APE-GAN++: An Improved APE-GAN to Eliminate Adversarial Perturbations

Rui Yang, Xiu-Qing Chen, Tian-Jie Cao

Abstract—**Deep neural networks (DNNs) have been deployed successfully in various scenarios, but numerous studies have shown that deep neural networks are vulnerable to the attack of adversarial examples. In order to protect deep neural networks against adversarial examples, a lot of countermeasures have been developed. The APE-GAN is one of these proposed countermeasures, which employ a generative adversarial network (GAN) to eliminate adversarial perturbations. Although it performs more excellently than other countermeasures, it still has some shortcomings. First, its training process is precarious and has a vanishing gradient problem. Second, its performance can be improved further. In this paper, we propose the APE-GAN++, which is an enhanced APE-GAN, to overcome its disadvantages. First, the proposed APE-GAN++ utilizes the WGAN-GP loss to make the training process stable. Then, it uses a newly added third-party classification loss to enhance the capacity of the generator to eliminate adversarial perturbations. Experiments are conducted on the MNIST and CIFAR-10 datasets to verify the proposed APE-GAN++'s performance. Experimental results show that the proposed APE-GAN++ has a stable training process and solves the vanishing gradient problem. Besides, it can also achieve a more excellent performance than other countermeasures when defending against adversarial examples. Experimental code is available at https://github.com/Afreadyang/APE-GAN-Plus-Plus.**

Index Terms—**Adversarial example, Deep neural network, Generative adversarial network, AI security, APE-GAN**

I. Introduction

Adversarial examples [1] have shown that deep neural networks are weak and vulnerable to small and imperceptible perturbations. Although these adversarial perturbations do not affect human perception, they can result in wrong predictions of deep neural networks.

In order to eliminate the effect of adversarial examples, numerous researchers have tried to develop countermeasures to protect deep neural networks. The first is to detect adversarial examples or eliminate adversarial perturbations before they are fed into deep neural networks. The second is to resist adversarial examples by improving the robustness of deep neural networks. The first category includes the adversarial example classifier [2], the statistical analysis [3], the prediction based on density and uncertainty [4], the modification loss [5], and the reconstruction loss [6]. Carlini et al. [7] have evaluated ten adversarial example detectors and found that they can all be defeated. They pointed out that it is challenging to detect adversarial examples as excepted. The second category includes data enhancement [8], regularization [9], randomization [10], and input transformation [11]. Adversarial training [8] is the usual method of data enhancement. Although it is considered the most effective countermeasure in improving the robustness of deep neural networks, it is also time-consuming.

Concerning the input transformation, the manifold-based denoisers are also effective in defending against adversarial examples. The APE-GAN [12] is one of these manifold-based denoisers. The idea behind the APE-GAN is that utilizing a generative adversarial network [27] to eliminate adversarial perturbations. Its architecture consists of a generator and a discriminator, which is a standard GAN framework. Its loss function includes a common GAN loss and a minimum square error loss. Although it performs more excellently than other countermeasures, it still has some shortcomings. First, its training process is volatile and has a vanishing gradient problem. Second, its performance can be improved further.

To overcome these disadvantages, we propose the APE-GAN++, which is an improved APE-GAN. First, we replace the standard GAN loss with the WGAN-GP [28] loss, which is presented to ensure a stable training process. Then, we add a third-party classifier to the architecture of the APE-GAN. The classification loss from the third-party classifier can enhance the performance of the APE-GAN to eliminate adversarial perturbations. Comparing with the APE-GAN, the architecture of the APE-GAN++ includes a generator, a discriminator, and a newly added third-party classifier. Comparing with the APE-GAN, the loss function of the APE-GAN++ consists of a WGAN-GP loss, a minimum square error loss, and a classification loss from the newly added third-party classifier. Experiments are conducted on the MNIST and CIFAR10 datasets, and results show that the proposed APE-GAN++ has a stable training process and solves the vanishing gradient problem. Besides, it can also achieve a more excellent performance than other countermeasures when defending against adversarial examples.

