordinate set of the target area $\{(x,y) \mid O(x,y) = 1\}$ is extracted from the SAR image by OTSU, while the other two parameters r and a are set by the attacker according to their requirements. Next, DE optimises the cost function with the initial population $\{(x, y, r, a)\}$ as input and returns the optimal solution s^* which is transformed into the optimal perturbation δ^* by G later. Finally, we add δ^* to the original SAR image to get AESI.

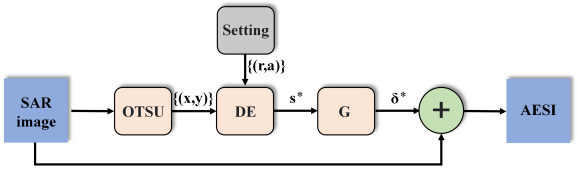

Fig 1 *Framework of the proposed algorithm (LAA)*

Experiments: In the experiments, we test our attack on five CNN-based SAR image classifiers and compare the performance of LAA with other algorithms, i.e., FGSM [2], and based iterative method (BIM) [3]. The results show that our attack is able to achieve a higher attack success rate with smaller and more controllable perturbations.

Dataset and implementation details: We use MSTAR, a SAR image dataset published by the U.S. Defence Advanced Research Projects Agency. It is obtained by the high-resolution spotlight SAR and mainly contains SAR images of Soviet military vehicle targets with different attitude and elevation angles. There are two collection conditions in MSTAR (SOC, EOC), and we use ten ground target classes collected by SOC to finish the experiments. The training dataset contains 2747 images collected at an elevation angle of 17°, and the test dataset contains 2426 images collected at an elevation angle of 15°.

The victim models are based on five different CNNs: GoogLeNet, ResNet50, ResNeXt50, InceptionV3, and DenseNet121. We form the validation dataset by uniformly sampling 20% data from the training dataset, and centre-crop the image to 128 × 128 in the pre-processing stage. During the training phase, we set the learning rate to 0.001, epochs to 50, and batch size to 64. All codes are written in Pytorch and run on a PC equipped with an NVIDIA GeForce RTX 2060 Max-Q graphics card

and an AMD Ryzen 7 4800HS processor. The training results in Figure 2 show that these models are reliable because there is no overfitting or underfitting during the training.

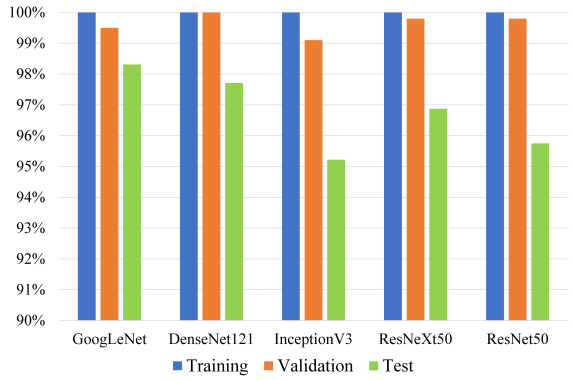

Fig 2 *Training results of five models*

Given that the algorithm's attack success rate is affected by model misclassification, we only generate adversarial examples for samples that can be correctly classified. When initializing the DE algorithm, we use uniform distribution $U(0,5)$ to generate r, $U(-255,255)$ to generate a and (x,y) is drawn randomly from the set $\{(x,y) \mid O(x,y)=1\}$. For FGSM, we set the perturbation amplitude ϵ to 0.01, while for BIM, we set the perturbation amplitude of each iteration α to 0.002 and the number of iterations N to 5.

Evaluation metrics: We use the attack success rate to evaluate the performance of the attack algorithm and the perturbation ratio to indicate the number of perturbed pixels, as follows:

$$\begin{cases} \text{attack success rate} = \frac{N_s}{N_t} \\ \text{perturbation ratio} = \frac{P_\delta}{P_I} \end{cases} \tag{9}$$

where N_s is the number of AESIs that satisfy $C_t \neq C_p$, N_t is the total number of AESIs generated by the attack algorithm against the victim model, P_δ denotes the number of non-zero pixels in δ and P_I is the number of pixels in I.

Experimental results: The results of adversarial attacks are shown in Table 1. First, the attack success rate of LAA is higher than other algorithms, while FGSM has the lowest attack success rate. It implies that performing specific perturbation on informative features is more effective than global perturbation, and multi-step perturbation works better than its one-step counterparts. Second, compared to global perturbation, our attack significantly reduces the number of perturbed pixels. In detail, for a 128×128 image, we can complete the attack just by perturbing dozens of pixels. Third, LAA performs better on the DSSIM metric. This is primarily due to the fact that LAA focuses the perturbation on the target area, improving the stealthiness of our attack. Finally, FGSM and BIM are white-box attacks that require internal gradient information of the victim model to complete the attack. In contrast, our method is a black-box attack that generates adversarial examples by querying the model's outputs, which significantly reduces the requirement for prior information. Overall, LAA performs better in terms of attack success rate, perturbation stealthiness, and prior information.

