IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 29, 2021, accepted December 15, 2021, date of publication December 22, 2021,
date of current version December 31, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3137637

Conditional Generative Adversarial Network-Based Image Denoising for Defending Against Adversarial Attack

HAIBO ZHANG[1] AND KOUICHI SAKURAI[2], (Member, IEEE)

[1]Department of Information Science and Technology, Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan

[2]Department of Information Science and Technology, Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan

Corresponding author: Haibo Zhang (zhang.haibo892@s.kyushu-u.ac.jp)

This work was supported in part by the Japan Science and Technology Agency (JST) through the Strategic International Collaborative Research Program (SICORP).

ABSTRACT Deep learning has become one of the most popular research topics today. Researchers have developed cutting-edge learning algorithms and frameworks around deep learning, applying them to a wide range of fields to solve real-world problems. However, we are more concerned about the security risks associated with deep learning models, such as adversarial attacks, which this article will discuss. Attackers can use the deep learning model to create the conditions for an attack, maliciously manipulating the input images to deceive the classification model and produce false positives. This paper proposes a method of pre-denoising all input images to prevent adversarial attacks by adding a purification layer before the classification model. The method in this paper is proposed based on the basic architecture of Conditional Generative Adversarial Networks. It adds the image perception loss to the original algorithm *Pix2pix* to achieve more efficient image recovery. Our method can recover noise-attacked images to a level close to the actual image to ensure the correctness of the classification results. Experimental results show that our approach can quickly recover noisy images, and the recovery accuracy is 20.22% higher than the previous state-of-the-art.

INDEX TERMS Adversarial attack, conditional generative adversarial network, image denoising.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

The development of machine learning technology has brought so much wonder and convenience to human life. Researchers have developed cutting-edge learning algorithms and frameworks around deep learning, applying them to a wide range of fields to solve real-world problems. As a branch of machine learning, the widespread use of deep learning has provided researchers with more diverse research directions. For example, deep learning algorithms have played an essential role in image recognition, and classification with significant success, the most widely used are neural networks. After repeated training with large amounts of data, the neural network model can be trained to become a competent classifier capable of correctly identifying images and giving appropriate classification results. However, the appearance of adversarial attacks brings challenges to neural networks.

An adversarial attack adds some noise to the input data that humans cannot detect to make the model make a wrong judgment on the input data. The added noise is called Adversarial Perturbation, and the sample obtained after adding noise is called Adversarial Sample.

The inputs to a deep learning algorithm can be images, text or numeric vectors. Constructing the input vectors in a specific way for the model to generate erroneous results is known as an adversarial attack. Such errors arise from the imperfection and sensitivity of machine learning models. A machine learning model consists of a series of specific transitions between nodes, and most of these transitions are very sensitive to slight changes in the input. An attacker can exploit this sensitivity to maliciously attack machine learning models, causing severe consequences such as erroneous judgments by Artificial Intelligence (AI) devices [1]. This type of behavior is an essential issue for AI security.

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-Hoon Kim.

It is a worthwhile research topic to study an efficient adversarial attack defense method to solve the above problems. In recent years, researchers have proposed different research solutions, such as enhancing the robustness of machine learning models by training them with attacked images as a way to counteract adversarial attacks or reducing noises of the input vectors to ensure the cleanliness [2].

To enable a better understanding of what we discuss in this paper, this paper is organized to answer the following questions:

1) What is an adversarial attack? What should we know before we understand counter-attacks?
2) How exactly is the approach proposed in this paper implemented? Is there a sufficient theoretical foundation to support it?
3) What are the results of the experiments? Does it prove that the method proposed in this paper is effective?
4) Does this method bring new inspiration to the field of adversarial attack? What are the future research prospects?

B. RELATED WORKS

There are various ways to defend against adversarial image attacks, and the applied objects can be divided into two main categories, the input images and the neural networks used for classification [3]. A common way to defend against adversarial image attacks is by detecting the input image data and recovering as much image information as possible to ensure the accuracy of the classifier. Another common defense is to train the neural network model with the attacked image samples to enhance the robustness of the model, thus making the classifier more efficient in defending against malicious adversarial attacks. Xia *et al.* [4] introduced one method, named gradient traps, for evaluating the error between the actual adversarial robustness and the evaluated one to help the community to develop more robust defenses.

Liao *et al.* [5] proposed a high-level representation guided denoiser, name HGD, as a defense for image classification. Their method can overcome the problem of the error amplification effect, in which slight residual adversarial noise is progressively amplified and causes the incorrect category. They use a loss function defined as the difference between the target model's outputs activated by the clean image and denoised image based on the Convolutional Neural Network algorithm.

Adversarial Logit Pairing (ALP) [6] is an adversarial training method in which a network of clean images and its adversarial samples are predicted similarly. They explained the idea by using the prediction results of the clean image as a ''noise-free'' reference so that the adversarial sample learns the features of the clean image for denoising. This method achieves 55.4% and 77.3% accuracy on the ImageNet dataset for white-box and black-box attacks, respectively.

Xie *et al.* [7] proposed an image feature-based denoising scheme based on the ALP method. In addition, they offered a denoising module on the high-level feature map of the network to facilitate better learning of neat features in the shallow part of the network. They trained and tested both white-box and black-box PGD attacks and obtained significantly better results than the ALP method.

Immediately afterward, Shafahi *et al.* [8] proposed free adversarial training. In contrast to the large-scale and time-consuming model training cycles of ALP [6] and Xie *et al.* [7], they proposed to re-use neural network backward pass to compute the descending gradients and update the model weights. Their approach can significantly reduce the model training period and obtains an accuracy of 40% on the Imagenet dataset against PGD attacks.

Li *et al.* [9] proposed to apply an iterative adversarial training scheme to an external auto-encoder to protect other models directly. Their method is based on purifying adversarial perturbations against white-box attacks. Similarly, Hwang *et al.* [10] introduced a process of purifying variational auto-encoder by eliminating an adversarial perturbation and determining the closest projection as a purified sample.

In terms of the smooth adversarial training, Miyato *et al.* [11] propose a method based on virtual adversarial loss: a new measure of local smoothness of the conditional label distribution is given input. Similar to Miyato's work, Xie *et al.* [12] proposed an adversarial training method specifically to improve the robustness of neural networks. They introduced the smooth adversarial training (SAT), which replaces the ReLU activation function with its smooth approximations to strengthen adversarial training. The SAT achieved the purpose of no drop in accuracy and no increase in computational cost compared to standard adversarial training.

Our paper is a denoising work of the input image data and does not change the pre-trained classifier model. However, based on this, we reviewed the recent related results of previous studies.

Santhanam and Grnarova [13] introduced a fundamental work of training the entire generative adversarial network on the same dataset for defending against adversarial attacks. Samangouei *et al.* [14] proposed a new framework, named Defense-GAN, which leverages the expressive capability of generative models to defend deep neural networks against adversarial attacks. Their work can be used with many classification models and as a defense against many attacks. It does not assume knowledge of the process for generating adversarial examples.

