Journal of Computers Vol. 32 No. 5, 2021, pp. 55-75
doi:10.53106/199115992021103205005

An Ensemble Denoiser Based on Generative Adversarial Networks to Eliminate Adversarial Perturbations

Rui Yang[1,3], Tian-Jie Cao[1,3], Xiu-Qing Chen[2*], Feng-rong Zhang[1,3], Yun-Yan Qi[1,3]

[1] Department of Computer Science and Technology, China University of Mining and Technology, XuZhou, Jiangsu 221116, PR China
tjcao@cumt.edu.cn

[2] School of Medicine Information and Engineering, XuZhou Medical University, XuZhou, Jiangsu 221000, PR China
xiuqingchen@126.com

[3] Mine Digitization Engineering Research Center of Ministry of Education, China University of Mining and Technology, Xuzhou, Jiangsu 221116, PR China

Received 4 March 2021; Revised 5 July 2021; Accepted 1 August 2021

Abstract. Deep neural networks (DNNs) have been applied in various machine learning tasks with the success of deep learning technologies. However, they are surprisingly vulnerable to adversarial examples, which can easily fool deep neural networks. Due to this drawback of deep neural networks, numerous methods have been proposed to eliminate the effect of adversarial examples. Although they do play a significant role in protecting deep neural networks, most of them all have one flaw in common. They are only effective for certain types of adversarial examples. This paper proposes an ensemble denoiser based on generative adversarial networks (GANs) to protect deep neural networks. This proposed method aims to remove the effect of multiple types of adversarial examples before they are fed into deep neural networks. Therefore, it is model-independent and cannot modify deep neural networks' parameters. We employ a generative adversarial network for this proposed method to learn multiple mappings between adversarial examples and benign examples. Each mapping behaves differently for different types of adversarial examples. Therefore, we integrate these mappings as the ultimate method to defend against multiple types of adversarial examples. Experiments are conducted on the MNIST and CIFAR10 datasets. We compare this proposed method with several existing excellent methods. Results show that this proposed method achieves better performance than other methods when defending against multiple types of adversarial examples. The code is available at https://github.com/Afreadyang/ensemble-ape-gan.

Keywords: adversarial example defense, generative adversarial networks, deep neural networks, deep learning, artificial intelligence security

1 Introduction

With the rise of deep learning technologies, deep neural networks have made great success in various machine learning fields, such as computer vision [1], natural language processing [2], speech recognition [3], etc. However, Szegedy et al. [4] pointed out that they are vulnerable to adversarial examples, maliciously designed to trick deep neural networks. It turns out that even perceptually indistinguishable adversarial examples can easily fool deep neural networks.

For adversarial example generation methods, a great deal of white-box or black-box algorithms have been proposed in recent years. In the white-box setting, adversaries have access to all the internal information of the deep neural network. L-BFGS [4] based on constraint optimization and FGSM [5]

* Corresponding Author

based on gradient optimization are typical white-box algorithms. Although the time efficiency of FGSM is faster than that of L-BFGS, the attack capacity of FGSM is weaker than that of L-BFGS. Based on these two types of white-box algorithms, numerous subsequent studies have focused on enhancing their performance [32, 35]. In the black-box setting, adversaries cannot obtain any internal information of the deep neural network. They can only get the probabilistic output of the deep neural network. These existing black-box algorithms include search-based algorithms [6], evolution-based algorithms [7], algorithms based on gradient estimation [8], algorithms based on decision boundary estimation [9], etc. Compared with white-box algorithms, black-box algorithms usually take more time and have weaker attack capability.

For adversarial example defense methods, numerous excellent algorithms have been proposed to eliminate the effect of adversarial examples. In general, these algorithms include two aspects. The first is to make deep neural networks more robust. These algorithms include data enhancement [5], regularization [15], randomization [16], input transformation [17], etc. They are usually model-dependent and can modify deep neural networks' parameters. The second is detecting adversarial examples or recovering adversarial examples to benign examples before they are fed into deep neural networks. These algorithms include adversarial example classifier [10], statistical analysis [11], prediction based on density and uncertainty [12], modification loss [13], reconstruction loss [14], etc. They are usually model-independent and cannot modify deep neural networks' parameters. Although these existing defense methods have achieved encouraging performance in defending against adversarial examples, most of them all have some bottlenecks. For example, data enhancement requires extensive data and is time-consuming. Besides, they are only effective for certain types of adversarial examples. Santhanam et al. [31] pointed out that current detection-based methods are challenging to make a reliable distinction between adversarial examples and benign examples.

Due to these drawbacks of existing adversarial example defense methods, we propose a novel method to protect deep neural networks. The idea behind this novel method is to remove the effect of multiple types of adversarial examples before they are fed into deep neural networks. Therefore, it is model-independent and cannot modify deep neural networks' parameters. This proposed method is an ensemble denoiser based on generative adversarial networks [18]. Generative adversarial networks are an excellent way to model data distribution. Ensemble-based methods can defend against more types of adversarial examples. Therefore, this proposed method can recover multiple types of adversarial examples to benign examples. First, we employ a generative adversarial network to learn multiple mappings between adversarial examples and benign examples. The used generative adversarial network is a combination of AC-GAN [19] and WGAN-GP [20]. The classification loss from AC-GAN can enhance the capacity of the generator to recover adversarial examples to benign examples. The loss function in the form of WGAN-GP can ensure the training process is stable. Besides, the network structure of the generator is based on UNET [24]. This structure can further improve the capacity of the generator to learn the mapping between adversarial examples and benign examples. Second, each mapping behaves differently for different types of adversarial examples. Therefore, we integrate these mappings as the ultimate method to defend against multiple types of adversarial examples. This proposed method makes full use of the advantages of generative adversarial network and ensemble-based methods. Experiments are conducted on the MNIST and CIFAR10 datasets. We compare this proposed method with several existing excellent methods. Results show that this proposed method achieves better performance than other methods when defending against multiple types of adversarial examples.

Contributions of this paper are summarized as follows:

(1) We propose an ensemble denoiser based on generative adversarial networks to protect deep neural networks. Generative adversarial networks are used to model data distribution. Ensemble-based methods can defend against more types of adversarial examples.

(2) The used generative adversarial network is a combination of AC-GAN and WGAN-GP. AC-GAN can improve the capacity of the generator to learn the mapping between adversarial examples and benign examples. WGAN-GP can ensure the training process is stable.

(3) The network structure of the generator is based on UNET. This structure has deeper network layers. Therefore, it can further enhance the capacity of the generator to recover adversarial examples to benign examples.

(4) Each learned mapping behaves differently for different types of adversarial examples. Therefore, we integrate these learned mappings as the ultimate defense.

The rest of this paper will be organized as follows. Section 2 is a brief overview of adversarial example defense methods. Then, in section 3, this paper introduces the detail of the proposed ensemble denoiser. The experiments and results analysis are conducted in Section 4. Finally, the conclusion is offered in Section 5.

2 Related Work

In this part, we will briefly review these recent adversarial example defense methods, especially GAN-based defense methods. Besides, we will also compare the differences between these existing methods and our proposed method.

2.1 Adversarial Example Defense Methods

Xu et al. [14] proposed two feature squeezing methods to eliminate redundant features from inputs. The first is color bit-depth reduction, and the other is local or non-local spatial smoothing. Based on these two methods, they proposed to detect adversarial examples by comparing the predictions between squeezed and unsqueezed inputs. The input is marked as adversarial if the predictions' difference is more significant than a certain threshold. Xu et al.'s method and our method are based on input transformation. Both are model-independent and cannot modify deep neural networks' parameters. The difference is that Xu et al.'s method is to detect adversarial examples, while our method is to recover adversarial examples to benign examples. Besides, Xu et al.'s input transformation is based on digital image processing technologies. Our input transformation is based on learning the mapping between adversarial examples and benign examples.

Assuming that adversarial examples lie inside the adversarial data manifold, Feinman et al. [12] proposed two detection strategies: Kernel Density Estimates (KDE) and Bayesian Uncertainty Estimates (BUE). The purpose of KDE is to identify whether data points are far from data manifolds, while BUE can be used to detect data points near low-confidence regions where KDE is not practical. Both Feinman et al.'s method and our method assume that adversarial examples lie inside the adversarial data manifold. They are model-independent and cannot modify deep neural networks' parameters. The difference is that Feinman et al.'s method is to detect adversarial examples, while our method is to recover adversarial examples to benign examples. Besides, Xu et al.'s strategy is based on statistics. Our method is based on learning the mapping between adversarial examples and benign examples.