Figure 3 shows the results of an adversarial attack against GoogLeNet. Before the attack, the original image could be correctly classified as BMP-2 by the model with a confidence level of 99.60%. After adding the adversarial perturbation, the model classifies the AESI generated by FGSM, BIM, and LAA as T-62 and T-72 with 94.42%, 99.40% and 98.51% confidence levels, while the confidence levels of BMP-2 drop to less than 1%. It follows that the adversarial attack seriously impacts the CNN-based SAR image classifiers. Meanwhile, compared to FGSM and BIM, the adversarial perturbation generated by LAA has the smallest scale and simplest structure. Furthermore, our method precisely limits the perturbation to the target area, making it easier to control the attack behaviour.

Table 1. Attack performance comparison of three algorithms on five CNNS

Model	Success rate	DSSIM (mean)	Perturbation ratio (mean)	Algorithm
GoogLeNet	**92.32%**	**0.00281**	**0.34%**	**LAA**
	77.61%	0.00478	100.00%	FGSM
	84.28%	0.00354	100.00%	BIM
DenseNet121	**88.17%**	**0.00275**	**0.32%**	**LAA**
	81.48%	0.00474	100.00%	FGSM
	87.22%	0.00341	100.00%	BIM
InceptionV3	**90.68%**	**0.00246**	**0.32%**	**LAA**
	80.30%	0.00482	100.00%	FGSM
	87.75%	0.00327	100.00%	BIM
ResNeXt50	**86.97%**	**0.00262**	**0.30%**	**LAA**
	76.09%	0.00484	100.00%	FGSM
	83.19%	0.00316	100.00%	BIM
ResNet50	**89.61%**	**0.00272**	**0.30%**	**LAA**
	65.09%	0.00480	100.00%	FGSM
	73.31%	0.00341	100.00%	BIM

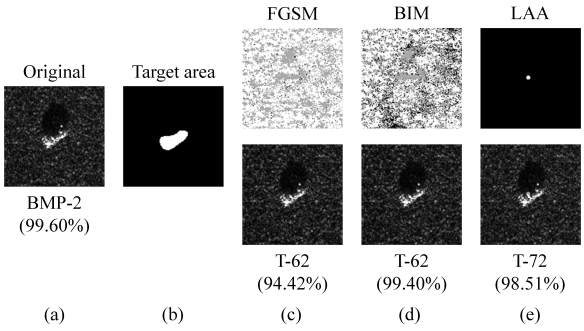

Fig 3 *(a) Original SAR image. (b) OTSU-labelled target area (the bright area is target and the dark area is background). (c), (d), (e) are the adversarial perturbations (top) and AESIs (bottom) generated by three algorithms.*

The experimental results in Table 2 and Figure 3 are obtained with the weight coefficient ω equals 0.6. Next, we analysed the sensitivity of adversarial attack results to ω. To simplify the experiments, we set ω to seven different values: 0.9, 0.75, 0.6, 0.45, 0.3, 0.15, 0, and only generated AESIs for the mini-dataset, which contains 238 images uniformly sampled from the original test dataset (sampling rate is 10%). Figure 4 shows that decreasing the value of ω can improve the attack success rate, but the value of DSSIM also increases, resulting in more image distortion.

Transferability of AESIs: In some extreme cases, the attacker can't even obtain the output information of victim models, which makes most attack algorithms ineffective. Meanwhile, some researchers have found that adversarial examples are transferable between different models [13], which means that adversarial examples generated for surrogate models can also be used to attack unknown victim models. Thus, the zero-access attack can be achieved by utilising the transferability of adversarial examples.

To explore the transferability of AESIs generated by our method, we let five models be the surrogate and victim models in turn and made them attack each other. The attack results are shown in Table 2. We can find that, without any transferability-enhancement operations, the transfer rates of AESIs generated by LAA on five surrogate models are over 35%, and the highest is close to 70%. Therefore, the transferability-enhancement methods will be the focus of our subsequent research.