Gupta and Rahtu [15] proposed a defense mechanism that applies reconstruction only to small and carefully selected image regions that have the most impact on the current classification results. The selection process presented in the paper is guided by a class activation map function corresponding to a certain number of top-ranked class labels. It is experimentally demonstrated that the most salient regions for the adversarial perturbations are the same, and therefore these are the ones that need the most purification.

To address the problem that DeepNude software can be easily recreated, Yeh *et al.* [16] proposed to take an

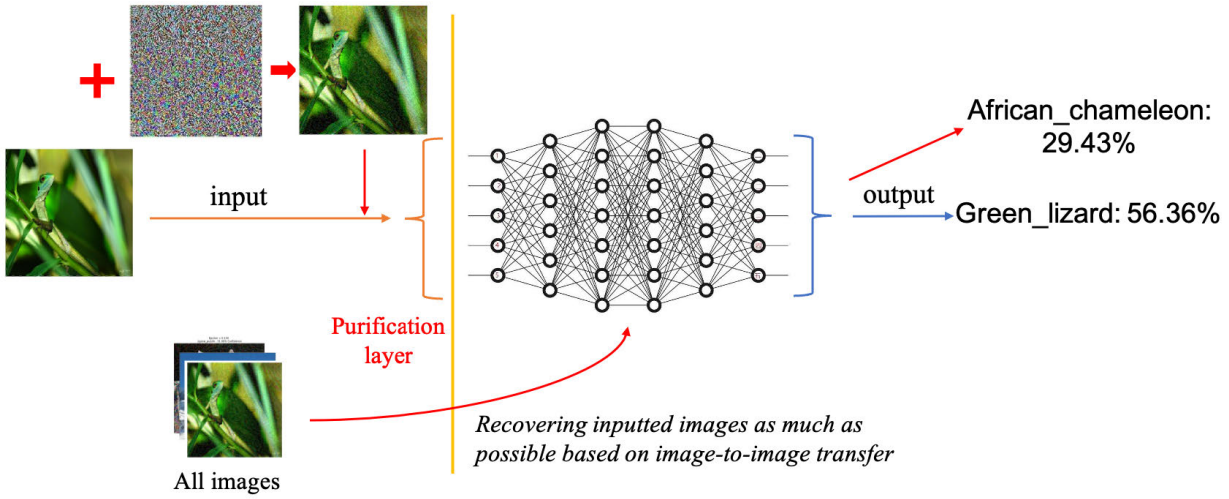


FIGURE 1. The overall structure of our proposed model. The purification layer is added before the image classifier. This purification layer pre-denoises all input images and uses the pre-denoised output as the input to the classifier.

alternative perspective of image translation algorithms, i.e., to effectively use the possibility of adversarial attacks against these algorithms to defend against malicious attacks. Furthermore, they presented modifying the input image following the adversarial loss so that these algorithms do not easily counterfeit the edited image.

The recent work, Li *et al.* [17] applied their Feature Pyramid Decoder (FPD) framework to all block-based convolutional neural networks, including denoising and image restoration modules. In this way, they need to train all neural network modules in the system.

Researchers have proposed many excellent and effective methods to defend against adversarial attacks. Both ALP and Xie *et al.*'s approaches have achieved significant results in training classifier models with adversarial samples to enhance the robustness of the models.

C. CHALLENGING ISSUES

Existing approaches to defending against adversarial attacks by adversarial training do enhance the robustness of neural network models. However, we can still ask whether more potent adversarial samples can successfully attack the adversarially trained models. We believe that the answer is possible. Pre-denoising the input images before reaching the classifier can prevent the neural network model from being adversarial attacked more effectively. Traditional machine learning methods can deal with image denoising, but the model training period is long, and the results are not optimal. How we can quickly pre-denoise input images and recover the original information of the images to the maximum extent to ensure the accuracy of the final classifier model output is a pressing problem.

D. OUR CONTRIBUTION

In this paper, we propose the Conditional Generative Adversarial Network-based image denoising method to defend against adversarial image attacks, which is an improvement on the *Pix2pix* algorithm [18]. Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to propose an improved image-to-image translation algorithm based on Conditional Generative Adversarial Network to solve the adversarial attack problem in the image recognition domain.
- Our method can quickly pre-denoise a single image in 0.02 seconds and recover more than 50% of the accuracy of the classification model.
- Compared to previous state-of-the-art results, our approach can be applied to many different types of adversarial attacks and recovers more accuracy of the classification model.
- Currently, no research explicitly proposes an approach for adversarial defense based on image transformation algorithms. And our paper fills this gap. Our research effectively exploits this classical algorithm to the field of adversarial attacks and proves its feasibility by extensive experiments.

We propose a practical purification function of all input images, i.e., one pre-denoising method. As a result, the final classifier model receives image information close to the original image information. In this way, even if an attacker uses high-intensity adversarial perturbation, our model can recovery the input images to the maximum extent.

Fig.1 shows the overall structure of our proposed model, i.e., a purification layer is added before the image classifier. This purification layer pre-denoises all input images and uses the pre-denoised output as the input to the classifier.

Compared with existing studies, our approach strives to be effective in image denoising from the perspective of image pre-processing. Image recovery using traditional machine learning methods is time-consuming and ineffective. Our approach incorporates the Conditional Generative Adversarial Network algorithm. It uses the Convolutional Neural

Network model to effectively extract feature information from each image layer to better train the model. Our model can quickly complete the image purification process without any additional training process for the classifier model. Therefore, in theory, our model can be applied to any classifier.

E. OVERVIEW OF THIS PAPER

The remainder of this paper is organized as follows. Section II introduces some terminologies closely related to this paper to facilitate understanding. Section III illustrates in detail the theoretical basis of our proposed method. Section IV describes all the experimental procedures and the results of this topic. Section V provides an objective evaluation of the experimental results. Finally, section VI discusses the experimental results in more depth and analyzes the reasons that enabled the current performance to be obtained.

II. TERMINOLOGY

This section introduces several relative terminologies to understand the proposed methodology in this article better.

A. DEEP LEARNING MODELS

Deep learning is derived from artificial neural networks and is part of a cluster of machine learning methods based on artificial neural networks. Deep learning can be divided into supervised, semi-supervised, or unsupervised learning and is usually represented by discovering the data's distributed features.

1) NEURAL NETWORKS

The neural network algorithm is essential in machine learning. It is the core algorithm of the whole deep learning, and deep learning is an extension based on the neural network algorithm [19].

A multi-layer neural network consists of the input layer, hidden layer, and output layer. Each layer is composed of units, in which the input layer is passed in from the instance feature vector in the training dataset and passed to the next layer based on the weights between the connected nodes, thus passing one layer forward. Both input and output layers have only one layer, and the number of hidden layers can be arbitrary. As a multi-layer forward neural network, it is theoretically possible to simulate any equation if there are enough hidden layers and training sets. Thus, we can use neural networks to solve classification and regression problems.