This paper makes the following contributions:

(1) To ensure the training process is stable, the proposed APE-GAN++ replaces the standard GAN loss with the WGAN-GP loss.

(2) To enhance the APE-GAN's performance, the

Manuscript received February 26, 2021; revised July 30, 2021.

This research was partially supported by the China Postdoctoral Science Foundation under Grant 2020T130098ZX and the Jiangsu Planned Projects for Postdoctoral Research Funds under Grant 1701061B.

Rui Yang is a master's student of the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu, China (e-mail: 2119344620@qq.com).

Xiu-Qing Chen is a lecturer of the School of Medicine Information and Engineering, Xuzhou Medical University, Xuzhou, Jiangsu, China (phone: +8615162112100; e-mail: xiuqingchen@126.com).

Tian-Jie Cao is a professor of the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu, China (e-mail: tjcao@cumt.edu.cn).

proposed APE-GAN++ adds a third-party classifier to the APE-GAN's architecture.

The rest of the paper will be organized as follows. Section II is a brief overview of adversarial example defenses. Then, in section III, we describe all details of the proposed APE-GAN++. Experiments are conducted in Section V. The conclusion is offered in Section VI.

II. RELATED WORK

In this section, we will briefly review these existing adversarial example defenses, especially GAN-based methods.

A. Adversarial Example Defenses

Goodfellow et al. [13] proposed adversarial training to enhance deep learning models' robustness, which is considered the most effective countermeasure at present. Guo et al. [14] applied the total variance minimization and the image quilting to adversarial images to eliminate adversarial perturbations. Buckman et al. [15] proposed a countermeasure based on deep neural networks' high-dimensional linearity. It can eradicate adversarial perturbations by quantizing and discretizing the input data of deep neural networks. Yang et al. [16] introduced a preprocessing-based countermeasure named ME-Net. It can drop adversarial images and reconstruct benign images. Xu et al. [6] argued that the dimensionality of input features is often unnecessarily large. They proposed two feature squeezing methods to eliminate adversarial perturbations. Metzen et al. [2] proposed a pre-trained binary classifier to detect adversarial examples. Feinman et al. [14] suggested the kernel density and bayesian uncertainty estimation perform malicious detection.

B. GAN-based Adversarial Example Defenses

Samangouei et al. [17] proposed the Defense-GAN to recovery adversarial examples to benign examples. Its defense strategy consists of two steps. First, it is trained to learn benign examples' distribution. Second, the generator takes adversarial examples as inputs and finds close outputs that are not adversarial. Shen et al. [12] proposed the APE-GAN to eliminate adversarial perturbations. Its defense strategy is a standard GAN training process. The generator takes adversarial examples as inputs and is trained to remove adversarial perturbations. The discriminator is employed to distinguish between original benign examples and fake benign examples. Lindqvist et al. [19] introduced the Auto-GAN, which can ease adversarial attacks by projecting perturbed data points into a benign lower-dimensional manifold. Lee et al. [18] combined a generative adversarial network with adversarial training to improve deep neural networks' robustness. It includes a generator and a deep neural network. The generator is employed to craft adversarial examples. They alternately trained both the deep neural network and the generator network, enhancing the deep neural network's robustness. Liu et al. [20] designed a GAN-based adversarial training. It utilizes a competition game to regulate the feature selection during the training process. Hashemi et al. [21] proposed the Noise-GAN to protect deep neural networks against adversarial attacks. Its defense strategy is based on adversarial training. It has a multi-class discriminator, which is employed to generate various adversarial perturbations.

III. APE-GAN++: AN IMPROVED APE-GAN

In this section, we will describe all details of the proposed APE-GAN++. First, we will overview the idea behind the APE-GAN++. Then, we will introduce its network structure and loss function.