Conclusion: This study demonstrates that minor local adversarial perturbations can also effectively fool deep learning models and are more physical feasibility than common global perturbations. We designed a

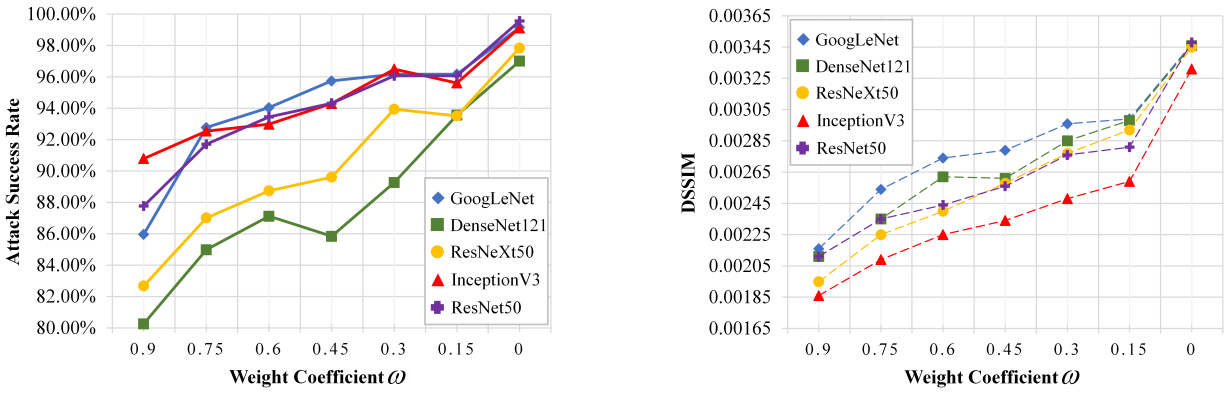

Fig 4 *Attack success rate and DSSIM as the weight coefficient ω changes. (a) Attack success rate. (b) DSSIM.*

Table 2. Transferability of AESIs

Vic \ Sur	GgN	DN121	IncV3	RNX50	RN50
GgN	–	59.66%	41.23%	43.29%	39.30%
DN121	48.09%	–	34.21%	41.13%	38.43%
IncV3	**68.94%**	66.09%	–	58.01%	48.47%
RNX50	53.62%	54.94%	38.16%	–	48.03%
RN50	56.60%	53.22%	**35.96%**	61.04%	–
Mean	56.81%	58.48%	37.39%	50.87%	43.56%

new black-box attack called local aggregative attack (LAA) for a scenario where the attacker can only access the probability label information. OTSU was introduced to segment SAR images such that our method could generate adversarial perturbations on a smaller scale and concentrate more on the informative area in SAR images. Meanwhile, the cost function optimised by the DSSIM metric can trade off the effectiveness and stealthiness of AESIs. Experimental results show that LAA completes adversarial attacks without any internal information about the victim model and performs better in terms of attack success rate, perturbation stealthiness, and generalisation ability. Future work will focus on enhancing the transferability of AESIs to make the adversarial attack more generalisable across different models.

Acknowledgments: This work was supported by the National Natural Science Foundation of China under Grant 62071479.

Received: DD MMMM YYYY *Accepted:* DD MMMM YYYY
doi: 10.1049/ell.10001

References

1. Li, J., et al.: Classification of very high resolution sar image based on convolutional neural network. In: 2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP), pp. 1–4. IEEE (2017)
2. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
3. Kurakin, A., Goodfellow, I.J., Bengio, S.. Adversarial examples in the physical world. In: Artificial intelligence safety and security, pp. 99–112. Chapman and Hall/CRC (2018).
4. Papernot, N., et al.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS&P), pp. 372–387. IEEE (2016)
5. Chen, P.Y., et al.: Zoo: Zeroth order optimisation based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM workshop on artificial intelligence and security, pp. 15–26. (2017)
6. Chen, J., Jordan, M.I., Wainwright, M.J.: Hopskipjumpattack: A query-efficient decision-based attack. In: 2020 ieee symposium on security and privacy (sp), pp. 1277–1294. IEEE (2020)
7. Xie, C., et al.: Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2730–2739. (2019)
8. Sharif, M., et al.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 acm sigsac conference on computer and communications security, pp. 1528–1540. (2016)
9. Li, H., et al.: Adversarial examples for cnn-based sar image classification: An experience study. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14, 1333–1347 (2020)
10. Yin, Z., et al.: Universal adversarial perturbation for remote sensing images. arXiv preprint arXiv:2202.10693 (2022)
11. Du, C., et al.: Fast c&w: A fast adversarial attack algorithm to fool sar target recognition with deep convolutional neural networks. IEEE Geoscience and Remote Sensing Letters 19, 1–5 (2021)
12. Junfan, Z., et al.: Sparse adversarial attack of sar image. Journal of Signal Processing 37(9), 11 (2021)
13. Papernot, N., et al.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp. 506–519. (2017)
14. Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. IEEE transactions on evolutionary computation 15(1), 4–31 (2010)
15. Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics 9(1), 62–66 (1979)
16. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, vol. 2, pp. 1398–1402. Ieee (2003)
17. Qian, G., Haipeng, W., Feng, X.: Research progress on aircraft detection and recognition in sar imagery. Journal of Signal Processing 9(3), 497–513 (2020)