Defensive distillation [22] is a defense that works by training two networks. The student network is trained to approximate the teacher network. First, the teacher network is trained on the output layer using a modified softmax function, including a temperature constant. The higher temperature results in a flatter softmax, which introduces more noise in the decision-making process. Second, the student network is trained using the predictions from the teacher network as the training labels. The hope is that this prevents the distilled network from overfitting. Defensive distillation is model-dependent and can modify deep neural networks' parameters. This is different from our method.

MagNet [23] is a robust multi-pronged method that includes a detector and a reformer network. They can all reconstruct benign examples. The detector is employed to detect adversarial examples. If the input is adversarial, the reconstruction loss will be high. Adversarial detection can be performed by setting a threshold value for reconstruction loss. The reformer network is used to recover adversarial examples to benign examples. The detector is similar to Xu et al.'s method. The reformer network is similar to our method. The difference is that the reformer network a simple autoencoder, while our method is a generative adversarial network. Besides, our method's generator is based on UNET, whose structure has deeper network layers.

Yang et al. [17] proposed a preprocess-based defense called Me-Net, which preprocesses the input in the hope of destroying adversarial examples. Me-Net works by first discarding pixels randomly in the input image based on a certain probability, assuming this eliminates adversarial perturbations. The image is then reconstructed using a matrix estimation algorithm, a method of recovering matrix data from noise observations. Yang et al.'s method is similar to our method. Both methods are model-independent and cannot modify deep neural networks' parameters. The difference is that Yang et al.'s input transformation is based on a matrix estimation algorithm. Our input transformation is based on a generative adversarial network.

2.2 GAN-based Adversarial Example Defense Methods

Samangouei et al. [25] proposed Defense-GAN to protect deep neural networks. It consists of two steps. First, a generative adversarial network is trained to learn the distribution of benign examples. Then, the generator takes adversarial examples as inputs and finds close outputs to benign examples. This is an optimization process. Samangouei et al.'s method is model-independent and cannot modify deep neural networks' parameters. This is the same as our method. The difference is that Samangouei et al.'s method is to find a new benign example that is close to adversarial examples, while our method is to recover an original adversarial example to a benign example.

Shen et al. [26] proposed APE-GAN to eliminate the effect of adversarial examples. APE-GAN's idea is the same as our method. They can recover adversarial examples to benign examples by learning the mapping between adversarial examples and benign examples. The difference is that our method is based on AC-GAN and WGAN-GP. The network structure of the generator is based on UNET, which has deeper network layers. APE-GAN is based on a simple generative adversarial network. The network structure of the generator is shallow. Therefore, our method is more stable and has a better performance than APE-GAN.

Lee et al. [27] proposed an adversarial training framework based on generative adversarial networks. They alternately train both classifier and generator. The generator crafts adversarial examples that can easily fool the classifier. Simultaneously, the classifier is trained to classify both adversarial examples and benign examples correctly. These procedures help the classifier to become more robust to adversarial examples. Liu et al. [28] also designed a GAN-based adversarial training defense, dubbed GanDef, which utilizes a competition game to regulate the feature selection. GanDef contains a classifier and a discriminator, which form a minimax game. The discriminator is used to determine whether the feature extracted by the classifier is robust. Hashemi et al. [29] proposed Noise-GAN to protect deep neural networks against adversarial attacks. Noise-GAN includes a multi-class discriminator that uses different loss functions to generate adversarial examples. Adversarial examples generated by Noise-GAN are used to train deep neural networks. The three methods are model-dependent and can modify deep neural networks' parameters. This is different from our method. They are based on data enhancement. Our method is based on input transformation.

3 Ensemble Denoiser Based on Generative Adversarial Networks

In this section, we will describe all the details of our proposed method. First, we will overview the denoiser based on generative adversarial networks. Then, we will introduce its network structure and loss function. Finally, we will integrate these learned mappings as the ultimate defense.

3.1 GAN-based Denoiser

The idea behind the denoiser is to eliminate adversarial examples before they are fed into deep neural networks. Therefore, it is model-independent and cannot modify deep neural networks' parameters. Santhanam et al. [30] have demonstrated that adversarial examples lie outside benign examples' data manifold. Based on this assumption, we can learn a manifold mapper to recover adversarial examples from the adversarial manifold to benign examples from the benign manifold. Therefore, the primary purpose of our proposed denoiser is to learn the mapping between adversarial examples and benign examples. Generative adversarial networks are an excellent way to model data distribution, and there have been some studies [25-26] for recovering adversarial examples to benign examples. Therefore, we employ a generative adversarial network to learn the mapping between adversarial examples and benign examples. The used generative adversarial network is a combination of AC-GAN and WGAN-GP. The classification loss from AC-GAN can enhance the capacity of the generator to recover adversarial examples to benign examples. The loss function in the form of WGAN-GP can ensure the training process is stable. The proposed GAN's architecture is shown in Fig. 1. It consists of a generator and a discriminator. The generator takes adversarial examples as inputs and recovers adversarial examples to benign examples. The discriminator has two purposes. The first is to distinguish adversarial examples from benign examples. The second is to classify all the examples into the correct categories. The generator and the discriminator play against each other. When they reach the nash equilibrium point, the generator does an excellent job of recovering adversarial examples to benign examples.

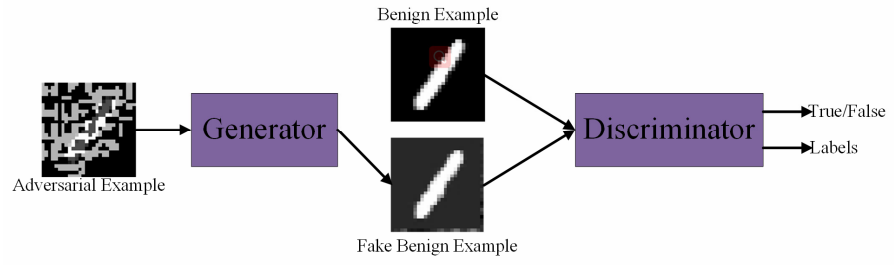


Fig. 1. The architecture of the proposed GAN

3.2 Network Structure and Loss Function

As you can see from Fig. 1, the generator aims to recover adversarial examples to benign examples. UNET is a convolutional encoder-decoder structure, which is good at image-to-image translation. UNET's advantage is that it has a residual structure, making the convolutional encoder-decoder structure deeper. As a general rule, the performance of the deeper network with residual structure is better than that of the external network without residual structure. Therefore, the generator's network structure is based on UNET. This structure can further improve the capacity of the generator to learn the mapping between adversarial examples and benign examples. The generator's network structure is shown in Fig. 2. For the discriminator, the network structure is a simple deep convolution neural network. Because the proposed method is based on the generator, the discriminator's improvement has little effect on improving the proposed method's performance. The discriminator's network structure is shown in Fig. 3. For the proposed GAN's loss function, we make full use of the advantages of AC-GAN and WGAN-GP. First, we add classification loss to the discriminator. The classification loss can enhance the capacity of the generator to recover adversarial examples to benign examples. Second, the loss function in the form of WGAN-GP can ensure the training process is stable. Besides, the total loss function also includes a minimum square error loss, which is calculated as the error between real benign examples and fake benign examples. This can be used to control recovered examples' semantic visual representation. The generator's loss function includes a WGAN-GP loss, a classification loss from AC-GAN, and a minimum square error loss. The discriminator's loss function consists of a WGAN-GP loss and a classification loss from AC-GAN. The total loss function can be expressed as,