2) GENERATIVE ADVERSARIAL NETWORKS

Goodfellow *et al.* [20] proposed Generative Adversarial Networks(GAN) in 2014. The GAN contains two types of neural networks, Generator (G) and Discriminator (D). The generator is used to generate the image; after inputting a random code (z), it will output a fake image G(z), which is automatically generated by the neural network. The other network, Discriminator, is used to judge. It accepts the image output by G as input and then determines whether this image is true or false, outputting 1 for true and 0 for false [21].

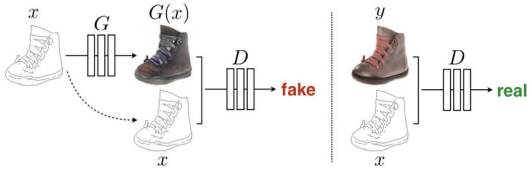


FIGURE 2. Architecture of pix2pix [18].

As the two networks play each other, both networks become more and more capable: G generates images that look more and more like real images, and D becomes better and better at determining the authenticity of images. Our learning goal is to minimize the ability of D to judge G while maximizing the capacity of D.

$$\min_G \max_D \sum [logD(G(z)) + log(1 - D(x))]$$

To enhance the capability of D, we consider the cases of inputting actual images and false images separately. The first term of D(G(z)) in the above equation deals with the false image G(z) when the score D(G(z)) needs to be reduced as much as possible; the second term deals with the actual image x when the score should be high.

Since researchers have discovered the power of GAN, different types of derivative models of GAN have been proposed for different problems. For example, Deep Convolutional-GAN [22], Conditional-GAN [18], which adds a condition to the discriminator, and info-GAN [23], ACGAN [24], and Cycle-GAN [25], [26], which are derivative models of Conditional-GAN.

3) CONDITIONAL GAN

In image recognition and classification, Convolutional Neural Networks (CNN) are excellent at extracting features at different levels of the image. The research methodology in this paper is based on the image-to-image translation algorithm. The most classical algorithm for image-to-image translation is the *pix2pix* method [18]. The *pix2pix* method introduces *Conditional GAN (cGAN)* as a general-problem solution. This model can be trained not only by learning the mapping from input images to output images but also by learning loss functions to tune the model [27].

Fig.2 describes the basic architecture of the *pix2pix* algorithm. The generator G uses the U-net structure. The reason for using the U-net design is that for the image translation task, the input and output should share some underlying information, so a skip connection such as U-net is used, where layer *i* is directly added to layer *n-i*. The input contour map *x* is encoded and then decoded to generate the real image. The discriminator D uses the conditional discriminator *PatchGAN* proposed by the original author, whose role is to judge the generated image *G(x)* as false and the real image as true under the condition of the contour map *x*.

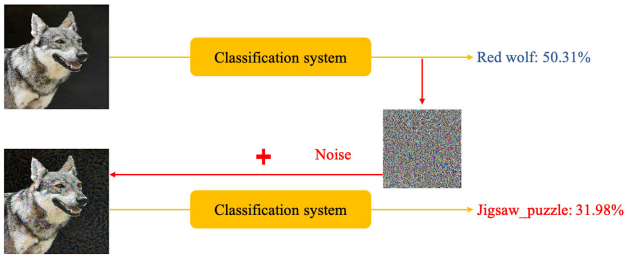


FIGURE 3. Adversarial threat model.

The core idea of cGAN is to add conditional constraints. The cGAN architecture has two inputs, potential variables and control conditions. Assuming that the inputs of G are z and y, y is a condition, which can usually be a vector or a value, and z is a potential vector, then the generated result can be expressed as $x = G(z|y)$. The discriminator has two functions, one is to determine how well the images generated by G match the real sample, and the other is to determine whether the input images match the given condition y.

B. ADVERSARIAL THREAT MODEL

In 2013, Szegedy *et al.* [28] found that adding some small perturbations to the data can change the classification results of the model and even make the model produce the same classification results for different data by adding perturbations. They first tried to solve the equation for the minimum perturbation that would allow the neural network to make a misclassification.

Recent research by Google Brain has shown that any machine learning classifier can be tricked into giving incorrect predictions. Artificial intelligence and machine learning technologies are currently being used in a wide range of areas such as human-computer interaction, recommendation systems and security protection. Attackers attempt to bypass or directly attack machine learning models for countermeasure purposes through various means. Particularly in human-computer interaction, with voice and images as emerging means of human-computer input, their convenience and usefulness are being welcomed by the public.

The accuracy of the recognition of speech and images is crucial to the effectiveness of the machine in understanding and executing the user's commands. At the same time, this aspect is also the easiest for attackers to exploit, by making minor modifications to the data source, to the extent that the user does not perceive it and the machine accepts it and makes a wrong subsequent action [29]. The attacker attempts to provide a maliciously altered image input to the classification system causing the classification system to produce a false classification [3].

Fig.3 shows the basic process of the adversarial attack. For the given image, both the human eye and the trained machine learning model can accurately identify the "Red wolf" without the adversarial attack. However, once the adversarial attack occurs, although the human eye can still accurately identify the "Red wolf," it has become another picture for the machine learning model. It thus is recognized as a "Jigsaw_puzzle."

There are two general ways of classifying adversarial attacks [3]:

1) White-box attack versus Black-box attack
   - **White-box attack:** The attacker has access to the algorithms used for machine learning and the parameters used by the algorithms. As a result, the attacker can interact with the machine learning system in generating the adversarial attack data.
   - **Black-box attack:** The attacker is not aware of the algorithms and parameters used for machine learning. However, the attacker can still interact with the machine learning system, for example, bypassing arbitrary inputs to observe the output and determine the result.
2) Targeted attack versus Untargeted attack
   - **Targeted attack:** For an image, an adversarial sample is generated such that the system's annotation on it is identical to the target annotation, i.e., it is required not only that the attack is successful but also that the generated adversarial sample belongs to a specific class.
   - **Untargeted attack:** An adversarial sample is generated for an image such that the annotation on it is independent of the original annotation, i.e., as long as the attack succeeds, there is no restriction on which class the adversarial sample ultimately belongs to.

Because of the vulnerability and instability inherent in deep neural networks, researchers have provided more research space in the field of adversarial attacks on deep neural networks by proposing various attack methods [30]–[33]. This paper presents several representative models of adversarial attacks.

1) FAST GRADIENT SIGN METHOD (FGSM)

Goodfellow *et al.* [1] developed a method that can efficiently compute the adversarial perturbation. And the method for solving the adversarial perturbation is referred to as FGSM in the original text.

FGSM is a single attack method where for the original sample x and the corresponding label y, it is shifted one small step ϵ in the direction of the gradient of the loss function, i.e.

$$x_{adv} = x + \epsilon sign(\nabla_x \mathcal{L}(\theta, x, y))$$

where $\mathcal{L}$ is the loss function, so that the generated adversarial samples can ensure that the l_∞ distance between the adversarial sample x and the original sample y does not exceed ϵ.