A. The Idea behind APE-GAN++

Santhanam et al. [22] have demonstrated that adversarial examples lie outside the manifold region of benign examples. Therefore, we can utilize a manifold mapping to project adversarial examples into the manifold region of benign examples. The APE-GAN is one of these manifold projectors. The APE-GAN can eliminate adversarial perturbations before they are fed into classification models. Although it performs more excellently than other countermeasures, it still has some shortcomings. First, its training process is precarious and has a vanishing gradient problem. Second, its performance can be improved further. To overcome these limitations, we proposed the APE-GAN++, which is shown in Fig.1. The architecture of the APE-GAN++ consists of a generator, a discriminator, and a third-party classification model. The generator takes adversarial examples as inputs and is trained to remove adversarial perturbations. The discriminator is used to distinguish between original benign examples and fake benign examples. It is a standard GAN training process. The APE-GAN++ replaces the initial GAN loss with the WGAN-GP loss to ensure the training process is stable. Besides, the third-party classification model is also employed to guide the training of the APE-GAN++. The third-party classification model can promote the stability of the APE-GAN++ and improve the capacity of the generator to eliminate adversarial perturbations.

B. Network Structure and Loss Function

As you can see from Fig.1, the generator's purpose is to remove adversarial perturbations. Therefore, the generator takes adversarial examples as inputs and outputs fake normal examples. In the APE-GAN++'s training process, the generator takes various adversarial examples instead of just one kind of adversarial example as inputs, different from the APE-GAN. This modification can make the APE-GAN++ defend against various adversarial examples. The generator's network structure adopts convolution encoder-decoder architecture, as shown in Fig.2 (a). Its loss function includes a WGAN-GP loss, a minimum square error loss, and a classification loss from the third-party classification model. The minimum square error loss is calculated as the error between real benign examples and fake benign examples. The classification loss from the third-party classification model refers to the error of the third-party classification model on the fake benign examples. The discriminator takes real benign examples and fake benign examples as inputs and distinguishes which one is real or fake. The discriminator's network structure adopts a simple convolution neural network, as shown in Fig.2 (b). Its loss only includes a WGAN-GP loss. The total loss functions can be expressed as:

$$L_{\text{APE-GAN++}} = L_{\text{WGAN-GP}} + \lambda_1 L_{\text{MSE}} + \lambda_2 L_{\text{CLC}} \quad (1)$$

$$L_{\text{MSE}} = \left\| X_{\text{real-benign}} - X_{\text{fake-benign}} \right\|^2 \quad (2)$$

$$L_{\text{CLC}} = \text{CrossEntropy}\left(Y_{\text{real-benign}} - Y_{\text{fake-benign}} \right) \quad (3)$$

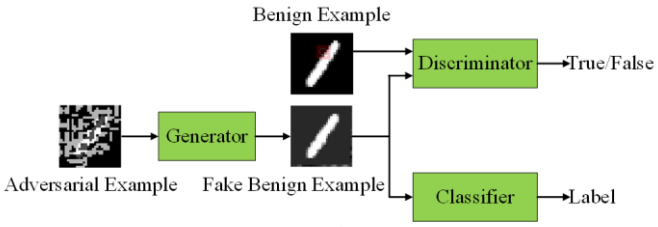
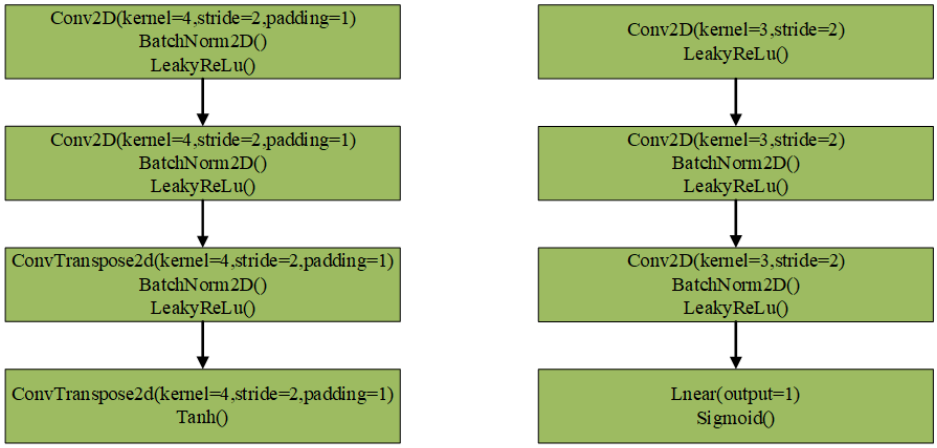


Fig.1.The architecture of the APE-GAN++.