$$L=L_{WGAN-GP}+L_{AC-GAN}+L_{MSE} \tag{1}$$

$$L_{AC-GAN}=CrossEntropy\left(Y_{real},Y_{prediction}\right) \tag{2}$$

$$L_{MSE}=\left\|X_{real}-X_{fake}\right\|_2^2 \tag{3}$$

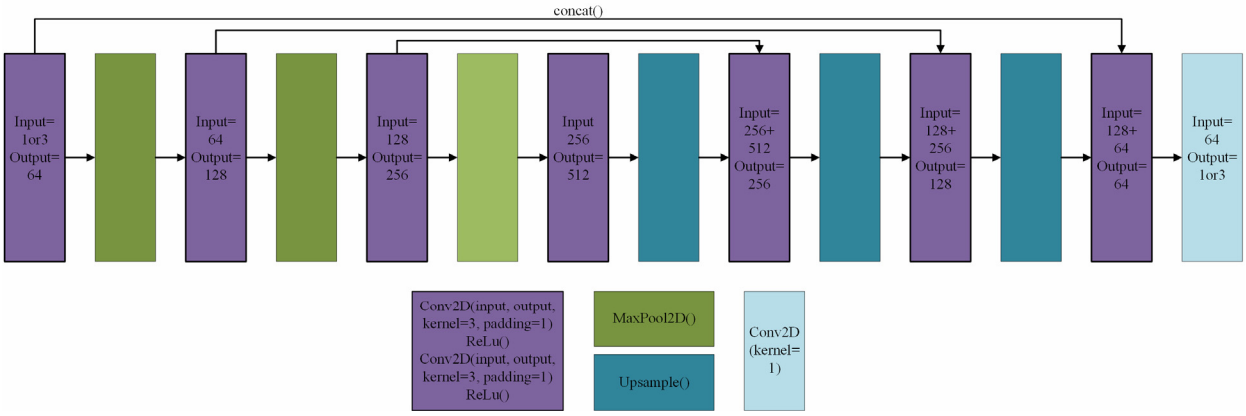


Fig. 2. The network structure of the generator

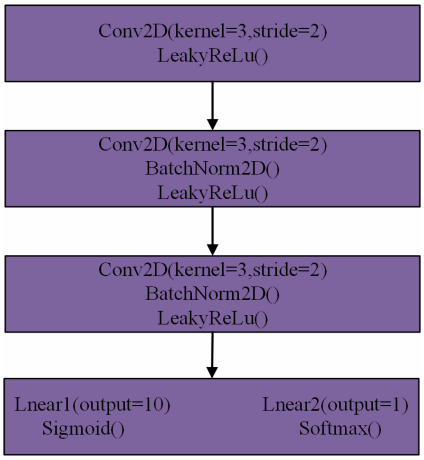


Fig. 3. The network structure of the discriminator

3.3 Ensemble Denoiser

When the generator reaches the nash equilibrium point, it learns multiple mapping, which can recover adversarial examples to benign examples. However, each mapping behaves differently for different types of adversarial examples. Ensemble Learning refers to the process of training multiple learning machines as a committee of decision makers and combining their outputs. Their outputs can be combined in several ways include averaging, voting, and probability, etc. Therefore, we make full use of the advantages of ensemble-based methods. We integrate these mappings as the ultimate method to defend against multiple types of adversarial examples. The used integration strategy is to average all outputs. Fig. 4 is the proposed ensemble denoiser based on generative adversarial networks. Algorithm1 is the pseudo-code of the proposed ensemble denoiser.

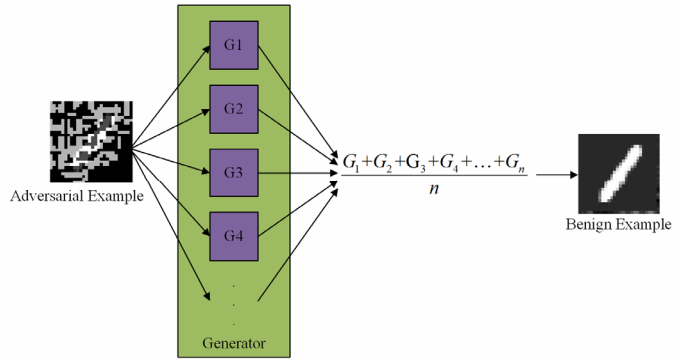


Fig. 4. The proposed ensemble denoiser based on generative adversarial networks

Algorithm 1. GAN-based Ensemble Denoiser

Input: Generator: G
Discriminator: D
Training data: $\left(x_{adversarial}, x_{benign}, y\right)$
Training epochs: N

Output: An ensemble denoiser

1. H = 0
2. For t = 1 to N:
3. $x_{fake\text{-}benign} = G_t\left(x_{adversarial}\right)$
4. $L_{MSE} = \left\| x_{benign} - x_{fake\text{-}benign} \right\|_2^2$
5. $0/1, y_{predicted} = D_t\left(x_{benign}, x_{fake\text{-}benign}\right)$

6. $L_{AC\text{-}GAN} = \text{Crossentropy}\left(y, y_{predicted}\right)$
7. $L_{D_t} = L_{WGAN\text{-}GP} + L_{AC\text{-}GAN}$
8. $D_{t+1} = \text{Adam}\left(D_t, L_{D_t}\right)$
9. $L_{G_t} = L_{WGAN\text{-}GP} + L_{AC\text{-}GAN} + L_{MSE}$
10. $G_{t+1} = \text{Adam}\left(G_t, L_{G_t}\right)$
11. $H = H \cup G_t$
12. $t = t + 1$
13. $G_{1 \text{ to } n} = \text{Select}(H)$
14. Return an ensemble denoiser $\frac{1}{n} G_{1 \text{ to } n}$

4 Experiment and Result Analysis

In this section, we will conduct experiments to verify the proposed method's performance. First, we will briefly describe the experiment settings. Second, we will present the results of all experiments and make a brief analysis of all experimental results. Finally, we will discuss the advantages and disadvantages of the proposed method.

4.1 Experiment Settings

Experiments are conducted on the MNIST and CIFAR10 datasets in the image classification task. The MNIST training data are single-channel greyscale images, and their size is 28x28 pixels. The CIFAR10 training data are three-channel color images, and their size is 32x32 pixels. Fig. 5 is the network structure of the classifier on the MNIST dataset. Its classification accuracy is 99%. Fig. 6 is the network structure of the classifier on the CIFAR10 dataset. Its classification accuracy is 83%. These two classifiers are used to generate adversarial examples and verify the defense method's performance. The adversarial example generation algorithms used in the experiments include FGSM [5], BIM [32], DeepFool [33], JSMA [34], and C&W [35]. They are all state-of-the-art white-box algorithms. This is mainly because black-box algorithms usually take more time and have weaker attack capability. The adversarial example defense methods used in the experiments include APE-GAN [26], Bit Depth [14], TotalVarMin [21], SpatialSmoothing [14], JpegCompression [21], Adversarial Training [5]. They are used to compare with our proposed method. We employ FGSM to generate adversarial examples with various perturbations on the MNIST or CIFAR10 training data. For the MNIST dataset, these perturbations include 0.1, 0.3, 0.5, and 0.7. For the CIFAR10 dataset, these perturbations include 0.01, 0.03, 0.05, and 0.07. Then, we combine these adversarial examples and their corresponding benign examples as the final training data. Fig. 7 shows these generated FGSM adversarial examples on the MNIST and CIFAR10 training data. After this, we train our proposed denoiser by using the final training data. Then, we employ all white-box algorithms to craft adversarial examples with various perturbations on the MNIST or CIFAR10 test data. These generated adversarial examples are used to verify all defense methods' performance. Fig. 8 shows these generated adversarial examples on the MNIST or CIFAR10 test data. The evaluation index used in the experiments is adversarial examples' classification accuracy on the classifier. Our proposed method is model-independent and cannot modify deep neural networks' parameters. Therefore, this evaluation index is sufficient.

4.2 Experiments on the GAN-based Denoiser

We first employ the final training data to train the GAN-based denoiser on the MNIST or CIFAR10 datasets. For the generator, we train a total of 300 epochs and pick the best ten generators. Fig. 9 is the loss curve of the generator and the discriminator on the MNIST and CIFAR10 training data. As you can see from the picture, the proposed GAN training process is highly stable. It has no gradient vanishing and