2) BASIC & LEAST-LIKELY-CLASS ITERATIVE METHODS

Kurakin *et al.* [34] proposed image perturbation by increasing the loss function of the classifier by one large step

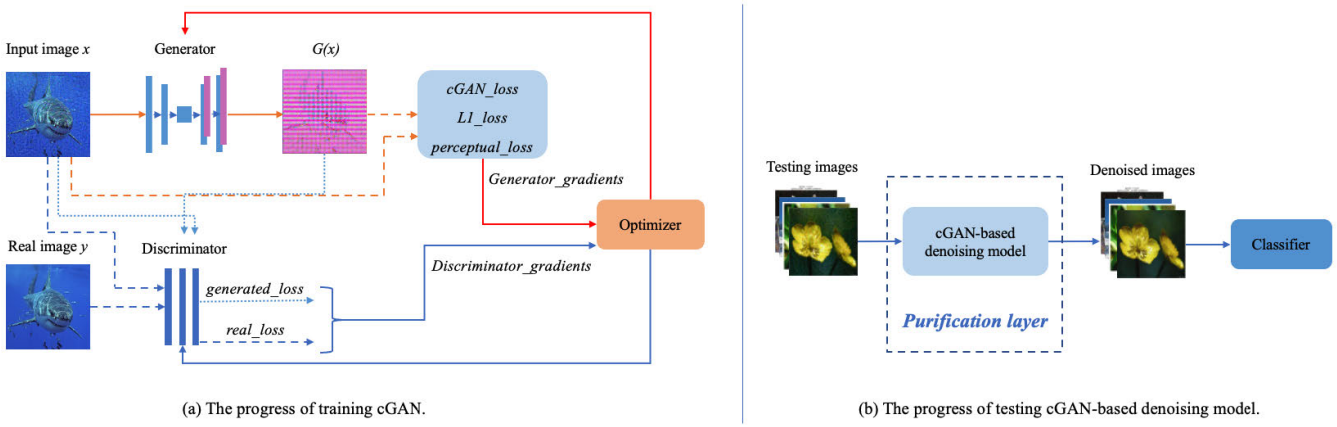


FIGURE 4. Description of cGAN-based model training progress and testing progress. The part (a) explains the whole training progress of the image denoising model. The input to the model is a pair of images, where the noise-attacked image *x* is used as the input to the *Generator* and the *Discriminator*, respectively, and the real image *y* is used as the input to the *Discriminator*. The part (b) depicts the complete process during the testing phase, which consists of two separate neural network models, cGAN-based purification neural network model and the image classification neural network model. The input to the system is a set of noise-attacked images, which first need to be passed through the purification layer based on a pre-trained cGAN-based denoising model, and the purified images are used as the input to the image classifier.

operation, and thus can be directly extended to a variant by increasing the loss function by multiple small steps.

3) DeepFool

Moosavi *et al.* [35] generated minimum normative adversarial perturbations by an iterative computational method that gradually pushes images located within the classification boundary outside the boundary until a misclassification occurs. The ideal prerequisite for using Deepfool is that the target deep neural network is completely linear, i.e., there exists a hyperplane that can partition different classes of data. The authors proved that they generated more minor perturbations than FGSM while having similar deception rates.

4) PROJECTED GRADIENT DESCENT (PGD)

Madry *et al.* [36] proposed the Projected Gradient Descent (PGD) adversarial attack method. PGD algorithm is a variation of FGSM, which is an attack algorithm that generates adversarial samples and a defense algorithm that is trained against them. PGD achieves the ability to deceive a neural network with only a slight perturbation, and It increases the robustness of the model by using adversarial samples to train the model.

5) JACOBIAN-BASED SALIENCY MAP

Papernot *et al.* [37] proposed a method using a Jacobian matrix to evaluate the sensitivity of the model to each input feature, then using a Saliency map to select perturbations to obtain an adversarial sample by combining its Jacobian matrix to rank the contribution of each input feature to the misclassified target. This approach presented to restrict the l_0 parametric value, i.e., to change the value of only a few pixels instead of perturbing the whole image.

6) UNIVERSAL ADVERSARIAL PERTURBATIONS

While methods such as FGSM and DeepFool can only generate adversarial perturbations for a single image, Universal Adversarial Perturbations proposed by Moosavi *et al.* [38] can generate perturbations that achieve attacks on any image, which are also nearly invisible to humans. The approach used in this paper is similar to DeepFool in that it uses adversarial perturbations to push images out of the classification boundary. Still, the same perturbation is used for all images.

7) ONE PIXEL ATTACK

Su *et al.* [39] used a differential evolutionary algorithm to generate sub-images by iteratively modifying each pixel and comparing them with the parent image, and retaining the sub-image with the best attack according to the selection criteria to achieve the adversarial attack. This adversarial attack does not require any information about the network parameters or gradients to be known.

III. PROPOSED METHOD

This paper refers to the *Pix2pix* method; in the cGAN environment, the paired dataset consists of noisy images and original images. We train our image denoising model based on the fundamental framework of cGAN, combined with *Pix2pix* and feature extraction algorithm, by minimizing the loss value between the input image and the original image.

A. ARCHITECTURE

Fig.4 (a) explains the whole training progress of the image denoising model. The input to the model is a pair of images, where the noise-attacked image x is used as the input to the *Generator* and the *Discriminator*, respectively, and the real image y is used as the input to the *Discriminator*. The *Generator* generates the images $G(x)$ based on the existing parameters. At this point, the loss values generated by the *Generator*

and the *Discriminator* need to be calculated separately. The *Generator* calculates loss values between *G(x)* and *x*, which includes *cGAN_loss*, *L1_loss* and *perceptual_loss*, and the specific calculation method will be introduced in the subsection B. We refer to these loss values as *generator_loss*. The *Discriminator* computes two loss values, one is the *real_loss* between the *x* and *y*, and the other is the *generated_loss* between the *x* and *G(x)*. The sum of these two loss values becomes the *discriminator_loss*. The next step is to optimize the *Generator* and *Discriminator* by using gradient descent.

It should be noted that the purpose of *Generator* is to confuse *Discriminator*. Therefore, as the model is trained, the *generator_loss* decreases and the *discriminator_loss* increases.

Fig.4 (b) depicts the complete process during the testing phase, which consists of two separate neural network models, the cGAN-based purification neural network model and the image classification neural network model. The input to the system is a set of noise-attacked images, which first need to be passed through the purification layer based on a pre-trained cGAN-based denoising model, and the purified images are used as the input to the image classifier. In this system, we do not use a neural network model that combines the two functions of filtering and classification. The image classifier also has a critical role as a judge of our proposed cGAN-based model. The classifier recognizes outputs of the purification layer to verify the effectiveness of the purification layer.

