FAWA: Fast Adversarial Watermark Attack

Hao Jiang, Jintao Yang, Guang Hua*, Lixia Li, Ying Wang, Shenghui Tu, and Song Xia

Abstract—Recently, adversarial attacks have shown to lead the state-of-the-art deep neural networks (DNNs) to misclassification. However, most adversarial attacks are generated according to whether they are perceptual to human visual system, measured by geometric metrics such as the ℓ_2-norm, which ignores the common watermarks in cyber-physical systems. In this paper, we propose a fast adversarial watermark attack (FAWA) method based on fast differential evolution technique, which optimally superimposes a watermark on an image to fool DNNs. We also attempt to explain the reason why the attack is successful and propose two hypotheses on the vulnerability of DNN classifiers and the influence of the watermark attack on higher-layer features extraction respectively. In addition, we propose two countermeasure methods against FAWA based on random rotation and median filtering respectively. Experimental results show that our method achieves 41.3% success rate in fooling VGG-16 and have good transferability. Our approach is also shown to be effective in deceiving deep learning as a service (DLaaS) systems as well as the physical world. The proposed FAWA, hypotheses, and the countermeasure methods, provide a timely help for DNN designers to gain some knowledge of model vulnerability while designing DNN classifiers and related DLaaS applications.

Index Terms—Adversarial attacks, watermark, differential evolution, DLaaS security

1 Introduction

Recently, deep learning has achieved great success across a wide range of fields of modern human society, bringing revolutionary performance improvements from algorithm, software, and hardware levels [1], [2]. To promote and popularize the use of the powerful deep learning technology, IBM proposed the concept of deep learning as a service (DLaaS) and its Watson platform provides a supporting software environment for deep learning [3]. On this basis, Aliyun and Amazon have also provided accessible APIs for developers and others to quickly deploy deep learning applications [4]. There have also been other companies such as Intel and Cambricon which have introduced deep learning hardware acceleration devices [5]. However, the adversarial attack technology that has emerged in recent years has posed a great challenge to the security and robustness of deep learning systems, which also restricts the further development of DLaaS and other related applications [6]. Adversarial attack [7], [8] is a kind of modification on the original clean samples, adding carefully designed perturbations, so that the generated adversarial samples can fool the deep neural network (DNN) classifiers. Taking a computer vision classifier as an example, a clean pig image that can be correctly classified by the classifier is input to the same classifier after adding some adversarial perturbations, and the output decision could become "airplane". Many advanced DNNs have been shown to be vulnerable to adversarial attacks. However, one of the evaluation criteria for most adversarial attacks is not easily detectable by human visual system, and the way to measure them is some geometric measurement, such as the ℓ_2-norm. Representative designs of adversarial attacks under ℓ_2-norm include fast gradient sign method (FGSM) [7], basic iterative method (BIM) [9], projected gradient descent (PGD) [10], and Carlini & Wagner attacks [11], and Adv-watermark attack [12]. Adv-watermark attack propose a basin hopping evolution(BHE) algorithm based on hopping evolution. When generating adversarial watermark attacks, BHE algorithm could gain the suitable transparency of the watermark image and the suitable position within the host image to embed watermark.

Watermarking is a digital media copyright protection technology, which encodes visible or invisible digital signals on multimedia content to achieve the effects of image marking, copyright enforcement, and forgery protection [13]. As an important branch of information hiding, digital watermarking technology provides a compromising method for digital multimedia copyright protection, and has been widely studied and applied by researchers and practitioners. According to the visual effects, image watermarks can be categorized into visible watermarks and invisible watermarks [14]. Visible watermarks, which are usually logos or trademarks, are easy to implement, and they have been shown to be robust against a series of attacks including post processing, re-compression, and geometric transforms. Different from visible watermarking, invisible watermarking is a technology that uses data redundancy. The embedded watermarks, either in spatial or in transform domain, could usually bypass human visual systems. Compared with visible watermarks, invisible watermarks are difficult to perceive directly, so it is difficult to make

- *This work was supported in part by the National Natural Science Foundation of China Enterprise Innovation Development Key Project under Grant U19B2004, the Special Fund for Science and Technology of Guangdong Province under Grant 2019SDR002, the Key Project of Earth Observation and Navigation under Grant 2017YFB0504100, the Equipment Project of the Equipment Development Department under Grant 41412010702, the Major R & D Platform Project of New R & D Institutions in Zhongshan City under Grant 2017F1FC0001, the Special Innovation Project of High-end Scientific Research Institutions in Zhongshan City under Grant 181129112748101.*
- *H. Jiang, J. Yang, G. Hua, S. Tu, and S. Xia are with the School of Electronic Information, Wuhan University, Wuhan 430079, China. L. Li and Y. Wang are with Wuhan Digital Engineering Institute, Wuhan 430205, China.*

Manuscript received ...
**Corresponding author: Guang Hua (E-mail: ghua@whu.edu.cn)*

targeted attacks and modifications to watermark content. Therefore, invisible watermarks are more secure and have more extensive application scenarios.

The patch attack method [15] which could be considered as a special type of visible image watermark, is to generate an interference image of a specific size and paste it on a clean image to achieve the effect of attacking DNNs. In the automatic check-out link of an automatic retailing system, a small sticker similar to a product trademark can seriously affect the computer vision recognition system, causing it to recognize expensive products as very cheap items [16]. In addition, in the field of autonomous driving, by putting adversarial patches on traffic signs, the target detector in the driving system can misrecognize the stop sign as the speed limit of 45km/h, causing the autonomous driving system to make wrong decisions [17].

Adversarial training [7] is currently one of the most effective defense methods against adversarial attacks. This method attempts to improve the robustness of neural networks by training with adversarial samples. It is also the only solution to deal with adversarial watermark attacks. [12]. However, the cost of adversarial training is higher, because the method needs to generate a large number of adversarial attack samples and add these samples to the training set to retrain the network. Moreover, It is difficult to generalize the knowledge obtained via adversarial training, making the trained networks still vulnerable to unseen adversarial attacks.

In this paper, we propose a fast adversarial watermark attack (FAWA) method, which is similar to the patch attack method. It attaches a translucent watermark to a clean image and adjusts the size and rotation to attack the deep learning classifier, resulting in incorrect output. The embedding strength of the patch is control to make it hard to recognize by visual system. We propose a fast differential evolution [18] algorithm to generate such attack samples, aiming to find a global optimal solution to complete the task of attacking the classifier. Our method is a black-box attack method, which only needs to obtain the category and confidence of the classifier output to implement the attack. Using this method, we can generate effective attack samples that can easily avoid existing adversarial attack evaluation criteria. In addition, we propose two countermeasures to withstand the proposed adversarial attack. The first method is based on random center rotation. It leverages the fact that trained DNN classifiers are usually insensitive to rotated objects, while the adversarial patch is sensitive to rotation. The second countermeasure method is based on median filtering [19]. We show that by using a 3×3 median filter, the image generated by FAWA could be correctly classified.

To the best of our knowledge, the proposed attack is the first adversarial watermarking attack against the DLaaS and real physical world. We successfully implemented the attack against Aliyun Image Recognition APIs; in addition, we printed watermark sticker and also successfully implemented the attack on photos taken by the camera in real time. Fig. 1 shows that by applying our watermark attack with an opacity of 0.2, the Aliyun Image Recognition APIs misclassify a killer whale image as a sea lion image. Moreover, we propose two adversarial defense methods to effectively defend against FAWA. Different from Adv-watermark's BHE algorithm [12], we propose a fast differential evolution method to generate watermark attacks. The one-pixel attack [20] is also based on the differential evolution method, but it only modifies a single pixel of the image to complete the attack. Our method is to complete the attack by adjusting the transparency, size, and rotation direction of the watermark, while the Adv-watermark method does not consider rotation. In terms of patch attacks, what we generate is a translucent patch, which is different from the existing patch generation style.

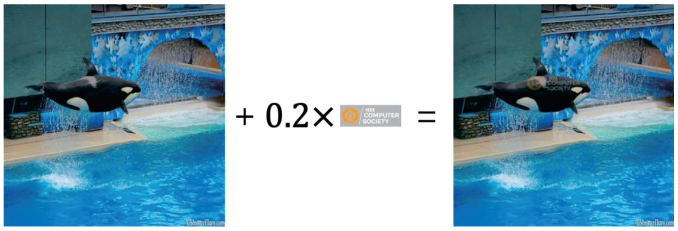


Fig. 1. A demonstration of fast adversarial watermark attack by adding a watermark with an opacity of 0.2 to clean image. The Aliyun Image Recognition APIs misclassify the attacked image as a sea lion image with a high confidence ratio.