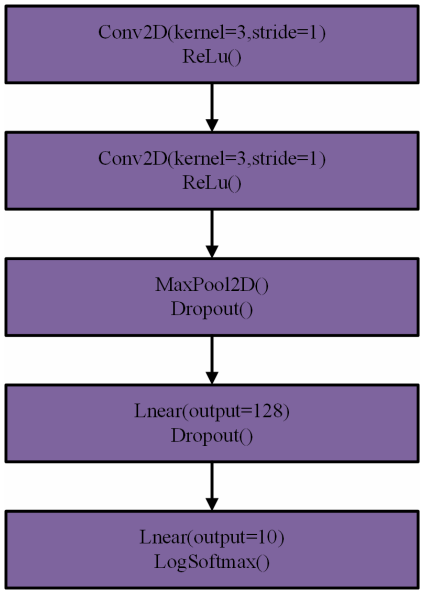


Fig. 5. The network structure of the classification model on MNIST

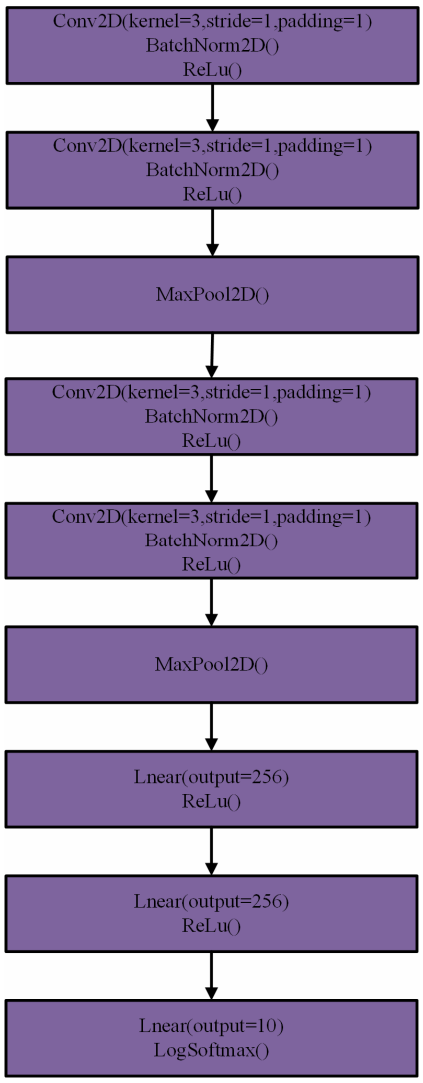
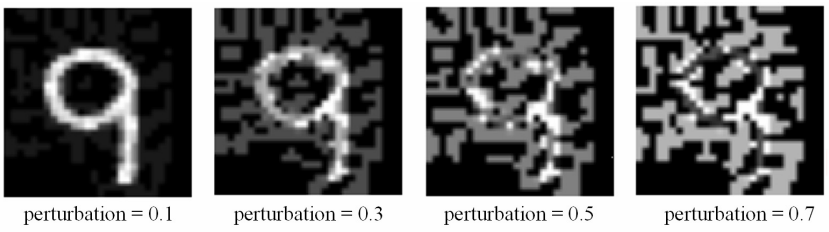
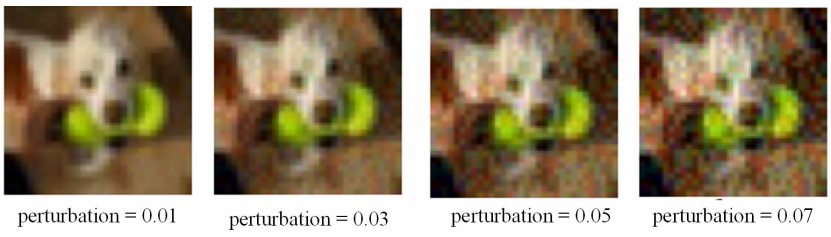


Fig. 6. The network structure of the classification model on CIFAR10

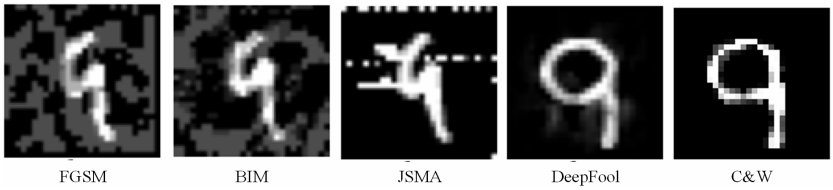


(a) MNIST

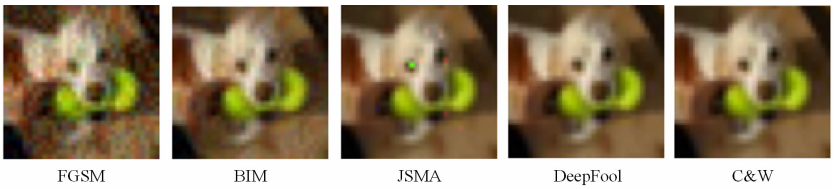


(b) CIFAR 10

Fig. 7. Adversarial examples on the MNIST and CIFAR10 training data



(a) MNIST



(b) CIFAR 10

Fig. 8. Adversarial examples on the MNIST and CIFAR10 test data

gradient explosion problems, and its loss value drops smoothly. Then, we generate adversarial examples to verify the GAN-based denoiser's performance on the MNIST or CIFAR10 test data. FGSM craftsadversarial examples by adding adversarial perturbations to benign examples along the gradient direction of the deep neural network's loss function. BIM is FGSM's variant. It iteratively adds multiple perturbations to benign examples along the gradient direction of the deep neural network's loss function. Therefore, the attack ability of FGSM is more robust than that of BIM. JSMA generates adversarial examples by using the saliency map of deep neural networks. Adversarial examples generated by DeepFool are more similar to benign examples. C&W proposes a specific loss function and uses Adam optimizer to generate adversarial examples iteratively.

Table 1 to Table 3 are the experimental results on the MNIST dataset. Fig. 10 is bar charts from Table 1 to Table 3. Table 1 shows the GAN-based denoiser's performance on the FGSM adversarial examples with various perturbations. Table 2 shows the GAN-based denoiser's performance on the BIM adversarial examples with multiple perturbations. In general, the GAN-based denoiser has advantages

(a) MNIST

(b) CIFAR 10

Fig. 9. The loss curve of the generator and the discriminator on the MNIST and CIFAR10 training data

and disadvantages on both FGSM and BIM adversarial examples. The best performance is achieved when the adversarial perturbation is equal to 0.1. Some generators have a 100% success rate of recovering adversarial examples to benign examples. When the adversarial perturbation is equal to 0.3, the GAN-based denoiser's performance on the BIM adversarial examples is better than that on the FGSM adversarial examples. When the adversarial perturbation is equal to 0.7, the GAN-based denoiser's performance on the FGSM adversarial examples is the same as that in the case of perturbation equal to 0.1. This is mainly because there is a cut when the pixel value is more significant than 255. Except for some exceptional cases, GAN-based denoiser's performance decreases with the increase of perturbation. Table 3 shows the GAN-based denoiser's performance on the JSMA, DeepFool, and C&W adversarial examples. In general, the GAN-based denoiser's performance in Table 3 is significantly weaker than that in Table 1 and Table 2. This is mainly because our training data is based on the FGSM adversarial examples. Theoretically speaking, adversarial examples generated by FGSM are entirely different from those generated by JSMA, DeepFool, and C&W. We can improve the results in Table 3 by adding JSMA, DeepFool, and C&W adversarial examples to the training data. Fig. 11 shows adversarial examples and benign examples recovered by GAN-based denoiser. Fig. 12 shows adversarial examples and benign examples recovered by APE-GAN. As can be seen from Fig. 11 and Fig. 12, benign examples recovered by our proposed method have better visual perception.

Table 1. The GAN-based denoiser's performance (%) on the FGSM adversarial examples with various perturbations

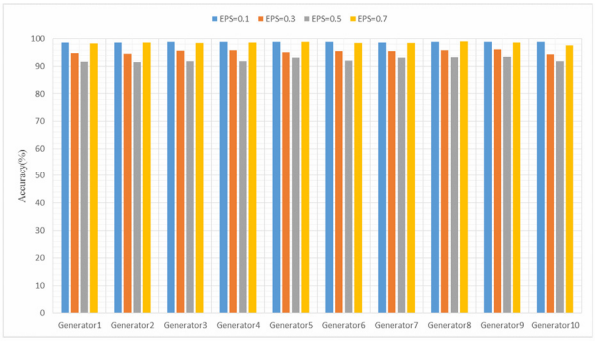
Denoiser	Eps = 0.1	Eps = 0.3	Eps = 0.5	Eps = 0.7
Generator1	98.71	94.75	91.70	98.35
Generator2	98.74	94.52	91.56	98.65
Generator3	98.91	95.72	91.80	98.56
Generator4	98.81	95.86	91.81	98.78
Generator5	98.81	95.13	93.18	98.93
Generator6	98.85	95.50	92.08	98.54
Generator7	98.75	95.42	93.10	98.57
Generator8	98.87	95.77	93.26	98.99
Generator9	98.83	96.14	93.51	98.73
Generator10	98.86	94.41	91.87	97.68

Table 2. The GAN-based denoiser's performance (%) on the BIM adversarial examples with various perturbations