IV. Experiments and Results Analysis

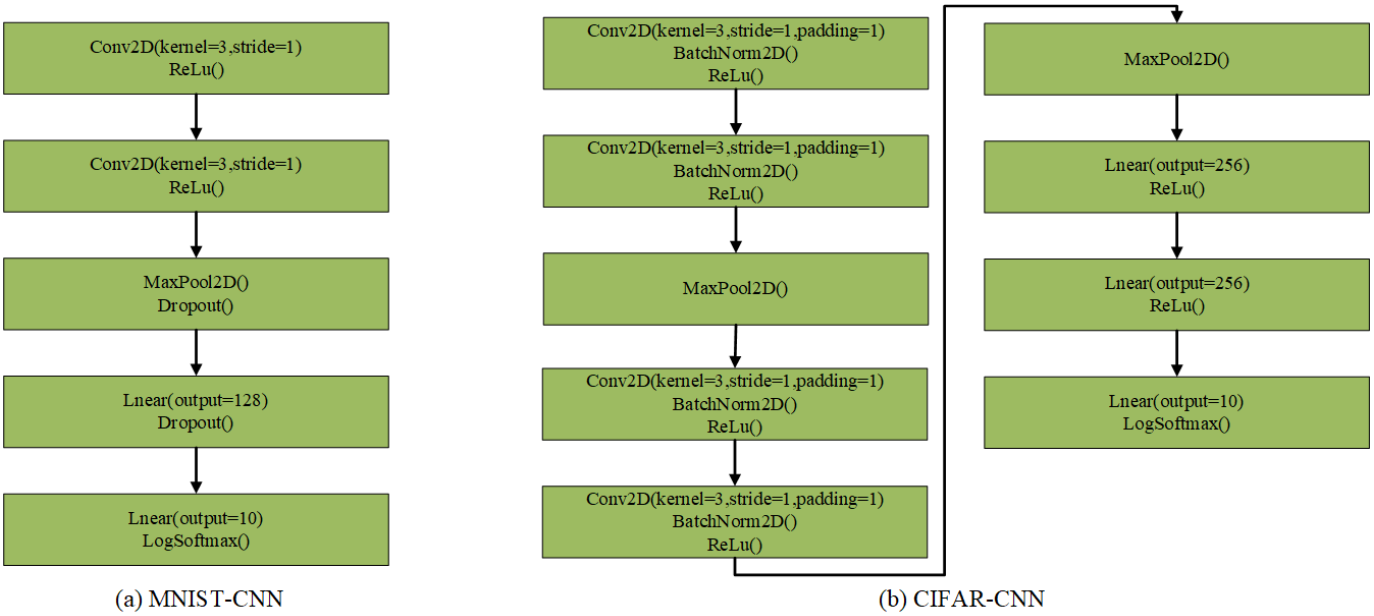
A. Experiment Settings

We conduct experiments on the MNIST and CIFAR-10 datasets in the image classification task. MNIST, which is available at http://yann.lecun.com/exdb/mnist/, is a handwritten digit dataset from 0 to 9. It contains 60000 training images and 10000 testing images. They are single-channel greyscale images, and their size is 28x28 pixels. CIFAR-10, which is available at https://www.cs.toronto.edu/~kriz/cifar.html, is a color image dataset in 10 classes. It contains 50000 training images and 10000 testing images. They are three-channel color images, and their size is 32x32 pixels. We utilize the FGSM algorithm 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 CIFAR-10 dataset, these perturbations include 0.01, 0.03, 0.05, and 0.07. Then, these generated adversarial examples are employed to train the APE-GAN++. For the MNIST dataset, the classification model includes MNIST-CNN, ResNet18 [29], VGG16 [30], LeNet [31]. For the CIFAR-10 dataset, the classification model includes CIFAR-CNN, ResNet18, VGG16, DenseNet [32]. MNIST-CNN and CIFAR-CNN are also used to generate adversarial examples due to their simple network structure. They are shown in Fig.3. For the APE-GAN++'s loss function, we set λ_1 equal to 0.5 and λ_2 equal to 0.5. The measure used in the experiment is the classification accuracy of examples on classification models. Table I shows the classification accuracy of benign examples on classification models. As can be seen from Table I, their classification accuracy is quite high. The used adversarial example generation algorithm inc-



(a) Generator (b) Discriminator

Fig.2.The network structure of the generator and the discriminator.



(a) MNIST-CNN (b) CIFAR-CNN

Fig.3.The network structure of MNIST-CNN and CIFAR-CNN.

ludes FGSM [13], BIM [23], DeepFool [24], JSMA [25], and C&W [26]. Table II shows the classification accuracy of adversarial examples on classification models. As can be seen from Table II, their performance is strong. The used adversarial example defense algorithm includes APE-GAN [12], Bit Depth [6], TotalVarMin [14], SpatialSmoothing [6], Adversarial Training [13]. Table III-VI show the classification accuracy of adversarial examples on the MNIST and CIFAR-10 datasets with the defense mentioned above. These results are employed to compare with the APE-GAN++.

B. Experiments