For the countermeasure methods, the proposed first one is based on the method of random center rotation, which is not used in the previous adversarial defense methods. The other proposed defense method is based on median filtering, which is not used in defense against adversarial watermark attack yet. In addition, the existing patch confrontation defense methods mainly adopt adversarial training methods and training modes based on ablation methods [21], which are also different from our defense methods. Our defense methods don't requiring retraining or fine-tuning the original model, which are different from adversarial training methods [7].

Our attack and defense methods, analyze the destructive effects of watermark attacks on DNNs, so that developers can consider more comprehensively and thoughtfully when developing DLaaS applications.

In general, the main contributions of this work include:

1) A FAWA method is proposed. To the best of our knowledge, it is the first known adversarial watermark method to successfully attack DLaaS (Aliyun Image Recognition APIs [22]) and real physical world.
2) We provide a theoretical analysis of the vulnerability of a DNN classifier as a function of feature space and number of classes. A closed-form solution is derived for the 1-D case. Experimental results are provided to verify higher dimensional cases.
3) We further propose two countermeasure methods, based on random center rotation and median filtering respectively, which can effectively improve the defense ability of any model against adversarial watermark attacks with a slight performance drop in terms classification accuracy. Our defense methods are easy to deploy, without requiring retraining or fine-tuning the original model.

The remainder of the article is organized as follows. Section 2 provides background information on adversarial attacks & defenses, and watermark embedding. Section 3 introduces the methodologies of FAWA, algorithm flow, and the hypotheses of FAWA's attack mechanism. Section 4 introduces two potential countermeasures against FAWA. Section 5 presents experimental results of attacking DLaaS and real physical world by FAWA, validation of attack hypotheses, and countermeasures. Finally, Section 6 concludes the article.

2 Related Works

2.1 Adversarial Attacks & Defenses

2.1.1 Adversarial Attacks

Adversarial attacks in DNNs were first discovered by Christian Szegedy et al [23]. Then Ian Goodfellow et al. [7] proposed an FGSM method to quickly and effectively generate adversarial samples. Carnili and Wagner [11] proposed a powerful attack method, which measures the distortion of adversarial examples in ℓ_1-, ℓ_2-, and ℓ_∞-norm. Eric Wong [24] proposed a Wasserstein adversarial sample generation method, which has good semantic characteristics in generating adversarial samples based on MNIST dataset. The adversarial perturbations generated is mostly concentrated in handwritten digits, and there is almost no modification on the background of the image. However, this method has no significant semantic characteristics when generating adversarial attacks based on the CIFAR-10 dataset. Amin Ghiasi et al. [25] proposed a semantic-based attack method for certification defense. This method can attack the certification defense method by generating a small amount of perturbations with semantic information. In addition, Su et al. [20] proposed a single-pixel attack method. Their method only modifies a single pixel of the sample through differential evolution to complete the generation of adversarial samples, which is a very powerful attack method.

2.1.2 Patch Attacks

Brown et al. [15] first proposed a patch attack, and the attack on the VGG-16 [26] network in the physical world was completed by putting a sticker on the desktop, which caused VGG-16 to misclassify bananas as a toaster. Simen Thys et al. [27] proposed a novel anti-patch attack, in whcih the patch is applied to the center of the human body to deceive the human body detector. Unlike the above-mentioned digital world attack, the attack is modified by adding an unprintable score item to the loss function, so as to get a printable and effective patch attack sample in real physical world. Liu et al. [28] used a PS-GAN method to generate patch attacks that can work effectively in the digital and physical world, using the attention mechanism to place adversarial patches into the most sensitive areas for classification, and the generated patches posted on street signs can effectively deceive artificial intelligence (AI) autonomous driving system.

2.1.3 Adversarial Defenses

Ian Goodfellow et al. [7] proposed the concept of adversarial training to deal with adversarial attacks. Adding the generated adversarial attack samples to the training set and retraining the model can make models more robust and generalized. Papernot et al. [29] proposed an adversarial attack defense method based on defense distillation. By training a model to predict the possible output of another model trained earlier, a more robust model is obtained. Defensive distillation is actually a typical gradient masking method to defend against adversarial attacks. In fact, even if we don't know the true gradient, we can successfully attack by using an approximate gradient. Or the migration feature of the adversarial sample can also be used to break the defense model. For example, C&W [11] later successfully attacked the defense model. Reuben Feinman et al. [30] proposed an adversarial sample detection method based on Bayesian uncertainty estimation, which has the characteristics of attack independence and blind detection. Pingyeh Chiang [21] proposed an adversarial defense method against patch attacks and designed a fast training algorithm for it. The research also found that the defense has the characteristics of patch shape migration. Qiu et al. [31] proposed the FenceBox framework, which is equipped with three different types of 15 data enhancement methods to resist various adversarial attacks. And it seems to be the latest comprehensive framework.

2.2 Watermark embedding Methods

The transparency of the watermark image is usually referred to as invisibility. Gray-scale image watermarking needs to compare the gray values of the original image and the watermarked image, and obtain the mean square error, signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR) between them, which constitute the invisibility index of the watermarking system. However, the measurement of transparency of color images is more complicated than that of grayscale images. When the embedded channel of watermark information is different from the color space, the formula for calculating transparency will change accordingly.

In the field of visible watermark embedding, Braudaway et al. [14] first introduced visible watermarks to digital images. They use adaptive non-linear pixel domain technology to add a visible watermark to the image as a means to identify the ownership of the image, while not obscuring the details behind the image and making the visible watermark difficult to remove. In this algorithm, the watermark image and the host image have the same size. That is to say, the watermark image has established a one-to-one correspondence with each pixel of the host image, so the pixels in the watermark image can be divided into transparent pixels and non-transparent pixels. After the watermark is embedded, the pixel value in the host image corresponding to the transparent area of the watermark will remain unchanged, while the pixel corresponding to the non-transparent area of the watermark will be changed according to the corresponding watermark value and the adopted embedding model.

Kankanhalli and Ramakrishnan [32] used the statistics of block discrete cosine transform (DCT) coefficients to determine the watermark embedding coefficient of each block. They later extended it with the consideration of texture sensitivity in the human visual system to better maintain the perceptual quality of the image. Hu and Kwong [33]

implemented an adaptive visible watermark in the wavelet domain to deal with visual discontinuities that may be introduced by DCT-based methods. These methods improve the visual quality of visible watermarks, but also make them difficult to remove.

3 Fast Adversarial Watermark Attack

3.1 Perceivable Color Watermark Generation

Here we adopt the $alpha$-blending [34] method, a common image processing technique to produce transparent effects, to generate perceptible color watermarks.

Image and Bitmap objects store the color of each pixel according to 32 bits: 8 bits each for Red, Green, Blue, and Alpha.

In our method, let $I(M, N)$ be the original clean image, M and N respectively represent the height and width of the image. We further define a series of parameters to describe a processed version of the watermark. Let α be the $alpha$-blending transparency, β be the scaling coefficient imposed on the height and width of the original mark, and let γ be the rotation angle. For simplicity, we set $\gamma \in \{0°, 90°, 180° 270°\}$. Denote the original watermark as W_0, where m and n are the height and width of the original watermark. And denote the transformed watermark patch as W, where βm and βn are the height and width of the transformed watermark patch. Further denote the area to superimpose the watermark W as $D(j, k)$, where j and k are the horizontal and vertical coordinates of the center of D. Then the embedding process of adversarial watermark is shown in Fig. 2. The sizes of the original clean image, the original watermark, and the scaled watermark are respectively marked on the right side of Fig. 2. We constrain the edge of the watermark to not exceed the edge of the original clean image. Therefore, the moving range of the center of D is within the blue rectangular box in Fig. 2, so that j, k satisfies the constraints $\frac{\beta n}{2} < j < N - \frac{\beta n}{2}$, $\frac{\beta m}{2} < k < M - \frac{\beta m}{2}$.

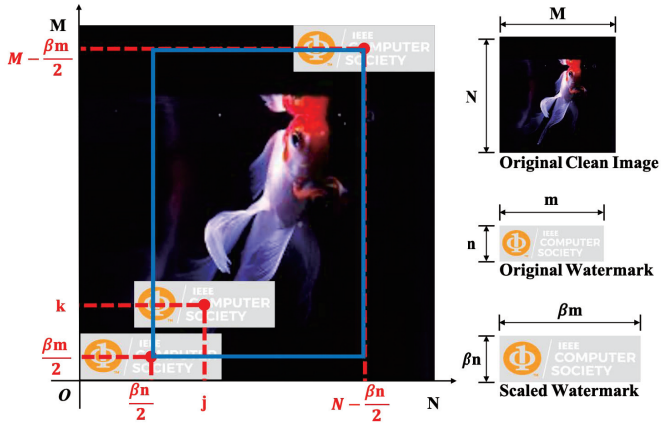


Fig. 2. Schematic diagram of adversarial watermark embedding.