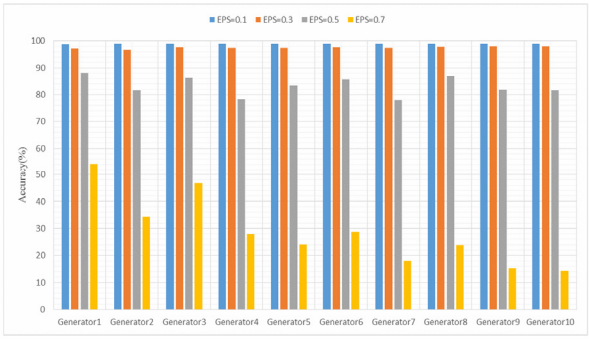
Denoiser	Eps = 0.1	Eps = 0.3	Eps = 0.5	Eps = 0.7
Generator1	98.72	97.12	88.05	54.21
Generator2	98.92	96.67	81.68	34.29
Generator3	98.92	97.52	86.26	46.79
Generator4	98.92	97.38	78.25	27.86
Generator5	98.94	97.36	83.30	24.09
Generator6	99.02	97.66	85.63	28.71
Generator7	98.99	97.49	77.87	18.09
Generator8	98.99	97.78	86.82	23.81
Generator9	99.03	97.92	81.92	15.21
Generator10	99.03	97.97	81.65	14.36

Table 3. The GAN-based denoiser's performance (%) on the JSMA, DeepFool, and C&W adversarial examples

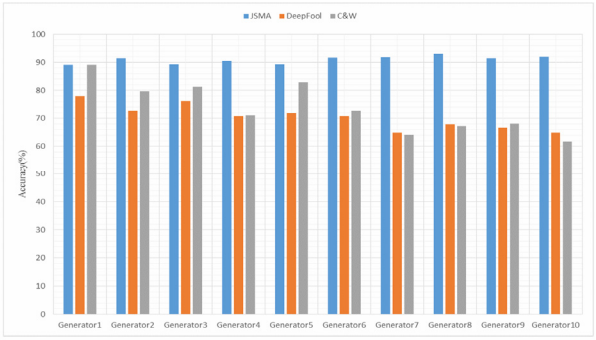
Denoiser	JSMA	DeepFool	C&W
Generator1	89.06	77.98	89.06
Generator2	91.41	72.61	79.69
Generator3	89.16	76.17	81.25
Generator4	90.53	70.80	71.09
Generator5	89.21	71.83	82.81
Generator6	91.60	70.70	72.66
Generator7	91.85	64.89	64.06
Generator8	93.12	67.97	67.19
Generator9	91.41	66.55	67.99
Generator10	91.99	64.84	61.72



(a) FGSM

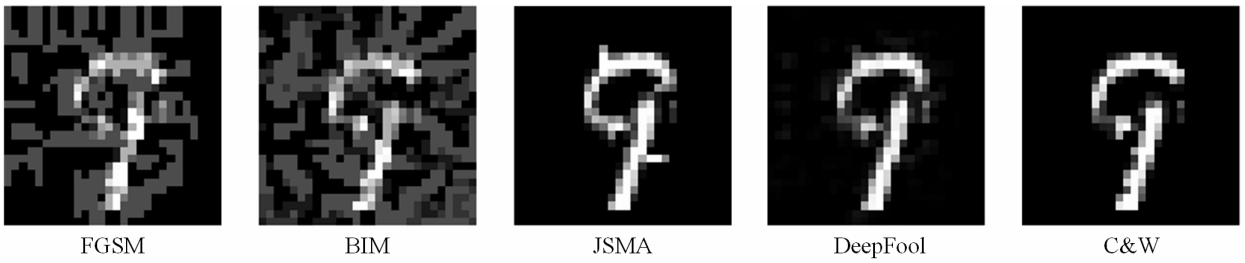


(b) BIM

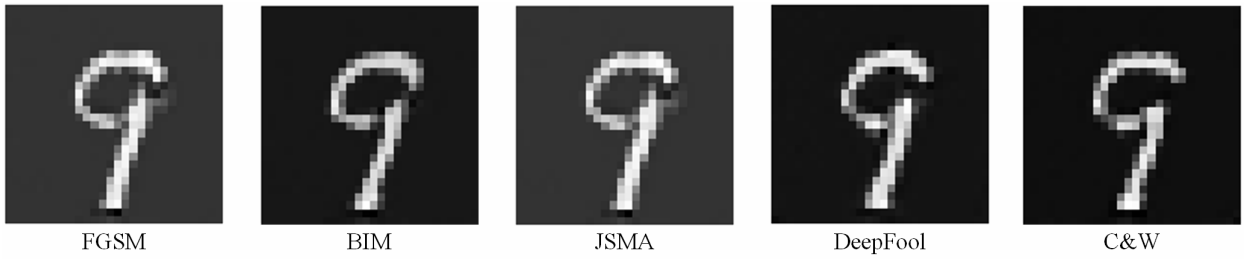


(c) JSMA、DeepFool、C&W

Fig. 10. Bar charts from Table 1 to Table 3

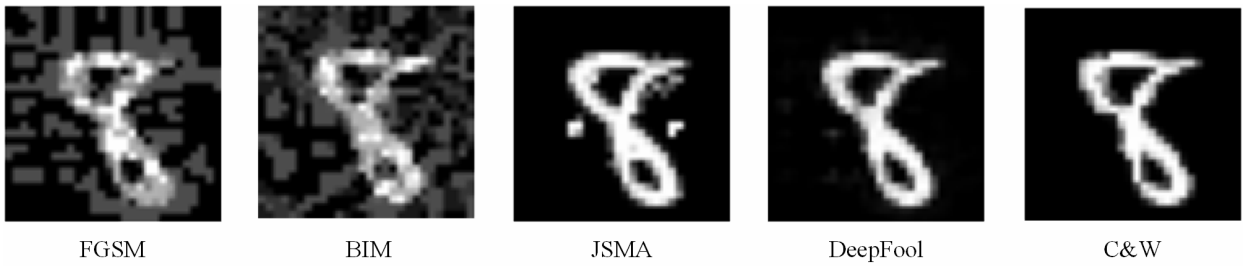


(a) Adversarial Example

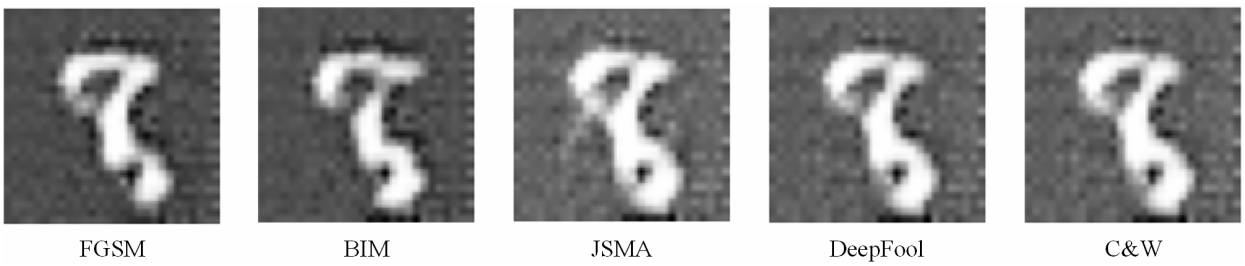


(b) Benign Example

Fig. 11. Adversarial examples and benign examples recovered by GAN-based denoiser



(a) Adversarial Example



(b) Benign Example

Fig. 12. Adversarial examples and benign examples recovered by APE-GAN

Table 4 to Table 6 are the experimental results on the CIFAR10 dataset. Fig. 13 is bar charts from Table 4 to Table 6. The GAN-based denoiser's performance on the CIFAR10 dataset is weaker than that on the MNIST dataset. This is mainly because three-channel color images have more space to generate adversarial examples than single-channel grayscale images. If you want to improve the GAN-based denoiser's performance, one way to do this is to train the denoiser with more adversarial examples. Table 4 shows the GAN-based denoiser's performance on the FGSM adversarial examples with various perturbations. Table 5 shows the GAN-based denoiser's performance on the BIM adversarial examples

with multiple perturbations. In general, the GAN-based denoiser has advantages and disadvantages on both FGSM and BIM adversarial examples. The GAN-based denoiser's performance on the FGSM adversarial examples is better than that of BIM adversarial examples. The best performance is achieved when the adversarial perturbation is equal to 0.01. Generators have a roughly 80% success rate of recovering adversarial examples to benign examples. When the adversarial perturbation is equal to 0.05, the GAN-based denoiser's performance on the FGSM adversarial examples is better than that in the case of perturbation equal to 0.03. This is mainly because there is a cut when the pixel value is more significant than 255. Except for some exceptional cases, GAN-based denoiser's performance decreases with the increase of perturbation. Table 6 shows the GAN-based denoiser's performance on the JSMA, DeepFool, and C&W adversarial examples. The GAN-based denoiser performs well on the DeepFool and C&W adversarial examples. We can improve the GAN-based denoiser's performance on the JSMA adversarial examples by adding JSMA adversarial examples to the training data. Fig. 14 shows adversarial examples and benign examples recovered by GAN-based denoiser. Fig. 15 shows adversarial examples and benign examples recovered by APE-GAN. As can be seen from Fig. 14 and Fig. 15, benign examples recovered by our proposed method have better visual perception.