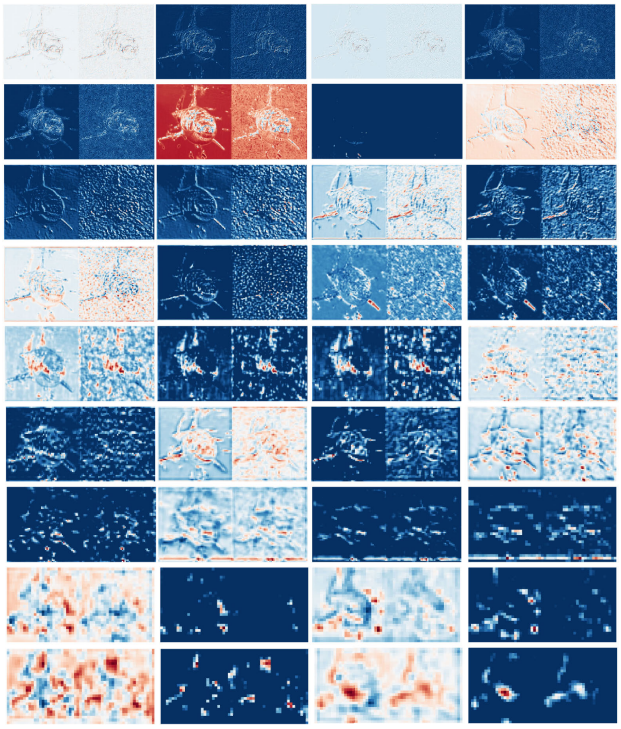


FIGURE 5. Extracted features for each image layer using VGG19 model. For each feature map, the left one is the feature of real image, the right one is the feature of noise-attacked image.

B. LOSS FUNCTIONS

An image can be divided into low-frequency parts, such as pixel values of color blocks, and high-frequency parts, such as image edges. In order to maximize the recovery of image information, it is necessary to calculate loss values for both the low-frequency part and high-frequency part.

1) PIXEL LOSS

Firstly, the most efficient way to recover the low-frequency information of an image is to calculate the pixel loss between color blocks [40]. *L1* and *L2* loss functions are both able to minimize the pixel loss of an image very well. Compared with *L1* loss, *L2* can achieve convergence faster. For image recovery processes that combine simultaneous computation of high-frequency information loss, *L1* loss can ensure that the optimal solution is not missed while the recovery effect converges slowly. Hence, we select *L1* loss as our pixel loss function:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x,z)||_1]$$

L1 loss calculates the loss value for each corresponding pixel blocks between the real image *y* and the generated image $G(x,z)$.

2) cGAN LOSS

The texture feature value is a good criterion for the recovery of high-frequency information in an image. The original *Pix2pix* method proposed the structure of *patchGAN* for the *Discriminator* in order to better judge the image partly. Specifically, the *patchGAN* divides the image into patches, judges the truth or falsity of each patch separately, and finally takes the average. This *patchGAN* can be regarded as another form of texture loss. The authors confirmed that a patch size of 70*70 could get better results through specific experiments.

We extended the loss function from the original cGAN model, i.e., the quality of the generated image has to pass the review of *Discriminator*. The optimization function is:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[logD(x,y)] + \mathbb{E}_{x,z}[log(1 - D(x, G(x,z)))]$$

The first term of the above equation, the input of *Discriminator* has two parts, the input image *x* and the condition *y* (real image). If the label is the same as the label of the real image, the output number is close to 1, otherwise it will only be close to 0. The second term contains $D(x, G(x,z))$, which compares the generated image $G(x,z)$ and the label of *y* to optimize *Discriminator* and *Generator*.

3) PERCEPTUAL LOSS

The original cGAN loss function was initially designed to be applied in studies with widely varying image styles. Therefore, cGAN loss is hardly helpful for highly-matched images, such as noise-attacked images versus real images. Perceptual loss function [41] is used to compare two different images that look similar; although the images are

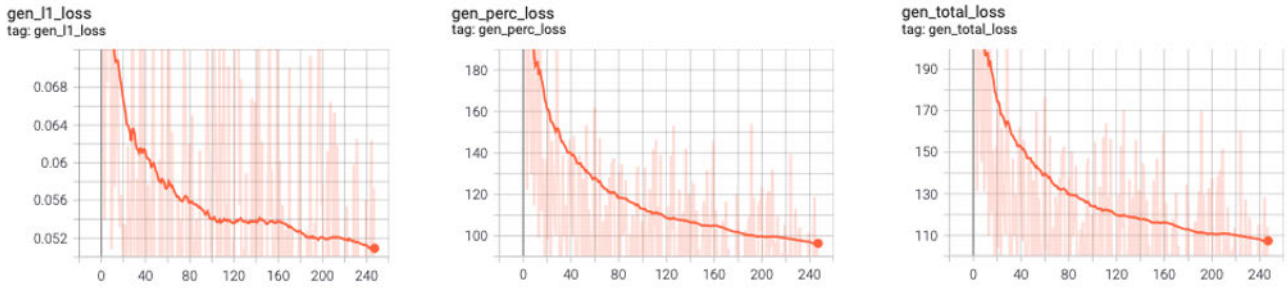


FIGURE 6. The trend of three loss-function values during the training progress. All three loss functions gradually reach the minimum value and converge after the epoch 240.

very similar, pixel loss will output a considerable error value. On the other hand, the Perceptual loss function compares high-level perceptual and semantic differences between images.

Without affecting the training speed of the model, we simultaneously extract features from each layer of both the generated image and the actual image, then calculate the perceptual loss for all layers except for the top layer. We use the pre-trained VGG19 model [42] to achieve the feature extraction.

$$\mathcal{L}_{perceptual}(G) = \mathbb{E}_{x,y}[\sum_{k} a_k ||V_k(y) - V_k(G(x))||_1]$$

Fig.5 shows the 36-layer feature maps extracted using the VGG19 model for noise-attacked images and real images, including the CONV, RELU, and POOL layers. It can be seen that there are also high loss values for the low-level features of images, and these low-level features are beneficial for comparing very similar images. A simple pixel-level loss can approximate these low-level feature tensors. The images are better recovered by calculating the loss values between feature maps.

To summarize, our final objective function is:

$$G^* = \min_G \max_D \mathcal{L}_c GAN(G, D) + \lambda_1 \mathcal{L}_{L1}(G) + \lambda_2 \mathcal{L}_{perceptual}(G)$$

where the *generator_loss* is minimized and the *discriminator_loss* is maximized by repeating the training and optimization process. In specific experiments, we found that the recovery of images is better when the value of *L1* is 100, and the value of *L2* is 1.

IV. EXPERIMENT

Our study is based on the image classification research area. For the judge of image recovery effect, we choose *MobileNetV2* [43] model as the final image classifier. We choose the FGSM attack as the adversarial attack model and generate 5000 noised images so that the *MobileNetV2* model cannot classify the input image correctly.

Table 1 shows the detailed configuration of the equipment used in our experiments.

TABLE 1. Configuration of the equipment used in our experiments.

System	Ubuntu 20.04.2 LTS
Processor	Intel Core i9-10900
Graphics	GeForce RTX 2080 SUPER
Programming	Python, Tensorflow, Jupyter

A. TRAINING PROGRESS

We randomly select 5000 images from the Imagenet ILSVRC2010 dataset [44] as our training set. We perform model training on this dataset based on both the original *Pix2pix* algorithm and our proposed cGAN-based improved algorithm with 300 epochs.

Fig.6 shows the trend of three loss-function values during the training progress. It can be seen that all three loss functions gradually reach the minimum value and converge after epoch 240.