We utilize generated adversarial examples to train the APE-GAN++ with various third-party classification models. After that, the APE-GAN++ is used to defend against adversarial examples generated on different classification models. We demonstrated that the APE-GAN++ has a stable training process and solves the vanishing gradient problem. Besides, it also achieves a better comprehensive performance than other defenses include the APE-GAN.

1. APE-GAN++'s Stability

For the APE-GAN++'s stability, we record the generator and the discriminator's loss values with 200 epochs during the training process. Figure.4 and Figure.5 show the generator and the discriminator's loss curve with various third-party classification models. Figure.4 (a) is on the MNIST dataset. Figure.5 (b) is on the CIFAR-10 dataset. For the MNIST dataset, these loss curves are nearly perfect. After the training process with 200 epochs, the generator and the discriminator's loss tend to converge. For the CIFAR-10 dataset, the generator's loss curves are a little shaky during the training process. However, it is still stable. If we increase the number of training rounds, the generator and the discriminator's loss will converge. As shown in Fig.4, we can demonstrate that the APE-GAN++ has a stable training process and solves the vanishing gradient problem.

2. APE-GAN++'s Performance

Figure.6 to 13 are the classification accuracy curves of adversarial examples on the MNIST and CIFAR-10 datasets with the APE-GAN++. Each small graph represents adversarial examples generated on various classification models. Each large graph represents the APE-GAN++ trained with different third-party classification models. For the MNIST dataset, the classification model includes MNIST-CNN, ResNet18, VGG16, LeNet. For the CIFAR-10 dataset, the classification model includes CIFAR-CNN, ResNet18, VGG16, DenseNet. As you can see from Fig.6 to 13, the classification accuracy curves are nearly stable, especially on the MNIST dataset. It also shows that APE-GAN++ has a steady training process. We can't compare APE-GAN here because it doesn't even have 50 training epochs. In Fig.6 to 9, the APE-GAN++ trained with MNIST-CNN, VGG16, and LeNet cannot defend well against JSMA adversarial examples. The APE-GAN++ trained with ResNet18 cannot defend well against DeepFool adversarial examples. In Fig.10 to 13, the APE-GAN++ trained with MNIST-CNN and ResNet18 cannot defend well against FGSM and BIM adversarial examples generated on ResNet18, VGG16, and DenseNet. It is mainly because we only use FGSM adversarial examples as the training data. In addition, differences in the network structure of classification models lead to this phenomenon.