Table 4. The GAN-based denoiser's performance (%) on the FGSM adversarial examples with various perturbations

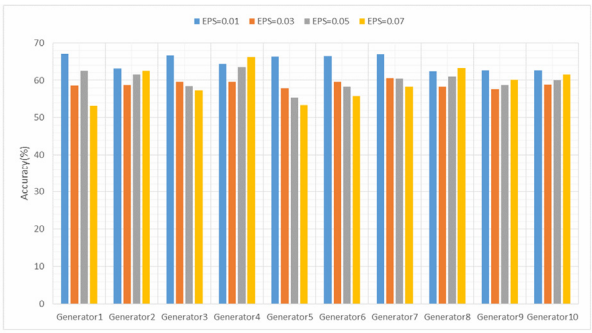
Denoiser	Eps = 0.01	Eps = 0.03	Eps = 0.05	Eps = 0.07
Generator1	66.99	58.61	62.50	53.26
Generator2	63.12	58.79	61.48	62.51
Generator3	66.64	59.59	58.40	57.34
Generator4	64.32	59.64	63.47	66.18
Generator5	66.35	57.84	55.34	53.44
Generator6	66.41	59.53	58.35	55.76
Generator7	66.85	60.57	60.41	58.24
Generator8	62.40	58.31	61.05	63.27
Generator9	62.70	57.60	58.78	60.07
Generator10	62.65	58.81	60.06	61.58

Table 5. The GAN-based denoiser's performance (%) on the BIM adversarial examples with various perturbations

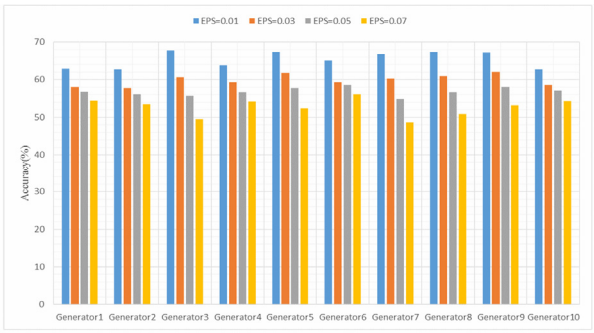
Denoiser	Eps = 0.01	Eps = 0.03	Eps = 0.05	Eps = 0.07
Generator1	62.90	58.03	56.76	54.51
Generator2	62.72	57.83	56.07	53.53
Generator3	67.77	60.71	55.74	49.52
Generator4	63.91	59.29	56.73	54.11
Generator5	67.38	61.72	57.81	52.41
Generator6	65.10	59.23	58.54	56.11
Generator7	66.78	60.21	54.83	48.59
Generator8	67.32	60.91	56.63	50.89
Generator9	67.13	62.04	57.98	53.23
Generator10	62.73	58.63	57.10	54.26

Table 6. The GAN-based denoiser's performance (%) on the JSMA, DeepFool, and C&W adversarial examples

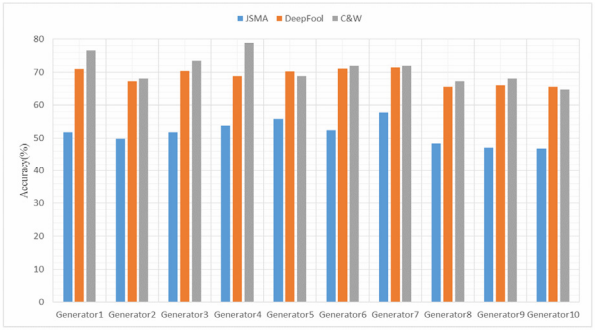
Denoiser	JSMA	DeepFool	C&W
Generator1	51.86	67.33	76.56
Generator2	49.71	70.41	67.97
Generator3	51.71	68.80	73.44
Generator4	53.86	70.21	78.91
Generator5	55.81	71.09	68.75
Generator6	52.39	71.48	71.88
Generator7	57.76	65.63	71.88
Generator8	48.39	65.97	67.19
Generator9	47.12	65.53	67.97
Generator10	46.78	70.41	64.84



(a) FGSM



(b) BIM



(c) JSMA、DeepFool、C&W

Fig. 13. Bar charts from Table 4 to Table 6

(a) Adversarial Example

(b) Benign Example

Fig. 14. Adversarial examples and benign examples recovered by GAN-based denoiser

(a) Adversarial Example

(b) Benign Example

Fig. 15. Adversarial examples and benign examples recovered by APE-GAN

4.3 Experiments on the Ensemble Denoiser

When finishing the training on the generator, we can get multiple mappings, which can recover adversarial examples to benign examples. However, each mapping behaves differently for different types of adversarial examples. Table 1 to Table 6 demonstrates this phenomenon. To defend against multiple types of adversarial examples, we integrate these mappings as the ultimate defense. The used integration strategy is to average all outputs. We craft adversarial examples on the MNIST or CIFAR10 test data to verify all defense methods' performance. We compare the ensemble denoiser's performance with other defense methods' performance. Among the defense methods being compared, APE-GAN, Bit Depth, TotalVarMin, and SpatialSmoothing are model-independent and cannot modify deep neural networks' parameters. Adversarial training is model-dependent and can modify deep neural networks' parameters.

Table 7 to Table 9 are the experimental results on the MNIST dataset. Fig. 16 is line charts from Table 7 to Table 9. Table 7 shows the performance of the ensemble denoiser and other defense methods on the FGSM adversarial examples with various perturbations. Table 8 shows the performance of the ensemble denoiser and other defense methods on the BIM adversarial examples with multiple perturbations. In general, the ensemble denoiser obtains the best performance comparing with other defense methods. The ensemble denoiser can recover adversarial examples with various perturbations to benign examples. Among the defense methods being compared, APE-GAN also has a good performance on the FGSM and BIM adversarial examples. However, APE-GAN's performance is weaker than that of the ensemble denoiser. Adversarial training performed well only on the FGSM adversarial examples. This is mainly because adversarial examples used in our adversarial training only include the FGSM adversarial examples. For the other defense methods, they only performed well against adversarial examples with perturbation equal to 0.1. Table 9 shows the performance of the ensemble denoiser and other defense methods on the JSMA, DeepFool, and C&W adversarial examples. APE-GAN's performance on the JSMA adversarial examples is slightly higher than that of the ensemble denoiser. APE-GAN's performance on the C&W adversarial examples is higher than that of the ensemble denoiser. Except for some exceptional cases, the ensemble denoiser obtains the best performance comparing with other defense methods.

Table 7. The performance (%) of the ensemble denoiser and other defense methods on the FGSM adversarial examples with various perturbations

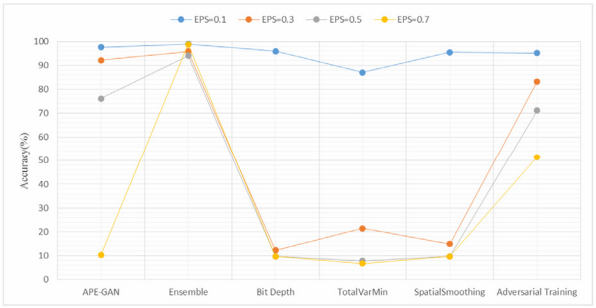
Defense	Eps = 0.1	Eps = 0.3	Eps = 0.5	Eps = 0.7
Non-Defense	90	10	10	10
Ensemble	98.87	95.75	94.00	98.81
APE-GAN	97.51	92.14	76.15	10.31
Bit Depth	95.86	12.28	9.66	9.74
TotalVarMin	86.97	21.44	7.79	6.73
SpatialSmoothing	95.39	14.92	9.69	9.74
Adversarial Training	95.12	83.25	71.16	51.52

Table 8. The performance (%) of the ensemble denoiser and other defense methods on the BIM adversarial examples with various perturbations