Then the watermarked image $I_{new}(M, N)$ is expressed as:

$$I_{new}(M, N) = I(M, N) - \alpha D(j, k) + \alpha W. \quad (1)$$

We further have the following constraints $0 < \beta < 1, max(\beta m, \beta n) < min(M, N)$, which means that the maximum value of transformed watermark patch's height and width must less than the minimum value of the host image's height and width, and $0 < \alpha < T$, where T controls upper limit of the perceptibility.

3.2 Problem Formulation

Our goal is to generate adversarial perturbations with perceptible watermarks to achieve adversarial attacks that are difficult to be detected as maliciously modified images by the human visual system. The position, transparency, scaling and rotation angle of the watermark affect the generation of adversarial samples. This process can be formulated as a constrained optimization problem.

According to the previous subsection, we could denote the watermark attack process as $\mathcal{A}(I, W, j, k, \alpha, \beta, \gamma) : I(M, N) \rightarrow I_{new}(M, N)$. Suppose $I(m, n)$ belongs to class l under a given classifier with output probability P_l, then the attack aims at minimizing the probability:

$$\hat{\mathcal{A}} = \arg\min_{\mathcal{A}} P_l. \quad (2)$$

This problem involves several variables: 1) The center point (j, k) in the host image where embedding the watermark; 2) The transparency α of the watermark; 3) The scaling factor β of the watermark; 4) The rotation angle γ of the watermark. Without affecting the aesthetics of the image, this method embeds the adversarial watermark into the clean image and modifies the image information to a lesser extent. At the same time, the adversarial watermarking attack misleads the trained image classifier through this modification of the clean image.

When attacking real physical world, the unprintable score S needs to be considered. The physical meaning of S is the value of the pixels' color of the image cannot be printed by the printer as it is. For common printers,

$$S = \sum_{W_p \in W} \min_{K_p \in K} |W_p - K_p|, \quad (3)$$

where W_p is a pixel in the watermark W, and K_p is a color in a set of printable colors K. This loss is beneficial for the colors in the watermark to be very close to the colors in the set of printable colors. Therefore, the objective function in (2) could be updated to

$$\hat{\mathcal{A}} = \arg\min_{\mathcal{A}} (P_l + \lambda S), \quad (4)$$

where λ is the weight scaling factor of printable scores determined empirically.

3.3 Fast Differential Evolution

We propose a fast differential evolution method to solve the above optimization problem. The whole process of the proposed fast differential evolution method for the FAWA is presented in Algorithm 1. The host image and the original watermark image are the input of the alrogithm. The FAWA image is the output of the alrogithm. Fisrt of all, the algorithm initializes a certain size of population and the other variables. In this case, the generation number g equals 1. By using the dynamic correction of the factor, the convergence speed could be faster and the convergence accuracy could be higher. After that, the algorithm uses a

projection search variogram function to get variant individuals. For the individuals previously obtained, the algorithm use a soft crossover factor during the crossover operation, that is, the crossover factor changes dynamically with the operation process and performs dynamic changes after each individual's optimal solution is obtained. Then select the optimal solution of this generation. If the stop condition (the watermarked image could mislead the classifier) is not met, the algorithm enter the next generation (In this case, generation number $g = g + 1$) ; else, the FAWA image (the optimal solution) is finally generated as the output of the system.

Algorithm 1 Fast Differential Evolution Algorithm

Input: I, W
Output: I_{new}

Set the size of population $PS = 50$, See Section 3.3.1
Random initialize co and φ, See Section 3.3.2
Initialize the correction coefficient θ, See Section 3.3.4
Define population vector $V = (j, k, \alpha, \beta, \gamma)$
Set upper and lower bounds of variables $j, k, \alpha, \beta, \gamma$
Random initialize the solution of population $V(1)$
Get $I_{new}(1)$ *based on* $V(1)$
// g is the generation number
for $g = 1, ..., 100$ **do**
  // Dynamic Correction, See Section 3.3.4
  Update θ
  for $i = 1, ..., PS$ **do**
    // Projection Searching, See Section 3.3.1
    Get target individuals $tar_i(g)$ *according to equation 5*
    Process over-limit individuals according to equation 6
    // Crossover, See Section 3.3.2
    $\varphi = \varphi + 1$
    $CO = co \times \cos\left(\frac{\varphi \pi t}{2}\right)$
    Update individuals $tar_i(g)$ *according to equation 7*
  end for
  // Selection, See Section 3.3.3
  Select the optimal solution of this generation as $V(g+1)$
  Get $I_{new}(g+1)$ *based on* $V(g+1)$
  if $I_{new}(g+1)$ could mislead the classifier **then**
    $I_{new} = I_{new}(g+1)$
    goto final
  end if
end for
$I_{new} = I_{new}(101)$
final:
return I_{new}

In the following content, the entire process of the fast differential evolution method will be introduced in detail.

3.3.1 Projection Searching

The projection searching process is to perform a uniform search near a certain point to prevent the problem of local searching due to the limited number of iterations. In this process, the optimal value of the previous generation is set as the center, and the points to be searched are evenly distributed within a circle whose radius is the distance of each individual to the center. Once certain dimensions of some individuals exceed the searching range, they are projected to the nearest point within the searching radius.

Suppose $\mathbf{v} = (v_1, v_2, ..., v_D)$, $a_i \le v_i \le b_i$, $i = 1, 2, ..., D$, where b_i and a_i represent the upper and lower bounds of the i-th dimension variable, and D is the dimension of the problem. Then the corresponding searching variogram is defined as: in the searching process of each generation, let the target individual generated by the current individual $\mathbf{v}(u)$ be $\mathbf{tar}_i(u)$, where u represents the generation number, η represents the searching scale factor, r is used for dynamically changing the searching scale, then:

$$\mathbf{tar}_i(u) = \mathbf{v}(u) + \eta^r \mathbf{v}(u) randn(1, D), \tag{5}$$

where $randn(1, D)$ returns a $1 \times D$ vector in which the value of each dimension obeys the standard normal distribution. Set $r = \pm \bmod (i, 5)$ empirically. The initial value of $\mathbf{v}(u)$ is $\mathbf{v}(0) = rand(1, D)$, $i = 1, 2, ..., PS$, and PS represents the population size. $rand(1, D)$ returns a uniformly distributed random number greater than or equal to 1 and less than D.

In the process of variation, it is easy for the variation individuals to exceed the limit in a certain dimension, and this situation will also occur in the correction process. The processing method is as:

$$tar_{i,j}(u) = \begin{cases} v_j(u), & tar_{i,j}(u) < a_j \\ v_j(u), & tar_{i,j}(u) > b_j \\ tar_{i,j}(u), & otherwise \end{cases} \tag{6}$$

where $v_j(u)$ is the j-th dimension of the u-th generation solution of the variable, $tar_{i,j}(u)$ is the j-th dimension of the u-th generation variable's value of the target individual.

3.3.2 Crossover

In the crossover process, a larger crossover factor will reduce the memory ability of retaining the historical optimal solution, and a smaller crossover factor is not conducive to the search and update of the population. Therefore, our method adopts the following soft crossover factor, and uses the following rules to dynamically change its value:

$$tar_{i,j}(u) = \begin{cases} tar_{i,j}(u), & rand(0,1) < CO \\ v_j(u), & otherwise \end{cases} \tag{7}$$

where $CO = co \times \cos\left(\frac{\varphi \pi t}{2}\right)$, co represents the crossover coefficient of the crossover factor and CO is the crossover factor, $co \in [0, 1]$, $rand(0, 1)$ is the scaling factor of searching range, φ represents the frequency of the periodic function.