Table III-IV show the classification accuracy of adversarial examples on the MNIST dataset with the defense algorithm mentioned in the experiment settings. Table V-VI show the classification accuracy of adversarial examples on the CIFAR-10 dataset with the defense algorithm mentioned in the experiment settings. These adversarial examples are generated on various classification models. From Table III-VI, we can find that the APE-GAN's overall performance is better than that of other countermeasures. Although the performance of some defense algorithms is better than that of the APE-GAN on some types of adversarial examples, they cannot defend well against all types of adversarial examples. For example, the APE-GAN can defend well against all types of adversarial examples generated on ResNet18. Bit Depth can only defend well against FGSM and BIM adversarial examples. Besides, the performance of the APE-GAN on the CIFAR-10 dataset can be further improved.

Table VII-VIII show the classification accuracy of adversarial examples on the MNIST dataset with the APE-GAN++. Table IX-X offer the classification accuracy of adversarial examples on the CIFAR-10 dataset with the APE-GAN++. These adversarial examples are generated on various classification models. The APE-GAN++ is trained with multiple third-party classification models. From Table VII-X, we can find that APE-GAN++'s overall performance is better than that of other countermeasures. However, there are still some shortcomings in the performance of the APE-GAN++. These shortcomings have been mentioned above. We will make a detailed comparative analysis of Table III-VI and Table VII-X in the following part of the result analysis.

C. Result Analysis

1. Visualization Analysis

Figure.14, Figure.15, Figure.16, and Figure.17 are the visualizations of adversarial examples recovering from the APE-GAN++. As shown in Fig.14 to 15, they are on the MNIST dataset. Figure.14 (a) represents the APE-GAN++ trained with MNIST-CNN. Figure.14 (b) illustrates the APE-GAN++ trained with ResNet18. Figure.15 (c) represents the APE-GAN++ trained with VGG16. Figure.15 (d) represents the APE-GAN++ trained with LeNet.

For each part, the first column represents adversarial examples generated on MNIST-CNN. The second column represents adversarial examples generated on ResNet18. The third column represents adversarial examples generated on VGG16. The last column represents adversarial examples generated on LeNet. As you can see from Fig.14 to 15, these adversarial perturbations are visible to human eyes. The APE-GAN++ can eliminate these adversarial perturbations very well and maintain examples' visual perception.

As shown in Fig.16 to 17, they are on the CIFAR-10 dataset. Figure.16 (a) represents the APE-GAN++ trained with CIFAR-CNN. Figure.16 (b) illustrates the APE-GAN++ trained with ResNet18. Figure.17 (c) represents the APE-GAN++ trained with VGG16. Figure.17 (d) represents the APE-GAN++ trained with DenseNet. For each part, the first column represents adversarial examples generated on CIFAR-CNN. The second column represents adversarial examples generated on ResNet18. The third column represents adversarial examples generated on VGG16. The last column represents adversarial examples generated on DenseNet. As you can see from Fig.16 to 17, these

adversarial perturbations are small, except for JSMA adversarial examples. The APE-GAN++ can eliminate these adversarial perturbations very well, and their visual perception is nearly perfect.

Figure.18 is the visualization of adversarial examples recovering from the APE-GAN. Figure.18 (a) is on the MNIST dataset. Figure.18 (b) is on the CIFAR-10 dataset. As you can see from Fig.18, their visual perception is poor comparing with Fig.14 to 15 and Fig.16 to 17. The APE-GAN focuses on recovering adversarial examples' semantic information. They ignore adversarial examples' local information, such as the background. On the contrary, the APE-GAN++ focuses on recovering not only semantic information but also regional information. That explains why the APE-GAN++ outperforms the APE-GAN.

2. Comparative Analysis

Figure.19 to 22 are the line chart of adversarial examples under various defense algorithms. As shown in Fig.19 to 20, it is on the MNIST dataset. As shown in Fig.21 to 22, it is on the CIFAR-10 dataset. Figure.19 (a), Figure.20 (a), Figure.21 (a), and Figure.22 (a) are the defense algorithm mentioned in the experiment settings. Figure.19 (b), Figure.20 (b), Figure.21 (b), and Figure.22 (b) are the APE-GAN++ trained with various third-party classification models. Adversarial examples are crafted on multiple classification models.