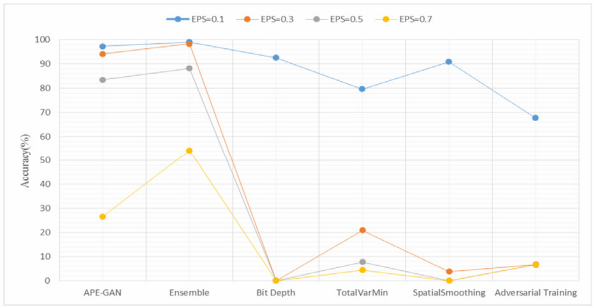
Defense	Eps = 0.1	Eps = 0.3	Eps = 0.5	Eps = 0.7
Non-Defense	41	0	0	0
Ensemble	99.07	98.32	88.23	54.21
APE-GAN	97.25	94.19	83.51	26.57
Bit Depth	92.48	0.00	0.00	0.00
TotalVarMin	79.67	20.90	7.71	4.35
SpatialSmoothing	90.87	3.76	0.00	0.00
Adversarial Training	67.65	6.65	6.60	6.60

Table 9. The performance (%) of the ensemble denoiser and other defense methods on the JSMA, DeepFool, and C&W adversarial examples

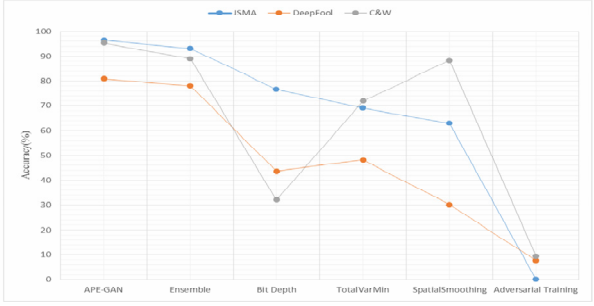
Defense	JSMA	DeepFool	C&W
Non-Defense	0	6	0
Ensemble	95.12	82.08	89.06
APE-GAN	96.37	80.91	95.31
Bit Depth	76.61	43.70	32.03
TotalVarMin	69.04	48.10	71.88
SpatialSmoothing	62.94	30.13	88.28
Adversarial Training	0.00	7.70	9.38



(a) FGSM



(b) BIM



(c) JSMA、DeepFool、C&W

Fig. 16. Line charts from Table 7 to Table 9

Table 10 to Table 12 are the experimental results on the CIFAR10 dataset. Fig. 17 is line charts from Table 10 to Table 12. JpegCompression is also model-independent and cannot modify deep neural networks' parameters. It can only be used to protect three-channel color images. Table 10 shows the performance of the ensemble denoiser and other defense methods on the FGSM adversarial examples with various perturbations. Table 11 shows the performance of the ensemble denoiser and other defense methods on the BIM adversarial examples with multiple perturbations. First, the ensemble denoiser obtains the best performance comparing with other defense methods. Its performance decreases with the increase of perturbation. Second, all the defense methods being compared achieve the best performance when the adversarial perturbation is equal to 0.01. Third, APE-GAN also has a good performance on the FGSM and BIM adversarial examples. However, APE-GAN's performance is weaker than that of the ensemble denoiser. Finally, adversarial training's performance on the CIFAR10 dataset is far inferior to that of the MNIST dataset. This is mainly because three-channel color images have more space to generate adversarial examples than single-channel grayscale images. The number of adversarial examples used to train the classifier is too small. Table 12 shows the performance of the ensemble denoiser and other defense methods on the JSMA, DeepFool, and C&W adversarial examples. The ensemble denoiser obtains the best performance on the DeepFool adversarial examples and achieves the worst performance on the JSMA adversarial examples. Surprisingly, APE-GAN, JpegCompression, and SpatialSmoothing also have an excellent performance. They approximate or even surpass the ensemble denoiser. For instance, JpegCompression's performance on the C&W adversarial examples is higher than that of the ensemble denoiser.

Table 10. The performance (%) of the ensemble denoiser and other defense methods on the FGSM adversarial examples with various perturbations

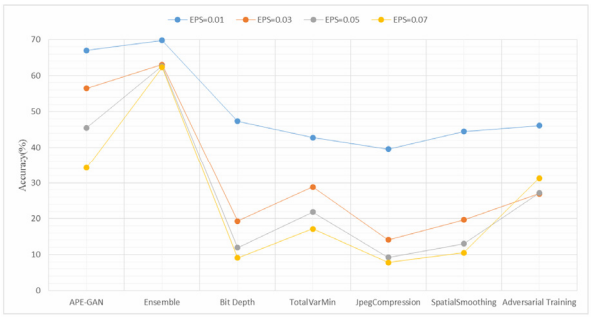
Defense	Eps = 0.01	Eps = 0.03	Eps = 0.05	Eps = 0.07
Non-Defense	28	11	8	7
Ensemble	69.78	63.02	62.67	62.67
APE-GAN	67.00	56.48	45.51	34.27
Bit Depth	47.38	19.33	11.89	9.04
TotalVarMin	42.79	28.84	21.85	17.15
JpegCompression	39.60	14.10	9.24	7.77
SpatialSmoothing	44.49	19.67	13.03	10.51
Adversarial Training	46.13	26.92	27.10	31.27

Table 11. The performance (%) of the ensemble denoiser and other defense methods on the BIM adversarial examples with various perturbations

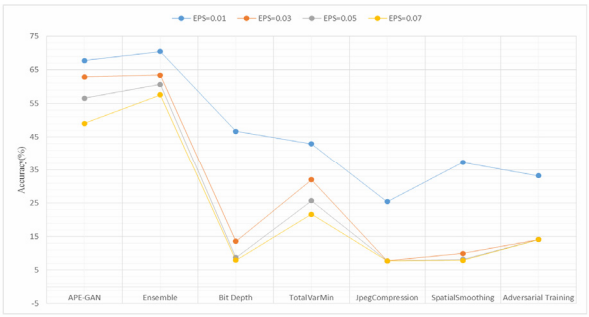
Defense	Eps = 0.01	Eps = 0.03	Eps = 0.05	Eps = 0.07
Non-Defense	11	8	8	8
Ensemble	70.46	63.42	60.64	57.56
APE-GAN	67.74	62.86	56.54	49.09
Bit Depth	46.66	13.57	8.73	7.94
TotalVarMin	42.95	32.00	25.68	21.72
JpegCompression	25.47	7.78	7.78	7.78
SpatialSmoothing	37.31	9.96	8.18	7.85
Adversarial Training	33.20	14.15	14.15	14.15

Table 12. The performance (%) of the ensemble denoiser and other defense methods on the JSMA, DeepFool, and C&W adversarial examples

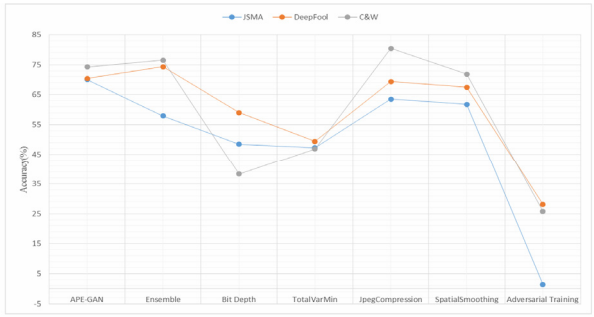
Defense	JSMA	DeepFool	C&W
Non-Defense	1	20	8
Ensemble	57.99	74.32	76.56
APE-GAN	70.07	70.41	74.29
Bit Depth	48.49	59.03	38.28
TotalVarMin	47.31	49.46	46.88
JpegCompression	63.53	69.43	80.47
SpatialSmoothing	61.82	67.53	71.88
Adversarial Training	1.50	28.15	25.78



(a) FGSM



(b) BIM



(c) JSMA、DeepFool、C&W

Fig. 17. Line charts from Table 10 to Table 12

4.4 Discussion

According to the above experimental results, we summarize several advantages and disadvantages of the proposed method. For the benefits, we summarize as follows:

(1) The proposed method is model-independent and cannot modify deep neural networks' parameters. Therefore, it can be easily deployed. Besides, we can combine the proposed method with other defense methods to protect deep neural networks.

(2) The proposed GAN training process is highly stable comparing with APE-GAN. It has no gradient vanishing and gradient explosion problems. This stability ensures that the proposed method can well recover adversarial examples to benign examples.

(3) This proposed method achieves better performance than other methods when defending against multiple types of adversarial examples. Although numerous defense methods do play a significant role in protecting deep neural networks, most of them are only effective for certain types of adversarial examples.