3.3.3 Selection

Suppose the objective function of optimization problem in Section 3.2 (which is $\hat{\mathcal{A}}$) as $h(\mathbf{v})$, and our goal is to minimize the objective function. First, the variation solutions obtained during the variation operation are substituted into the objective function to obtain the corresponding values. Solve the minimum value of the target function, then the corresponding solution is the optimal solution $\mathbf{o}(u)$ of the variogram in this generation,

$$\mathbf{o}(u) = h^{-1} \min_{i=1,2,...,PS} h(\mathbf{tar}_i(u)), \tag{8}$$

where h^{-1} is the inverse function of $h()$, and then substitute the obtained mutation optimal solution $\mathbf{o}(u)$ into the objective function, compare it with the optimal solution of

the previous generation, select the optimal solution of this generation, and enter the next generation:

$$\mathbf{v}(u+1) = \begin{cases} \mathbf{o}(u), & h(\mathbf{o}(u)) \leq h(\mathbf{v}(u)) \\ \mathbf{v}(u), & otherwise. \end{cases} \tag{9}$$

3.3.4 Dynamic Correction

Due to the limited accuracy of the solution produced in the variation process, it is necessary to dynamically correct the optimal solution. The idea is to perform variation, crossover and selection on the optimal solution in this iteration, the operation steps are as follows:

Step 1. Determine the coefficient θ, which is used to limit the floating range during the dynamic correction process. It can effectively increase the convergence speed after 0 to 20 times of polling processes. The operation function is: $\theta = \mathrm{mod}(i, 20), i = 1, 2, ..., PS$.

Step 2. Update $\mathbf{v}(u)$ via $\mathbf{v}(u) = \mathbf{v}(u) \times 2^{-\theta}$ to determine the maximum floating value. Then obtain the floating range via

$$\begin{aligned} &\mathbf{f}_i = (-1)^i \times (\mathbf{v}(u) + \mathbf{v}(u) \times rand(1, c)) \\ &i = 1, 2, ..., PS. \end{aligned} \tag{10}$$

After that, the variogram solution based on the optimal solution of the current generation is obtained, which is given by $\mathbf{tar}_i(u) = \mathbf{v}(u) + \mathbf{f}_i$.

Step 3. Crossover process (7), and the purpose of using the crossover process in the correction process is to selectively retain the memory solution while variation to achieve a faster and better convergence effect, making it faster and closer to the optimal solution.

Step 4. In the selection process, the variogram solutions of the correction function generated from **Step 1 to 3** are respectively substituted into the objective function, and solve the minimum value. Then compare the minimum value with the uncorrected optimal solution of this generation, select the optimal solution with a smaller objective function value, which is the optimal solution for this iteration.

3.4 Attack Mechanism Hypotheses

To provide more insights on the vulnerability of trained DNN classifiers, we propose the following two attack mechanism hypotheses.

3.4.1 Hypothesis 1: On Number of Classes and Feature Space Size

A trained DNN classifier is established upon a predefined feature space whose dimension and boundaries are implicitly fixed, and the trained weights together with nonlinearities within the network determine the decision boundaries within that given feature space. In this way, An effective adversarial watermark attack is in fact a modification of the sample feature, pushing the feature to cross its near decision boundary. Therefore, we hypothesize that the vulnerability of a trained DNN classifier is proportional to the size of the feature space and inversely proportional to the underlying number of classes.

To illustrate this, let us consider a 1-D case as an example before we generate the situation into arbitrary dimensional spaces. Let us define the following parameters: feature dimension $F_{dim} \in \mathbb{Z}_+$ (In 1-D case, $F_{dim} = 1$), sample feature descriptor $\mathbf{x} \in \mathbb{R}^N$, number of classes $N_{class} \in \mathbb{Z}_+$, feature space boundary $\|\mathbf{x}\|_2 \leq B$, where $B \in \mathbb{R}_+ < \infty$. Decision boundaries ξ_q, $q \in \{1, 2, \ldots, N_{class} - 1\}$. Without loss of generality, assume $-B < \xi_1 < \xi_2 < \cdots < \xi_{N_{class}-1} < B$. Vector $\mathbf{x}$ becomes scalar x. Decision rule:

$$x \in \text{Class } q \quad \text{if} \quad \xi_{q-1} \leq x < \xi_q, \tag{11}$$

where $\xi_0 \doteq -B$ and $\xi_{N_{class}} \doteq B$. If $x \in$ Class q, the condition for another sample $y = x + \delta$ to shift into other classes is

$$\delta < \xi_{q-1} - x \leq 0 \quad \text{or} \quad \delta \geq \xi_q - x > 0, \tag{12}$$

thus the absolute cost to make y classified into a different class than x is lower bounded by

$$|\delta| > \min\{x - \xi_{q-1}, \xi_q - x\}, \tag{13}$$

and we have the conditional expectation inequality

$$\begin{aligned} &E\left\{|\delta| \middle| x \in \text{Class } q\right\} \\ >&\min\left\{E\left\{x \middle| x \in \text{Class } q\right\} - \xi_{q-1}, \xi_q - E\left\{x \middle| x \in \text{Class } q\right\}\right\} \\ =&\frac{\xi_q - \xi_{q-1}}{2}. \end{aligned} \tag{14}$$

The overall expected minimum cost $\mathcal{C}$ to alter the class of an arbitrary sample x is then given by

$$\begin{aligned} \mathcal{C} &= \sum_{i=1}^{N_{class}} P\{x \in \text{Class } q\} E\left\{|\delta| \middle| x \in \text{Class } q\right\} \\ &= \sum_{i=1}^{N_{class}} \frac{\xi_q - \xi_{q-1}}{2B} E\left\{|\delta| \middle| x \in \text{Class } q\right\} \\ &> \sum_{i=1}^{N_{class}} \frac{\xi_q - \xi_{q-1}}{2B} \frac{\xi_q - \xi_{q-1}}{2} = \sum_{i=1}^{N_{class}} \frac{(\xi_q - \xi_{q-1})^2}{4B}. \end{aligned} \tag{15}$$

According to Cauchy-Schwarz inequality we have

$$\begin{aligned} \sum_{i=1}^{N_{class}} \frac{(\xi_q - \xi_{q-1})^2}{4B} &\geq \sum_{i=1}^{N_{class}} \frac{4B^2}{N_{class}^2 B} \\ &= \frac{4N_{class}B^2}{4N_{class}^2 B} = \frac{B}{N_{class}}, \end{aligned} \tag{16}$$

where the equality holds only if $\forall q \in \{1, 2, \ldots, N_{class}\}, \xi_q - \xi_{q-1} = 2B/N_{class}$, i.e., the classes are equal-spaced. Thus in the best situation when the classes are equal-spaced, the expected minimum cost to alter a sample's class label is half of the class width, proportional to feature space boundary B and inversely proportional to number of classes N_{class}.

From the result (16) we observe a neat expression of the needed least modification of a sample to alter its class label, given the feature space boundary B and the number of classes N_{class}. This agrees with the intuition that for a fixed space, more classes lead to condensed class boundaries, and vice versa. Such a notion also applies to higher dimensional feature spaces. In the higher N-dimensional situation, the decision boundaries have a dimension of N-1 and could be nonlinear, and whether these higher dimensional boundaries could be analytically derived is unknown to us at the moment. However, the underlying rule still holds: for the network to be robust to adversarial attacks, the feature space should be evenly partitioned so that there does not

exist a class with a smaller region that is easier to attack. In this situation, more classes or a lower feature dimension will reduce the size per region, making the network more vulnerable, and vice versa. In summary, hypothesis 1 believes that when the feature space is unchanged, the cost of attacking the classifier decreases as the number of classes increases; when the number of classes remains the same, the cost of attacking the classifier increases as the feature space increases.

For image classifiers, when the parameters of a single sample are the same (the same dataset), the more the classes of samples, the easier it is to be attacked; when the classes of samples are the same, the feature space of a single sample in different datasets is larger (e.g., the more number of pixels) the harder it is to be attacked.

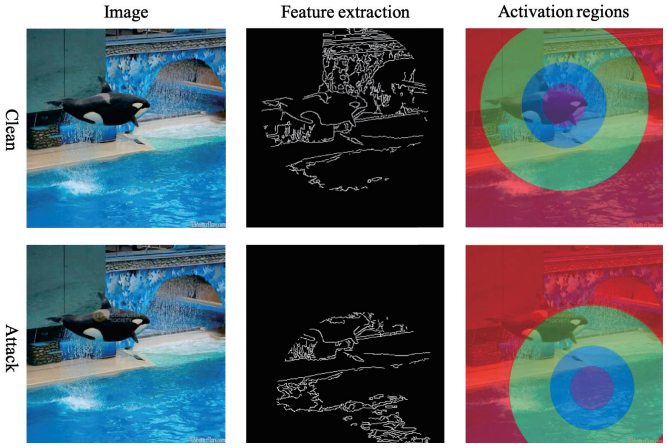


Fig. 3. A demonstration of the expected impact of the attack on the higher-layer features extracted by the classifier and the activation region of the classifier.