As you can see from Fig.19(a), Fig.20 (a), Fig.21 (a), and Fig.22 (a), the shortcomings of the defense algorithm mentioned in the experiment settings, except for the APE-GAN, are apparent. They cannot defend well against all types of adversarial examples. They only perform well only on some specific types of adversarial examples. That explains why Fig.19 (a) and Fig.20 (a) are so shaky. In this respect, the APE-GAN and the APE-GAN++ outperform them. Although they do have a few exceptions, the APE-GAN and the APE-GAN++ defend well against all types of adversarial examples. These exceptions have been mentioned above. In general, the APE-GAN and the APE-GAN++ perform better than the defense algorithm mentioned in the experiment settings, except for the APE-GAN. This indicates that the APE-GAN++ has better overall performance than other defenses when defending against various adversarial examples.

Next, we compared the performance of the APE-GAN and the APE-GAN++. The APE-GAN++'s advantage is that adversarial examples recovering from the APE-GAN++ have a higher classification accuracy than the APE-GAN. As you can see from Fig.19 (b) and Fig.20 (b), the APE-GAN++ can recovery adversarial examples with a nearly 100% success rate, except for a few exceptions. These exceptions can be solved by increase training data. As you can see from Fig.21 (b) and Fig.22 (b), the performance of the APE-GAN++ trained with CIFAR-CNN is so encouraging. Its overall performance is ultimately stronger than that of the APE-GAN. Although the APE-GAN++ trained with other third-party classification models has better performance, it cannot defend well against FGSM and BIM adversarial examples. In general, the APE-GAN++ performs better than the APE-GAN. This shows that the APE-GAN++ has better overall performance than the APE-GAN when defending against various adversarial examples.

V. Conclusion

In this paper, we propose an improved APE-GAN, named APE-GAN++. First, the APE-GAN++ is based on WGAN-GP, which makes the APE-GAN's training process stable. Then, we add a third-party classification model to the standard GAN architecture. The additional classification model can guide the APE-GAN's training process and enhance the generator's ability to eliminate adversarial perturbations. Experiments are conducted on the MNIST and CIFAR-10 datasets. Experimental results demonstrate that the training process is stable, and the APE-GAN++ achieves better performance than other defenses when defending against various adversarial examples. In the future, we will explore how to utilize the APE-GAN++ to defend against adversarial examples in the speech field.

TABLE I
THE CLASSIFICATION ACCURACY (%) OF BENIGN EXAMPLES ON CLASSIFICATION MODELS.

	MNIST				CIFAR-10			
	MNIST-CNN	ResNet18	VGG16	LeNet	CIFAR-CNN	ResNet18	VGG16	DenseNet
Benign samples	98.51	99.65	99.66	99.02	83.98	87.43	89.91	92.58

TABLE II
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON CLASSIFICATION MODELS.

	MNIST				CIFAR-10			
	MNIST-CNN	ResNet18	VGG16	LeNet	CIFAR-CNN	ResNet18	VGG16	DenseNet
FGSM	10.57	9.63	30.13	31.57	10.16	11.69	41.00	37.60
BIM	1.00	0.33	0.29	1.25	10.06	9.02	15.46	11.75
JSMA	54.83	0.03	0.05	1.42	1.70	1.27	1.09	1.03
DeepFool	51.32	5.86	0.41	1.23	43.10	20.91	8.99	9.52
C&W	70.31	0.39	0.00	1.17	8.59	7.03	7.03	3.91

TABLE III
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON THE MNIST DATASET UNDER DEFENSE ALGORITHMS MENTIONED IN THE EXPERIMENT SETTINGS.