Fig.7 and Fig.8 show the training progress of *Pix2pix* method and our method on epoch 0, epoch 50 and epoch 100. Comparing these two training processes shows that our approach can reach convergence more quickly and better recovery. Naked eyes can well observe the solid color area in the image for the image recovery effect. At epoch 50, the blue area in the predicted image of Fig.8 is closer to the blue area of the real image, and there is still much noise in the predicted image of Fig.7. At epoch 100, the green space in the predicted image of Fig.8 is already very close to the green area of the real image, while the blue area in the predicted image of Fig.7 can still be seen with apparent noise.

B. TESTING RESULTS

As the test dataset, we randomly selected 5000 images from the ImageNet ILSVRC2010 dataset to test the image recovery effect of the above two trained models.

We train the *Pix2pix* model and our proposed cGAN-based model for 5000 noise-attacked images, respectively, and then test 5000 attacked images. Fig.9 shows that the classification results of the shot images all have significantly reduced confidence and even were incorrectly classified into other categories. The images recovered by the Pix2pix-based method

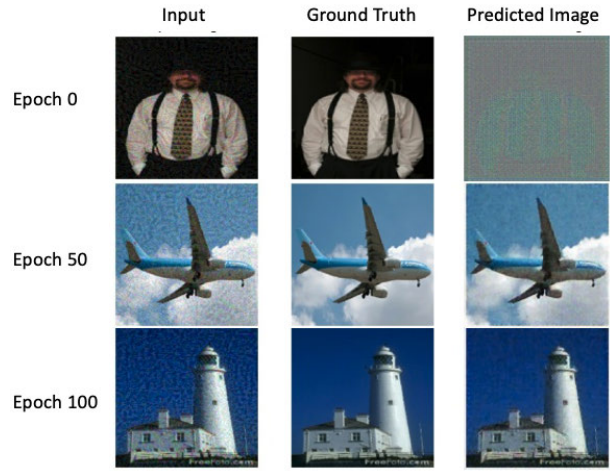


FIGURE 7. Training progress of Pix2pix-based method.

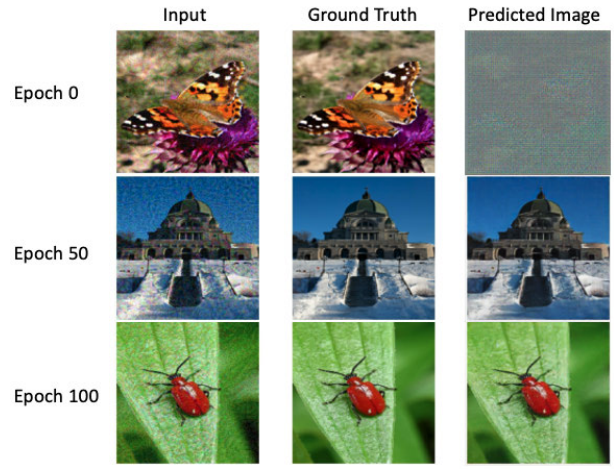


FIGURE 8. Training progress of our method.

were able to successfully recover to the class of the original images, although there was still a gap in the confidence level. The images recovered by our method not only succeeded in restoring the images to the category of the original images but also had a higher confidence level compared to *Pix2pix* method. And from the naked eye, we can find that images recovered by our method are closer to the clarity of the original images.

Fig.10 shows that in some cases, the *Pix2pix* method can only reduce the confidence of image misclassification, while our approach can successfully restore the images to their original categories. For that matter, the images recovered by our method are more easily recognized by the classifier.

Fig.11 shows that there are also cases where neither method can restore the image to the original category. Still, our process can reduce the confidence of image misclassification to a greater extent compared to the *Pix2pix* approach.

V. EVALUATIONS

In this section, we evaluate the results of our experiments using a variety of evaluation criteria.

FIGURE 9. The images recovered by our method not only succeeded in restoring the images to the category of the original images but also had a higher confidence level compared to *Pix2pix* method.

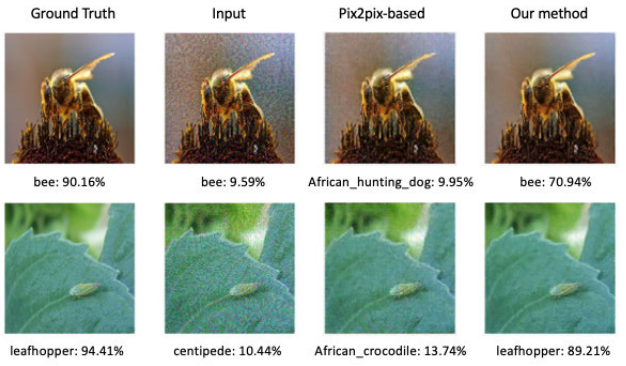


FIGURE 10. The *Pix2pix* method can only reduce the confidence of image misclassification, while our method can successfully restore the images to their original categories.

A. VISUAL QUALITY

First, by the judgment of human eyes, it can be seen in Fig.9, Fig.10 and Fig.11 that the images recovered by our method are closer to the quality of the real images. Especially in the solid color region, the images recovered by the *Pix2pix* method still have a certain degree of noise visible to human eyes. In contrast, the image recovered by our method has almost no noise visible. Therefore, such a degree of recovery of the image is recognized.

B. ACCURACY

We first calculate the accuracy of image classification using *MobileNetV2* classifier for different cases. Table 2 shows

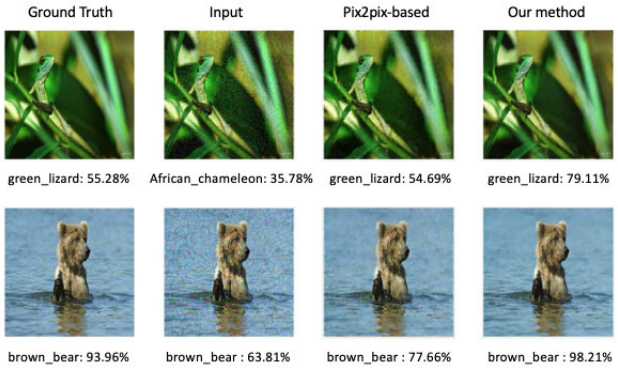


FIGURE 11. There are also cases where neither method can restore the image to the original category. Still, our method can reduce the confidence of image misclassification to a greater extent compared to the *Pix2pix* approach.

TABLE 2. Accuracy of images in different situations against FGSM attacks.

Method	Accuracy
Ground-truth	54.83%
FGSM-attacked	24.64%
Pix2pix	44.78%
Our method	50.34%

TABLE 3. The property of misclassification for different method.

Method	Misclassification
FGSM-attacked	46.66%
Pix2pix	29.22%
Our method	14.09 %

that the classification accuracy of FGSM-attacked images decreases by 30.19% relative to the accuracy of real images, rising by 20.14% after recovery by *Pix2pix* method, and increases by 25.7% after recovery by our practice.

C. MISCLASSIFICATION

Next, we calculate the success rate of misclassifying images. For example, if the label of the real image is a cat and the classification result of the attacked image is a dog, then we consider this misclassification as successful. If the classification result of the recovered image is still not a cat, then the misclassification is still successful. Table 3 shows that the misclassification rate of the recovered images after the *Pix2pix* method decreased by 17.44%, and the misclassification rate of the recovered images after our method decreased by 32.57% relative to FGSM-attacked images.