3.4.2 Hypothesis 2: On Higher-Layer Features and Activation Regions

Since the classifier pays more attention to local information, by adding limited watermark perturbations, the local features especially higher-layer features extracted by the classifier and activation regions of the classifier are changed, which makes the classifier vulnerable to attacks. Higher layers of DNNs learn more invariant representations, and single image can not maximally activating those neurons. Therefore, we hypothesize that FAWA embeds the watermark into the image, thereby modifying the network to extract higher-layer features, or changing the neurons with the maximal activation. According to Hypothesis 2, the high-level features extracted from the attack image will lose part of the main information of the target in the original image, and the maximum activation area of the attack image will be far away from the position of the target in the original image. Fig. 3 provides an illustrative example for Hypothesis 2. It can be seen from this figure that adding restricted watermark perturbations could change the higher-layer features extracted by the classifier and the activation regions of the classifier, and ultimately lead to changes in the classification results.

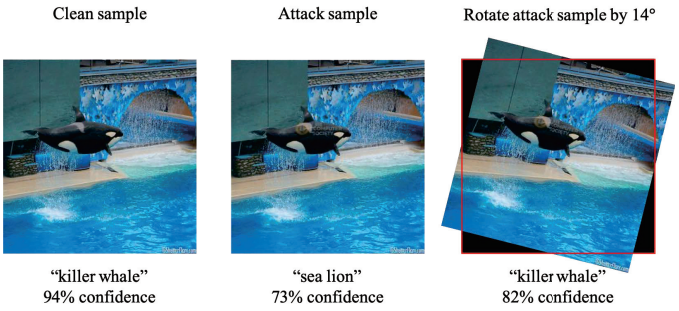


Fig. 4. A demonstration of rotation countermeasure.

4 Countermeasure Methods

According to the previous content, it can be known that the attack effect of the FAWA method on a single image is very sensitive to the transparency, size, direction and embedding position of the watermark. In other words, the attack effect of FAWA is very sensitive to changes in the watermark (such as rotation and filtering). However, it is widely known that DNNs are very good at object detection under rotation and filtering operations. Therefore, we propose two defense methods against FAWA based on rotation and filtering respectively.

The first defense method is based on randomly rotate around the center of the image. By rotating the FAWA sample at a random angle θ_{rotate}, where $|\theta_{rotate}| \leq \theta_{max}$, $\theta_{max} > 0$, the feature spatial distribution of the sample changes [35], and there is a certain possibility to take it return to the decision boundary of the original image's class, thereby changing the output category and restoring it to be consistent with the original image, as shown in Fig. 4. θ_{max} is a hyperparameter. After the centering rotation, the vacant part is filled with black pixels. The defense method is additive and can be applied to a variety of different networks. Only by preprocessing the input data with random rotation can improve the robustness.

The other defense method is based on median filtering. The median filter will smooth the superimposed adversarial watermark pixels and in a way alter its attacking effects. However, for a well trained DNN classifier whose high level feature is normally invariant to smoothed local details, it could still effectively perform classification for smoothed object. Therefore, median filtering of FAWA samples can significantly reduce the FAWA attack effect, but at the same time, it may also reduce the classification accuracy of the original image. Noticeably, the two countermeasure methods do not require any retraining and fine-tuning operations, and they are very easy to implement.

5 Experiments

5.1 Experiment Setting

All the experiments are carried out using an Intel Core I9-10900X CPU and a Nvidia RTX2080Ti GPU. The dataset is prepared as follows. We randomly selected 100 images from the ImageNet [36] validation set, in which all image samples have the same size of $299 \times 299 \times 3$. Experimental results are obtained by averaging over 10 repeated realizations.

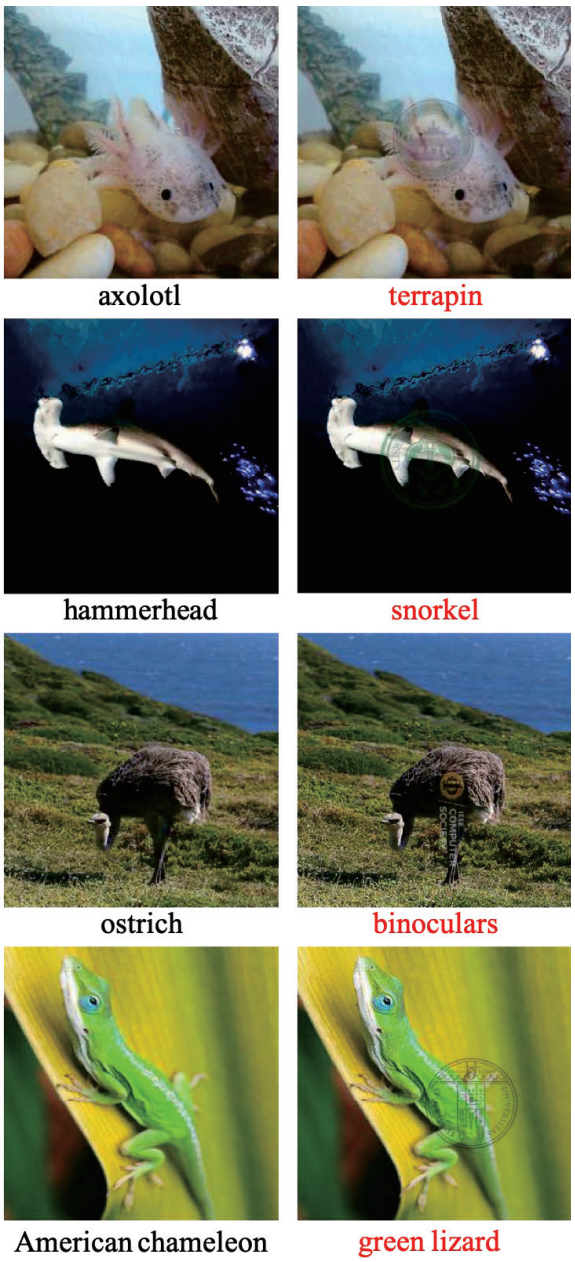


Fig. 5. FAWA samples attacking information systems generated based on the ImageNet dataset, the left column shows clean samples, the right column shows FAWA samples. The watermarks are the logos of Wuhan University, Sun Yat-Sen University, IEEE Computer Society, and Zurich University, from top to bottom respectively.

We tested several publicly available pre-trained models, including VGG-16 [26], Inception v3 [37], GoogLeNet [38], ShuffleNet v2 [39], MobileNet v2 [40], and MNASNet 1.0 [41]. These networks are all pre-trained on the ImageNet dataset. We did not retrain and fine-tune these networks during the entire experiment. To implement reasonable attacks, we set that the height and width of the adversarial watermark cannot exceed 30% of the original image height and width respectively, and the visibility (depth) of the watermark does not exceed 30%. The considered watermark images include the logos of Wuhan University, Sun Yat-sen University, IEEE Computer Society, and Zurich University.

TABLE 1
The classification top-1 accuracy of FAWA samples generated for VGG-16 and ShuffleNet v2 on different networks.

Model	Clean	VGG-16 attack (Minus Clean)	ShuffleNet v2 attack (Minus Clean)
VGG-16	71.1%	**29.8%(-41.3%)**	53.4%(-17.7%)
ShuffleNet v2	70.8%	54.1%(-16.7%)	**29.2%(-41.6%)**
Inception v3	66.7%	50.1%(-16.6%)	47.1%(-19.6%)
GoogLeNet	69.6%	50.7%(-18.9%)	48.9%(-20.7%)
MobileNet v2	73.2%	57.2%(-16.0%)	55.7%(-17.5%)
MNASNet 1.0	72.1%	54.5%(-17.6%)	54.8%(-17.3%)

5.2 Experiments of Attacking Information System, DLaaS, and Real Physical World

Our attack is divided into three parts, including attacking information systems, attacking DLaaS, and attacking real physical world.

5.2.1 Attacking Information System

For VGG-16 network, the impact of the samples generated by the attack on the accuracy of different networks is illustrated in Table 1. It can be observed that the attack generated based on VGG-16 has a strong attacking effect on itself, reducing the top-1 classification accuracy of VGG-16 from 71.1% to 29.8%. This attack has transfer capabilities. For example, it reduces the accuracy of GoogLeNet's top-1 classification from 69.6% to 50.7%. A few of such FAWA examples are shown in Fig. 5.

5.2.2 Attacking DLaaS

For Aliyun Image Recognition APIs, our black-box attack actually faces more unknown situations. Firstly, the number of classes contained in the APIs is unknown to us. Secondly, the APIs can output top-5 category labels and confidence scores which are not equal to 1. Finally, Aliyun provides the free trial version of 5000 queries, after that, the cost is about $1.5 per 5000 queries. From an attacker's point of view, there is a great need for an adversarial generation algorithm with high query efficiency.