(4) Benign examples recovered by our proposed method have better visual perception. Compared with APE-GAN's shallow structure, UET has deeper network layers. This can enhance the capacity of the generator to recover adversarial examples to benign examples.

For the disadvantages, we summarize as follows:

(1) The proposed method cannot eliminate the effect of adversarial examples with large perturbations. For FGSM and BIM adversarial examples, the proposed method's performance decreases with the increase of perturbation. If you want to improve the experimental results, you can use a deeper network structure.

(2) The proposed method performs poorly on the JSMA, DeepFool, and C&W adversarial examples. This is mainly because our training data is based on the FGSM adversarial examples. You can improve the experimental results by adding JSMA, DeepFool, and C&W adversarial examples to the training data.

5 Conclusion

This paper proposes an ensemble denoiser based on generative adversarial networks to protect deep neural networks. This proposed method makes full use of the advantages of generative adversarial network and ensemble-based methods. First, the used generative adversarial network is a combination of AC-GAN and WGAN-GP. The classification loss from AC-GAN can enhance the capacity of the generator to recover adversarial examples to benign examples. The loss function in the form of WGAN-GP can ensure the training process is stable. Second, the network structure of the generator is based on UNET. This structure can further improve the capacity of the generator to learn the mapping between adversarial examples and benign examples. Third, each mapping behaves differently for different types of adversarial examples. Therefore, we integrate these mappings as the ultimate method to defend against multiple types of adversarial examples. Besides, this paper also has two limitations. First, it doesn't verify the effectiveness of the proposed method on large resolution datasets, such as ImageNet, COCO, etc. This limits the generality of the proposed method. Second, the proposed method has a roughly 80% success rate of recovering adversarial examples to benign examples on the CIFAR10 dataset. This success rate still has room for improvement compared to the nearly 100% success rate on the MNIST dataset. For the second limitation, you can try to improve the performance of the proposed method by increasing the training data. This is a simple and effective improvement. Besides, you can also add a third-party classifier to the proposed GAN architecture. The generator, the discriminator, and the third-party classifier play games with each other. You can explore the effects of third-party classifiers on the generator. This can serve as future research. For further future research, you can explore adversarial example defense methods based on life-long learning, multi-task learning, or reinforcement learning. All of these three learning methods allow the model to discover new knowledge constantly. We qualify the proposed defense method always know how to defend against adversarial examples. Will the performance of the proposed defense method be better? This question remains to be answered in the future.

Acknowledgements

Rui Yang, Tian-Jie Cao, and Fengrong Zhang are supported by the National Natural Science Foundation of China (Grant No. 61972400). Xiu-Qing Chen is supported by China Postdoctoral Science Foundation (Grant No. 2020T130098ZX) and Jiangsu Planned Projects for Postdoctoral Research Funds (Grant No. 1701061B).

References

[1] K. Safari, S. Prasad, D. Labate, A multiscale deep learning approach for high-resolution hyperspectral image classification, IEEE Geoscience and Remote Sensing Letters 18(1)(2020) 167-171.

[2] S. Long, X. He, C. Yao, Scene text detection and recognition: The deep learning era, International Journal of Computer Vision 129(1)(2021) 161-184.

[3] S.S. Mahmoud, A. Kumar, Y. Tang, Y. Li, X. Gu, J. Fu, Q. Fang, An Efficient Deep Learning Based Method for Speech Assessment of Mandarin-Speaking Aphasic Patients, IEEE Journal of Biomedical and Health Informatics 24(11)(2020) 3191-3202.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv:1312.6199 (2013), [Online Available]: https://arxiv.org/abs/1312.6199.

[5] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv:1412.6572 (2014), [Online Available]: https://arxiv.org/abs/1412.6572.

[6] W. Brendel, J. Rauber, M. Bethge, Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, arXiv:1712.04248 (2017), [Online Available]: https://arxiv.org/abs/1712.04248.

[7] J. Su, D.V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Transactions on Evolutionary Computation 23(5)(2019) 828-841.

[8] J. Uesato, B. O'Donoghue, P. Kohli, A. Oord, Adversarial risk and the dangers of evaluating against weak attacks, in: Proc. International Conference on Machine Learning (PMLR), 2018.

[9] Y. Zhang, H. Foroosh, P. David, B. Gong, CAMOU: Learning physical vehicle camouflages to adversarially attack detectors in the wild, in: Proc. International Conference on Learning Representations, 2019.

[10] J.H. Metzen, T. Genewein, V. Fischer, B. Bischoff, On detecting adversarial perturbations, arXiv:1702.04267 (2017), [Online Available]: https://arxiv.org/abs/1702.04267.

[11] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, arXiv:1610.02136 (2016), [Online Available]: https://arxiv.org/abs/1610.02136.

[12] R. Feinman, R.R. Curtin, S. Shintre, A.B. Gardner, Detecting adversarial samples from artifacts, arXiv:1703.00410 (2017), [Online Available]: https://arxiv.org/abs/1703.00410.

[13] T. Pang, C. Du, J. Zhu, Robust deep learning via reverse cross-entropy training and thresholding test, arXiv:1706.00633 (2017), [Online Available]: https://arxiv.org/abs/1706.00633.

[14] W. Xu, D. Evans, Y. Qi, Feature squeezing: Detecting adversarial examples in deep neural networks, 1704.01155 (2017), [Online Available]: https://arxiv.org/abs/1704.01155.

[15] S. Gu, L. Rigazio, Towards deep neural network architectures robust to adversarial examples, arXiv:1412.5068 (2014), [Online Available]: https://arxiv.org/abs/1412.5068.

[16] G.S. Dhillon, K. Azizzadenesheli, Z.C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, A. Anandkumar, Stochastic activation pruning for robust adversarial defense, arXiv:1803.01442 (2018), [Online Available]: https://arxiv.org/abs/1803.01442.

[17] Y. Yang, G. Zhang, D. Katabi, Z. Xu, Me-net: Towards effective adversarial robustness with matrix estimation, arXiv:1905.11971 (2019), [Online Available]: https://arxiv.org/abs/1905.11971.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proc. Advances in neural information processing systems (NIPS), 2014.

[19] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, in: Proc. International conference on machine learning (PMLR), 2017.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein gans, arXiv:1704.00028 (2017), [Online Available]: https://arxiv.org/abs/1704.00028.

[21] C. Guo, M. Rana, M. Cisse, L. Van Der Maaten, Countering adversarial images using input transformations, arXiv:1711.00117 (2017), [Online Available]: https://arxiv.org/abs/1711.00117.

[22] N. Papernot, P. Mcdaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in: Proc. 2016 IEEE symposium on security and privacy (SP), 2016.

[23] D. Meng, H. Chen, Magnet: a two-pronged defense against adversarial examples, in: Proc. Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, 2017.

[24] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Proc. International Conference on Medical image computing and computer-assisted intervention, 2015.

[25] P. Samangouei, M. Kabkab, R. Chellappa, Defense-gan: Protecting classifiers against adversarial attacks using generative models, arXiv:1805.06605 (2018), [Online Available]: https://arxiv.org/abs/1805.06605.

[26] S. Shen, G. Jin, K. Gao, Y. Zhang, Ape-gan: Adversarial perturbation elimination with gan, arXiv:1707.05474 (2017), [Online Available]: https://arxiv.org/abs/1707.05474.

[27] H. Lee, S. Han, J. Lee, Generative adversarial trainer: Defense to adversarial perturbations with gan, arXiv:1705.03387 (2017), [Online Available]: https://arxiv.org/abs/1705.03387.

[28] G. Liu, I. Khalil, A. Khreishah, GanDef: A GAN based adversarial training defense for neural network classifier, in: Proc. IFIP International Conference on ICT Systems Security and Privacy Protection, 2019.

[29] A.S. Hashemi, S. Mozaffari, Secure deep neural networks using adversarial image generation and training with Noise-GAN, Computers and Security 86(2019) 372-387.

[30] G.K. Santhanam, P. Grnarova, Defending against adversarial attacks by leveraging an entire GAN, arXiv:1805.10652 (2018), [Online Available]: https://arxiv.org/abs/1805.10652.

[31] A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in: Proc. International conference on machine learning (PMLR), 2018.

[32] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, arXiv:1611.01236 (2016), [Online Available]: https://arxiv.org/abs/1611.01236.

[33] S. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[34] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: Proc. 2016 IEEE European symposium on security and privacy (EuroS&P), 2016.

[35] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: Proc. 2017 ieee symposium on security and privacy (sp), 2017.