D. CONFIDENCE

The confidence value of image classification results is also a good measurement of image recovery effectiveness. As shown in Table 4, the average classification confidence

TABLE 4. Average confidence of different method.

method	average confidence
Ground-truth	76.1%
FGSM-attacked	37.32%
Pix2pix	53.57%
Our method	69.2 %

TABLE 5. Accuracy of images in different situations against PGD (10 iterations) attacks.

Method	Accuracy	
	Inception_V3	ResNet_101
Ground-truth	55.35%	61.50%
PGD-attacked	0.9%	29.45%
ALP	27.9%	-
Xie et al.	-	49.7%
Our method	48.12%	58.99%

of the FGSM-attacked images decreased by 38.7% relative to real images. While the average confidence value of the *Pix2pix*-recovered images increased by 16.25%, the average confidence value of the recovered images by our method increased by 31.88% relative to the attacked images.

E. COMPARISON WITH PGD-ATTACK

To compare with the previous state-of-the-art, we also trained and tested our model for the PGD attack. As mentioned in section III, ALP and Xie *et al.* have achieved good results on PGD attacks. For the credibility of the comparison, we also use the Inception_V3 model and ResNet_101 model as the final image classifier, respectively.

Table 5 shows that the classification accuracy of real images is 54.83% when the classifier is model Inception_V3. The classification accuracy of PGD-attacked images drops to 0.9%, a decrease of 53.93%. ALP method can increase the accuracy by 27%, and our method can grow 47.22%. The classification accuracy of real images is 61.50% when the classifier model is ResNet_101, and the classification accuracy of PGD-attacked images drops to 29.45%, a decrease of 32.05%. Xie's method can increase the accuracy by 20.25%, and our method can grow by 29.54%.

It can be seen that by comparing ALP and Xie's method, our method can better recover noise-attacked images and can significantly improve the accuracy of image classification.

VI. DEEP ANALYSIS

A. WHY CHOOSE IMAGE-TO-IMAGE TRANSLATION

In this paper, we choose image-to-image translation algorithms as the basis of our research. The main reason for this choice is that the image-to-image translation-based cGAN model has a powerful ability to solve the image denoising

problem. For the image denoising problem, all we need to do is restore the noisy image to the clarity of the original image as much as possible, and the presence of the original image becomes a must. This is also similar to the principle of the ALP algorithm. cGAN can learn and make difficulties for the *discriminator* in the presence of both the noisy image and the original image, thus making both the *generator* and the *discriminator* progressively more powerful, which also dramatically exploits the learning ability of GAN.

On the other hand, the image-to-image translation method can avoid the counter-effect of denoising images that might be equivalent to the addition of adversarial perturbations to the image. This is because recovering the image by calculating the loss value from the original image does not add additional unnecessary information and only, to some extent, mitigates the attack on the image caused by the adversarial perturbation.

B. WHY PERFORMS BETTER WITH PERCEPTUAL_LOSS

The original *Pix2pix* method has only *L1_loss* and *cGAN_loss* for the *generator_loss*, which makes the discriminator and generator lose the ability to analyze image information at the feature level. Our method remedies this deficiency by adding *perceptual_loss*. The generator converges more slowly with the addition of *perceptual_loss*. This slow convergence enhances the ability of image recovery.

C. ADVERSARIAL PERTURBATION

In the world of adversarial attacks, there are many ways to generate adversarial perturbations. The FGSM and PGD attack methods mentioned in this paper are both based on developing an adversarial sample x_{adv} by shifting it a small step ϵ in the direction of the gradient of the loss function for the real sample x and the corresponding label y. This noise is global concerning the image, i.e., denoising can be performed by computing pixel point loss and perceptual loss values.

However, we consider other kinds of adversarial perturbations, such as pixel attacks. Su *et al.*'s one-pixel attack presented an extreme attack in which only one pixel needs to be changed to change the classification result of the model. In our experiments, when we use our method on one-pixel attacked images, it does not produce any effect, i.e., the images do not change in any way. The reason for this is that our model defaults that altered pixel to be part of the real image for the attacked image because the difference of this pixel is negligible for both pixel loss and perceptual loss. Experimentally, we demonstrate that our image denoising method is not applicable to defend against pixel attacks, and this may not be treated as the denoising task.

Therefore, for defending against pixel attacks, the Convolutional Neural Network (CNN) is possibly more efficient than multi-layer perceptions used for cGAN. CNN is better able to extract image features and even subtle differences.

D. DATASETS

All experiments in this paper were performed on the ImageNet dataset. We argue that the global adversarial perturbation has a more pronounced effect on high-resolution images. Even to be used as an input vector for the model, the resolution of adjusted images is still 256*256 pixels. Nowadays, ImageNet is utilized in most papers as one of the most general datasets for image recognition. Therefore, it is very suitable as a criterion to evaluate research results.

For those attacked images (FGSM and PGD), we attacked all images by ourselves. For the FGSM attack, we attacked 5000 images with ϵ 0.3. For the PGD attack, we attacked 5000 images with ϵ 0.3 and iterated ten times. Since only these two attacks were verified, this is the drawback of our study. Our subsequent work requires experiments on more kinds of attacks to confirm whether our proposed method is effective for any attacks.

E. ROBUSTNESS

The robustness of the classifier can be enhanced by training the model using adversarial samples, which is mentioned in many studies. The method proposed in this paper adds an image purification layer before the classification neural network and does not change the classification model. Therefore, the robustness of our model considers whether all images will be purified. The answer is yes. Even with high-intensity adversarial perturbations, it is difficult to trick the purification layer into missing specific images. Therefore, the robustness of our model is guaranteed.

F. TIME CONSUMING

The method proposed in this paper adds an image purification layer before the classification neural network, which then necessarily causes additional time consumption. The average extra time consumed by our model is experimentally derived as 0.02 seconds. The addition of the image purification layer does not significantly affect the efficiency of the overall image classification process.

Another aspect of the time-consuming problem of the method proposed in this article is that the pre-processing for all input images inevitably generates some additional time consumption that could have been saved. If an attacker makes a large-scale adversarial attack on a particular input image dataset, our model can efficiently clean up these noisy images. However, suppose the attacker only makes adversarial attacks on a few of these images or does not perform any adversarial attacks. In that case, our model can cause a large amount of unnecessary time consumption. If there are no attacks, the image will not change after the filtering layer processes it and will only incur excessive time consumption. Therefore, the next step for this model is to perform the necessary adversarial attack detection before the purification layer. Efficient adversarial attack detection can improve the speed of image pre-processing to a great extent.

VII. CONCLUSION

This paper proposes a method to defend against adversarial image attacks by adding a perceptual-loss function to the traditional cGAN method. Our approach maximizes the recovery of the original image's information by applying the denoising method to all input images to help the classifier produce more accurate classification results. It isn't easy to fully recover the original information of an image. All training processes are required to prevent overfitting problems, considering that overtraining can cause overfitting. Therefore, it isn't easy to achieve 100% image recovery, but this paper aims to recover the maximum amount of original information and correct the classifier output.