We use the same attack settings and use the APIs' top-1 prediction of the original image as a metric. When the top-1 prediction label changes, the algorithm will terminate early. Under this condition, our method can attack Aliyun Image Recognition APIs successfully. Fig. 6 show some of the successful attacking. It is worth mentioning that on a sample subset of 25 images of ImageNet randomly selected, our method achieves a 60% success rate for fooling Aliyun Image Recognition APIs.

5.2.3 Attacking Real Physical World

We further evaluate the proposed FAWA method in real physical world scenario. We randomly selected 20 pictures that successfully attacked the VGG-16 model in Information System, and printed them out with a printer at a ratio of 1:1. Then we used the web camera to take pictures of them, and finally sent the pictures back to the VGG-16 model. Under these conditions, the attack success rate is 45%. Fig. 7 show some of the samples successfully misleading the VGG-16 model in the physical world.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2021.3065172, IEEE Transactions on Computers

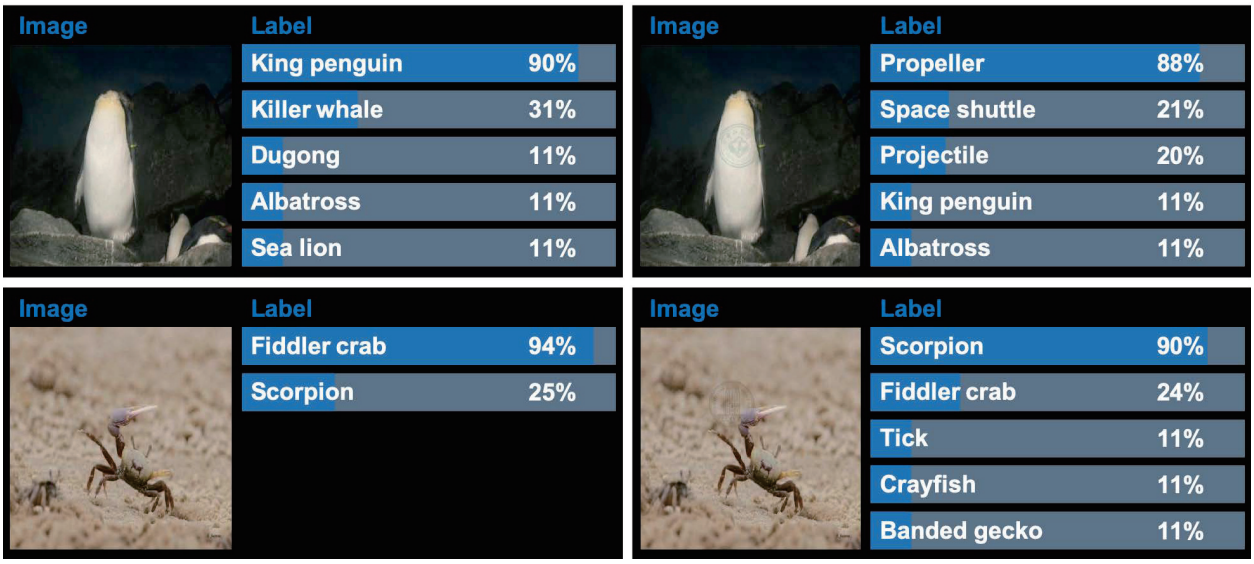


Fig. 6. FAWA samples attacking Aliyun Image Recognition APIs, the left column shows clean samples, the right column shows FAWA samples.

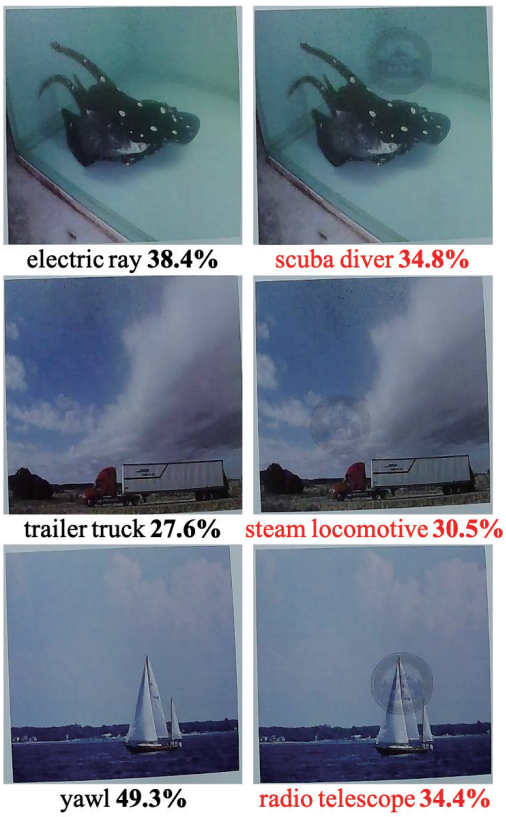


Fig. 7. FAWA samples attacking real physical world generated based on the ImageNet dataset, the left column shows clean samples, the right column shows FAWA samples. These samples are all printed by the printer and captured by the webcam.

Fig. 8. FAWA samples generated based on the Caltech-256 dataset, the left column shows clean samples, the right column shows FAWA samples.

5.3 Validation of Attack Hypotheses and Countermeasures

5.3.1 Verification 0f Attack Hypothesis 1

Under the Caltech-256 dataset [42], samples of 5, 10, 20, 40, and 80 categories are taken to train the VGG-16 classifier. There are 80 samples in each training set and 20 samples in each test set for each category of samples. We randomly generate 100 attack samples for each category. The generated attack samples are shown in Fig. 8.

The top-1 classification accuracy of the VGG-16 model with different categories for clean images and attack images is shown in Fig. 9. Since the experimental results are obtained on the fixed VGG-16, the underlying feature space could be considered as a constant parameter. From this figure we observe that the accuracy curves generally decreases monotonically as a function of the number of classes. Even without the attack, the accuracy of VGG-16 slightly drops for increased number of classes. The proposed

attack successfully leads to drops of accuracy across different number of classes by a clear margin. This clearly verifies our hypothesis 1 on the vulnerability versus feature space and the number of classes.

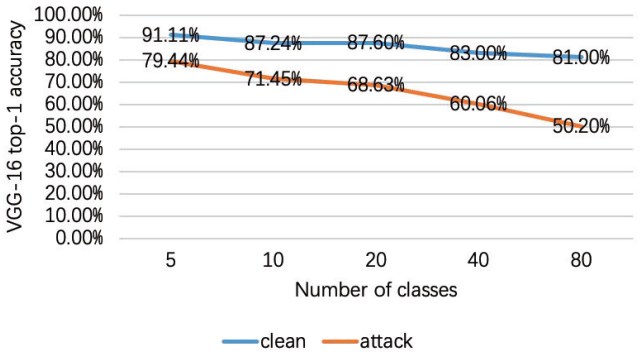


Fig. 9. The top-1 classification accuracy of the VGG-16 model with different numbers of classes for clean samples and FAWA samples.

5.3.2 Verification Attack Hypothesis 2

For the ImageNet dataset, the successfully attacked samples and the corresponding clean samples, the grayscale-guided backpropagation map [43] (obtained from Layer 24 of VGG16) and the score-weighted activation heat map [44] are respectively visualized in Fig. 10.

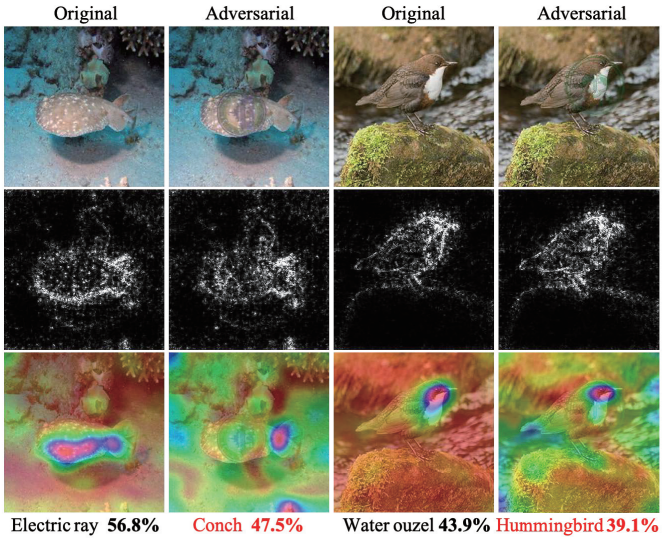


Fig. 10. High-level features and activation heat maps of clean samples and FAWA samples. The first row is the samples, the second row is the grayscale-guided backpropagation map visualization and the third row is the score-weighted activation heat map visualization.

It can be observed from the second row in Fig. 10 that via the proposed FAWA, the higher-layer features extracted from the attack image are very different from the higher-layer features extracted from the original image. Compared with the image in the second row and the first column, the contour information of the target in the second row and second column is partially missing. Moreover, some contour features are added inside the target. Further, from the third row we observe that via the proposed FAWA, the activation area of the attack image are very different from the activation area of the original image. For example, the activation area of the image in the third row and the first column is mainly concentrated in the area of the target object, while the activation area of the third row and second column deviates from the area of the target object. The proposed Hypothesis 2 is hence verified by the above experiments.

5.3.3 Countermeasures

We now evaluate the effectiveness of the proposed countermeasure methods to the FAWA. For the random rotation based method, we randomly rotate the FAWA images generated from VGG-1 and ShuffleNet v2, where the rotation angle is upper bounded by θ_{max}. The attacked and rotated samples are then fed back into VGG-16 and ShuffleNet v2 respectively for image classification. Results averaged on 20 random realizations are summarized in Tables 2 and 3 respectively.

TABLE 2
Countermeasure experimental results using different maximum random rotation angles for the VGG-16 model

θ_{max}	Clean (Minus $\theta_{max} = 0$)	VGG-16 attack (Minus $\theta_{max} = 0$)
0	71.1%	29.8%
5	67.7%(-3.4%)	**43.2%(+13.4%)**
15	62.1%(-9.0%)	41.2%(+11.4%)
35	50.2%(-20.9%)	32.2%(+2.4%)

TABLE 3
Countermeasure experimental results using different maximum random rotation angles for the ShuffleNet v2 model

θ_{max}	Clean (Minus $\theta_{max} = 0$)	ShuffleNet v2 attack (Minus $\theta_{max} = 0$)
0	70.8%	29.2%
5	62.4%(-8.4%)	41.1%(+11.9%)
15	60.1%(-10.7%)	**43.4%(+14.2%)**
35	53.8%(-17.0%)	38.2%(+9.0%)

It can be observed from the two tables that random rotation is effective in compensating the FAWA. Although the accuracy of this method on clean samples in the VGG-16 model is reduced by 3.4%, its defense success rate is increased by 13.4%. Obviously, the classification accuracy of clean samples is sensitive to larger rotation angles. At the same time, the classification accuracy of FAWA samples is sensitive to small rotation angles. Moreover, features and activation area of images in Fig. 4 are shown in Fig. 11. The activation area of the rotated attack image is very close to that of the original image, and even performs better. This not only proves the effectiveness of countermeasure method based on random rotation, but also enhances the persuasive power of Hypothesis 2.

We now evaluate the effectiveness of median filtering on withstanding the FAWA. For ImageNet dataset, FAWA samples generated based on ShuffleNet v2 are filtered through median filtering of different kernel sizes, then input into the pre-trained ShuffleNet v2. The FAWA samples and the FAWA samples after median filtering of different kernel

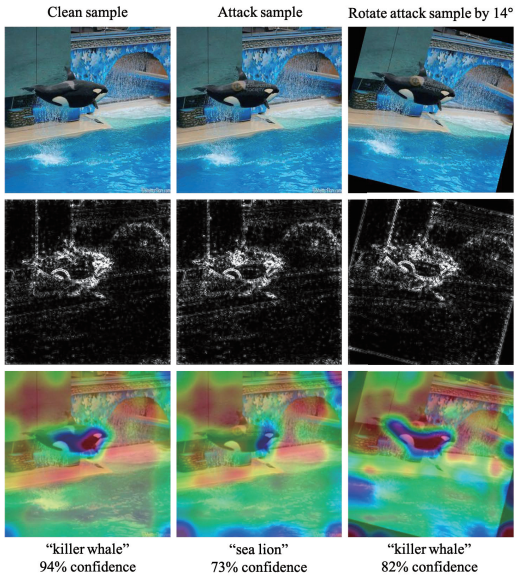


Fig. 11. High-level features and activation heat maps of clean samples of images in Fig. 4.

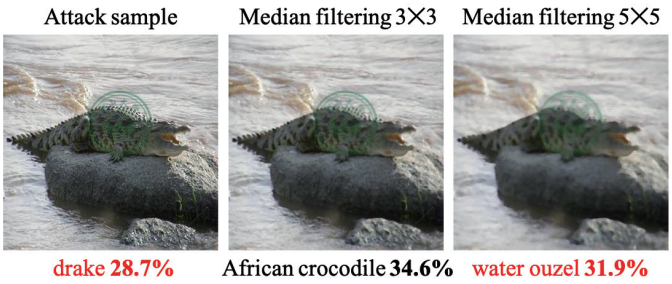


Fig. 12. FAWA samples generated based on ShuffleNet v2 and the FAWA samples after median filtering of different kernel sizes.

sizes are shown in Fig. 12. Table 4 and 5 present the defense experiments of the median filter of different kernel sizes against FAWA for VGG-16 model and ShuffleNet v2 model. Experiments show that median filtering of 3×3 filters achieve the best defense effect against FAWA and can improve the classification accuracy of FAWA samples generated based on ShuffleNet v2 by 20.1%.

TABLE 4
Countermeasure experimental results using the median filter of different kernel sizes against FAWA for the VGG-16 model

Row	Samples	VGG-16 (Minus Row2)
1	Clean samples	71.1%
2	Attack samples based on VGG-16	29.8%
3	Attack samples after 3×3 median filtering	**47.5%(+17.7%)**
4	Attack samples after 5×5 median filtering	45.2%(+15.4%)
5	Attack samples after 7×7 median filtering	40.6%(+10.8%)

Based on the attack samples of VGG-16, mixing two countermeasures with different parameters to generate samples, which are input to the VGG-16 classifier to obtain the top-1 accuracy rate, the results are shown in Table 6. Obviously, mixing the two countermeasures is still effective and slightly better than the single countermeasure, which improves the classification accuracy of FAWA samples generated based on VGG-16 by 24.9%.

TABLE 5
Countermeasure experimental results using the median filter of different kernel sizes against FAWA for the ShuffleNet v2 model

Row	Samples	ShuffleNet v2 (Minus Row2)
1	Clean samples	70.8%
2	Attack samples based on ShuffleNet v2	29.2%
3	Attack samples after 3×3 median filtering	**49.3%(+20.1%)**
4	Attack samples after 5×5 median filtering	45.9%(+16.7%)
5	Attack samples after 7×7 median filtering	40.4%(+11.2%)

TABLE 6
Mixing two countermeasures experimental results against FAWA for VGG-16 attack

θ_{max}	3×3 median filtering	5×5 median filtering
5	54.7%	51.2%
15	52.4%	49.5%

In general, these two defense methods can be deployed as pre-processing procedures if one needs to improve the robustness of the model. Thus, the two methods can be mixed with adversarial training and other defense methods. Furthermore, the two defense methods are easy to calculate and implement. Due to the influence of our defense methods, the models' robustness can be improved to a certain degree.

5.4 Discussions

In this subsection, we note the following open issues pertaining adversarial watermark to DNN models and the corresponding countermeasures.

Rigorous Derivation of the vulnerability of a DNN classifier as a function of the feature space and the number of classes. In this paper, we have derived the expected minimum cost of an adversarial attack for it to be successful in altering the class label of a sample in 1-D feature space. While the underlying concept agrees with our intuition for general cases, the mathematical generalization of this result to higher dimensional spaces seems not straightforward. This is thus considered as one of our future works.

Potential Application in Related Fields. The main task considered in this paper is DNN based image classification. Our future work will further apply adversarial watermark attacks and defenses in other fields, such as semantic segmentation, object detection, speech recognition, and recommendation systems. Moreover, we also attempt to apply adversarial attacks in DLaaS and the physical world.

Advanced Countermeasures. Although the proposed random rotation and median filtering processes have demonstrated certain effectiveness in dealing with the proposed FAWA, these processes have to alter the content of the image. Therefore, it would be preferable if we could remove the watermark attack while maintaining more information about the original clean image. We plan to look into advanced countermeasure methods with watermark removal

capability, which could reconstruct the original image based on mixed loss functions, probably including ℓ_1-loss, multi-scale structural similarity (MS-SSIM) loss, and perceptual loss [45]. ℓ_1-loss and MS-SSIM loss are responsible for controlling the pixel and structure similarity between the reconstructed image and the original image. And perceptual loss is responsible for controlling the similarity between the reconstructed image and the original image in style and content (texture). By reconstruct the original image from the adversarial watermarked image, we expect to obtain an image that is extremely similar to the original image and does not contain watermarks. Furthermore, the reconstructed image can be correctly classified by the image classifier and it is photo-realistic to human perception.