In this paper, our study focuses on experimentally verifying the feasibility of the proposed method with significant results. However, we do not discuss the theoretical derivation more in-depth. In the following work, we will focus on the theoretical derivation to verify the feasibility of this method.

REFERENCES

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

[2] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, pp. 1–41, May 2020.

[3] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," 2018, *arXiv:1810.00069*.

[4] P. Xia, Z. Li, H. Niu, and B. Li, "Understanding the error in evaluating adversarial robustness," 2021, *arXiv:2101.02325*.

[5] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1778–1787.

[6] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial logit pairing," 2018, *arXiv:1803.06373*.

[7] C. Xie, Y. Wu, L. V. D. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 501–509.

[8] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" 2019, *arXiv:1904.12843*.

[9] H. Li, Q. Xiao, S. Tian, and J. Tian, "Purifying adversarial perturbation with adversarially trained auto-encoders," 2019, *arXiv:1905.10729*.

[10] U. Hwang, J. Park, H. Jang, S. Yoon, and N. Ik Cho, "PuVAE: A variational autoencoder to purify adversarial examples," 2019, *arXiv:1903.00585*.

[11] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[12] C. Xie, M. Tan, B. Gong, A. Yuille, and Q. V. Le, "Smooth adversarial training," 2020, *arXiv:2006.14536*.

[13] G. K. Santhanam and P. Grnarova, "Defending against adversarial attacks by leveraging an entire GAN," 2018, *arXiv:1805.10652*.

[14] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," 2018, *arXiv:1805.06605*.

[15] P. Gupta and E. Rahtu, "CIIDefence: Defeating adversarial attacks by fusing class-specific image inpainting and image denoising," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6708–6717.

[16] C.-Y. Yeh, H.-W. Chen, S.-L. Tsai, and S.-D. Wang, "Disrupting image-translation-based DeepFake algorithms with adversarial attacks," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW)*, Mar. 2020, pp. 53–62.

[17] G. Li, S. Ding, J. Luo, and C. Liu, "Enhancing intrinsic adversarial robustness via feature pyramid decoder," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 800–808.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[19] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1, no. 2. Cambridge, MA, USA: MIT Press, 2016.

[20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[21] H. Chen, "Challenges and corresponding solutions of generative adversarial networks (GANs): A survey study," *J. Phys., Conf. Ser.*, vol. 1827, no. 1, Mar. 2021, Art. no. 012066.

[22] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[23] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," 2016, *arXiv:1606.03657*.

[24] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[26] D. Bashkirova, B. Usman, and K. Saenko, "Adversarial self-defense for cycle-consistent GANs," 2019, *arXiv:1908.01517*.

[27] G. G. Chrysos, J. Kossaifi, and S. Zafeiriou, "RoCGAN: Robust conditional GAN," *Int. J. Comput. Vis.*, vol. 128, nos. 10–11, pp. 2665–2683, Nov. 2020.

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

[29] H. Bae, J. Jang, D. Jung, H. Jang, H. Ha, H. Lee, and S. Yoon, "Security and privacy issues in deep learning," 2018, *arXiv:1807.11655*.

[30] W. Zhao, P. Yang, R. Ni, Y. Zhao, and W. Li, "Cycle GAN-based attack on recaptured images to fool both human and machine," in *Proc. Int. Workshop Digit. Watermarking*. New York, NY, USA: Springer, 2018, pp. 83–92.

[31] Z. Che, A. Borji, G. Zhai, S. Ling, J. Li, X. Min, G. Guo, and P. L. Callet, "SMGEA: A new ensemble adversarial attack powered by long-term gradient memories," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 9, 2021, doi: 10.1109/TNNLS.2020.3039295.

[32] L. Zhai, F. Juefei-Xu, Q. Guo, X. Xie, L. Ma, W. Feng, S. Qin, and Y. Liu, "It's raining cats or dogs? Adversarial rain attack on DNN perception," 2020, *arXiv:2009.09205*.

[33] Z. Che, A. Borji, G. Zhai, S. Ling, J. Li, Y. Tian, G. Guo, and P. L. Callet, "Adversarial attack against deep saliency models powered by non-redundant priors," *IEEE Trans. Image Process.*, vol. 30, pp. 1973–1988, 2021.

[34] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, 2016.

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

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

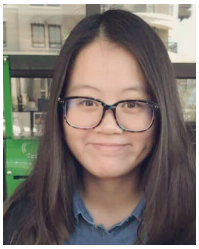
[37] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 2016, pp. 372–387.

[38] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[39] J. Su, D. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.

[40] Z. Pan, W. Yu, B. Wang, H. Xie, V. S. Sheng, J. Lei, and S. Kwong, "Loss functions of generative adversarial networks (GANs): Opportunities and challenges," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 4, pp. 500–522, Aug. 2020.

[41] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 694–711.
[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.



HAIBO ZHANG received the B.S. degree in software engineering from Anhui University, China, in 2015, and the M.S. degree in cyber security engineering from the Viterbi School of Engineering, University of Southern California, USA, in 2018. She is currently pursuing the Ph.D. degree in cyber security with Kyushu University, Japan. She is currently working as a Research Assistant of Strategic International Collaborative Research Program (SICORP) for the Internet of Things related research work. Her research interests include Internet of Smart Things security, digital supply chain security, and vulnerable software clone detection.

KOUICHI SAKURAI (Member, IEEE) received the B.S. degree in mathematics from the Faculty of Science, Kyushu University, in 1986, and the M.S. degree in applied science and the doctorate degree in engineering from the Faculty of Engineering, Kyushu University, in 1988 and 1993, respectively. He was engaged in research and development on cryptography and information security at the Computer and Information Systems Laboratory, Mitsubishi Electric Corporation, from 1988 to 1994. Since 1994, he has been working with the Department of Computer Science, Kyushu University, in the capacity of an Associate Professor, and became a Full Professor, in 2002. He is also concurrently working with the Institute of Systems Information Technologies and Nanotechnologies, as the Chief of the Information Security Laboratory, for promoting research co-operations among the industry, university, and government under the theme "Enhancing IT-security in social systems." He has been successful in generating such co-operation between Japan, China, and South Korea for security technologies as the Leader of a Cooperative International Research Project supported by the National Institute of Information and Communications Technology (NICT), during 2005 to 2006. In March 2006, he established research co-operations under a memorandum of understanding in the field of information security with Professor Bimal Kumar Roy, the first time Japan has partnered with The Cryptology Research Society of India (CRSI). He directs the Laboratory for Information Technology and Multimedia Security and is also working with the CyberSecurity Center of Kyushu University. He is also currently working with Department of Advanced Security of Advanced Telecommunications Research Institute International and involved in a NEDO-SIP-Project on supply chain security. He has published about 400 academic papers around cryptography and cybersecurity (See http://dblp.uni-trier.de/db/indices/a-tree/s/Sakurai:Kouichi.html).