6 Conclusion

In this paper, we have proposed a black-box adversarial watermark attack method to DNN models called FAWA, which consists of transparent watermark generation and fast differential evolution based optimal watermark parameter search. Image samples processed by the proposed FAWA can successfully attack the ImageNet pre-trained DNNs. The experimental results show that our method has good performance in terms of success rate and transferability. The proposed FAWA has been further evaluated against DLaaS and physical world channels respectively, obtaining promising results in misleading the underlying DNN based classifiers. Furthermore, to provide more insights, we have proposed two hypotheses regarding the adversarial watermark attack, which respectively reveal how the feature space and number of classes affect the vulnerability of the model and how the attack misleads deeper feature extraction and activations. Besides, two countermeasure methods are also proposed to compensate the FAWA, which are based on random rotation and median filtering respectively. Generally, this paper could help DNN designers to gain some knowledge of model vulnerability while designing DNN classifiers and related DLaaS applications.

References

[1] J. Park, Y. Chung, and J. Choi, "Codr: Correlation-based data reduction scheme for efficient gathering of heterogeneous driving data," *Sensors*, vol. 20, no. 6, p. 1677, 2020.

[2] J. Dean, "1.1 the deep learning revolution and its implications for computer architecture and chip design," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020, pp. 8–14.

[3] R. R. Cecil and J. Soares, "Ibm watson studio: a platform to transform data to intelligence," in *Pharmaceutical Supply Chains-Medicines Shortages*. Springer, 2019, pp. 183–192.

[4] G. Zhang and M. Ravishankar, "Exploring vendor capabilities in the cloud environment: A case study of alibaba cloud computing," *Information & Management*, vol. 56, no. 3, pp. 343–355, 2019.

[5] G. Li, X. Wang, X. Ma, L. Liu, and X. Feng, "Xdn: towards efficient inference of residual neural networks on cambricon chips," in *International Symposium on Benchmarking, Measuring and Optimization*. Springer, 2019, pp. 51–56.

[6] Y. Zeng, H. Qiu, G. Memmi, and M. Qiu, "A data augmentation-based defense method against adversarial attacks in neural networks," in *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 2020, pp. 274–289.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[8] H. Qiu, T. Dong, T. Zhang, J. Lu, and M. Qiu, "Adversarial attacks against network intrusion detection in iot systems," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2020.

[9] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[11] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[12] X. Jia, X. Wei, X. Cao, and X. Han, "Adv-watermark: A novel watermark perturbation for adversarial examples," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1579–1587.

[13] C. De Vleeschouwer, J.-F. Delaigle, and B. Macq, "Invisibility and application functionalities in perceptual watermarking an overview," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 64–77, 2002.

[14] G. W. Braudaway, "Protecting publicly-available images with an invisible image watermark," in *Proceedings of international conference on image processing*, vol. 1. IEEE, 1997, pp. 524–527.

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

[16] M. Li, H. Ren, E. Zhang, W. Wang, L. Sun, and D. Xiao, "A vq-based joint fingerprinting and decryption scheme for secure and efficient image distribution," *Security and Communication Networks*, vol. 2018, 2018.

[17] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *arXiv preprint arXiv:1707.08945*, vol. 2, no. 3, p. 4, 2017.

[18] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Black-box adversarial sample generation based on differential evolution," *Journal of Systems and Software*, vol. 170, p. 110767, 2020.

[19] D. R. Brownrigg, "The weighted median filter," *Communications of the ACM*, vol. 27, no. 8, pp. 807–818, 1984.

[20] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[21] P.-y. Chiang, R. Ni, A. Abdelkader, C. Zhu, C. Studor, and T. Goldstein, "Certified defenses for adversarial patches," *arXiv preprint arXiv:2003.06693*, 2020.

[22] Aliyun image recognition apis. [Online]. Available: https://help.aliyun.com/product/142958.html

[23] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[24] E. Wong, F. R. Schmidt, and J. Z. Kolter, "Wasserstein adversarial examples via projected sinkhorn iterations," *arXiv preprint arXiv:1902.07906*, 2019.

[25] A. Ghiasi, A. Shafahi, and T. Goldstein, "Breaking certified defenses: Semantic adversarial examples with spoofed robustness certificates," *arXiv preprint arXiv:2003.08937*, 2020.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: adversarial patches to attack person detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[28] A. Liu, X. Liu, J. Fan, Y. Ma, A. Zhang, H. Xie, and D. Tao, "Perceptual-sensitive gan for generating adversarial patches," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1028–1035.

[29] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[30] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

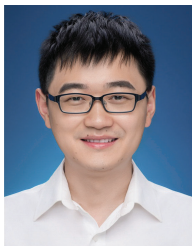
[31] H. Qiu, Y. Zeng, T. Zhang, Y. Jiang, and M. Qiu, "Fencebox: A platform for defeating adversarial examples with data augmentation techniques," *arXiv preprint arXiv:2012.01701*, 2020.

[32] M. S. Kankanhalli, K. Ramakrishnan *et al.*, "Adaptive visible watermarking of images," in *Proceedings IEEE International Conference*

on Multimedia Computing and Systems, vol. 1. IEEE, 1999, pp. 568–573.
[33] Y. Hu and S. Kwong, "Wavelet domain adaptive visible watermarking," *Electronics Letters*, vol. 37, no. 20, pp. 1219–1220, 2002.
[34] S. Bo and B. Vasudev, "Dct domain alpha blending," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, 2002.
[35] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," in *International Conference on Machine Learning*, 2019, pp. 1802–1811.
[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
[39] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.
[40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
[41] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2820–2828.
[42] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.
[43] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *ICLR (workshop track)*, 2015.
[44] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 24–25.
[45] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

Hao Jiang received the B.Eng. degree in communication engineering and the M.Eng. and Ph.D. degrees in communication and information systems from Wuhan University, China, in 1999, 2001, and 2004, respectively. He undertook his Post-Doctoral research work with LIMOS, Clermont-Ferrand, France, from 2004 to 2005 and was a Visiting Professor with the University of Calgary, Canada, and ISIMA, B. Pascal University, France. He is currently a Professor with Wuhan University. His research interest includes adversarial attack, machine learning, and mobile big data. He has authored over 60 papers in different journals and conferences.

Jintao Yang received the B.Eng. degree in information engineering from Wuhan University of Technology, China, in 2015. He is currently pursuing the Ph.D. degree with Wuhan University Electronic Information School. His research interests include wireless communication networks, applied machine learning, and adversarial attacks.



Guang Hua (Member, IEEE) received the B.Eng. degree in communication engineering from Wuhan University, China, in 2009, and the M.Sc. degree in signal processing and the Ph.D. degree in information engineering from Nanyang Technological University, Singapore, in 2010 and 2014, respectively. From July 2013 to November 2015, he was a Research Scientist with the Department of Cyber Security and Intelligence, Institute for Infocomm Research, Singapore. After that, he was with the School of Electrical and Electronic Engineering, Nanyang Technological University, as a Research Fellow, until 2017. He is currently with the School of Electronic Information, Wuhan University. His research interests include multimedia forensics and security, applied convex optimization, applied machine learning, and general signal processing topics. He serves as an Associate Editor for the IEEE Signal Processing Letters.

Lixia Li received the B.Eng. degree in Computer Science from Hubei University of Technology, China, in 2002, and the M.Eng degree from National University of Defense Technology, China, in 2004. He is currently a Professor with Wuhan Digital Engineering Institute. His research interest includes software engineering, machine learning, and data analysis.

Ying Wang received the Ph.D. degree in system analysis and integration from Huazhong University of Science and Technology, China, in 2012. He is currently a Senior Engineer with Wuhan Digital Engineering Institute. His research interests include machine learning, command and control systems, and software engineering.

Shenghui Tu received the B.Eng. degree in information engineering from Wuhan University, China, in 2019. He is currently pursuing the M.Eng. degree with Wuhan University Electronic Information School. His research interests include applied machine learning and adversarial attacks.

Song Xia received the B.Eng. degree in information engineering from Wuhan University, China, in 2020. He is currently with the School of Electronic Information, Wuhan University. His research interests include applied machine learning and certified adversarial robustness.