	MNIST-CNN					ResNet18				
	FGSM	BIM	JSMA	DeepFool	C&W	FGSM	BIM	JSMA	DeepFool	C&W
APE-GAN	74.15	71.29	89.19	90.10	88.28	89.97	91.48	93.45	80.04	85.94
Bit Depth	81.75	77.03	64.80	95.94	75.09	92.79	95.65	64.99	44.25	16.02
TotalVarMin	34.48	14.29	70.57	83.03	81.64	21.39	22.19	71.93	47.70	72.66
SpatialSmoothing	29.24	6.05	80.07	82.67	89.45	20.78	20.43	95.22	57.28	87.89
Adversarial Training	68.39	9.02	64.55	69.35	91.00	70.59	14.47	68.52	72.63	89.26

TABLE IV
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON THE MNIST DATASET UNDER DEFENSE ALGORITHMS MENTIONED IN THE EXPERIMENT SETTINGS.

	VGG16					LeNet				
	FGSM	BIM	JSMA	DeepFool	C&W	FGSM	BIM	JSMA	DeepFool	C&W
APE-GAN	88.89	90.64	92.03	95.10	92.58	62.88	42.64	56.03	82.20	55.47
Bit Depth	91.26	91.45	54.36	98.17	34.77	83.59	81.46	49.98	96.38	20.70
TotalVarMin	52.69	40.84	65.81	88.55	76.17	34.79	2.18	42.74	81.23	56.25
SpatialSmoothing	42.88	26.83	80.43	95.24	90.63	38.30	17.43	47.12	78.20	81.25
Adversarial Training	71.46	13.72	67.31	73.58	90.77	74.59	14.47	68.52	72.63	89.26

TABLE V
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON THE CIFAR-10 DATASET UNDER DEFENSE ALGORITHMS MENTIONED IN THE EXPERIMENT SETTINGS.

	CIFAR-CNN					ResNet18				
	FGSM	BIM	JSMA	DeepFool	C&W	FGSM	BIM	JSMA	DeepFool	C&W
APE-GAN	58.25	62.94	67.21	68.59	69.14	56.48	65.31	68.67	69.7	71.09
Bit Depth	22.52	28.67	46.83	79.64	82.42	21.43	30.02	49.39	64.92	79.30
TotalVarMin	35.10	42.36	48.76	53.71	53.13	28.98	36.81	45.39	48.40	50.39
SpatialSmoothing	32.87	40.42	69.65	73.29	79.69	27.15	37.15	74.50	66.02	73.83
Adversarial Training	59.23	49.65	29.9	39.50	32.50	64.79	54.63	34.14	44.15	37.81

TABLE VI
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON THE CIFAR-10 DATASET UNDER DEFENSE ALGORITHMS MENTIONED IN THE EXPERIMENT SETTINGS.

	VGG16					DenseNet				
	FGSM	BIM	JSMA	DeepFool	C&W	FGSM	BIM	JSMA	DeepFool	C&W
APE-GAN	65.95	67.35	69.21	65.97	71.88	65.23	64.51	68.16	68.12	69.92
Bit Depth	42.22	37.07	47.98	34.85	71.48	42.82	39.48	50.39	68.70	78.13
TotalVarMin	45.53	45.47	50.76	44.27	57.03	42.19	42.74	45.36	51.66	55.08
SpatialSmoothing	48.29	46.54	75.14	47.57	76.95	47.90	43.93	82.03	74.41	80.86
Adversarial Training	66.16	56.92	36.47	46.39	39.83	69.34	59.52	39.70	49.6	42.55

TABLE VII
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON THE MNIST DATASET UNDER THE APE-GAN++.

	MNIST-CNN					ResNet18				
	FGSM	BIM	JSMA	DeepFool	C&W	FGSM	BIM	JSMA	DeepFool	C&W
MNIST-CNN	97.53	96.68	76.13	98.17	98.83	94.78	94.66	86.49	98.03	98.05
ResNet18	97.47	98.53	93.85	83.59	96.48	96.45	98.39	95.14	82.77	95.70
VGG16	97.52	97.55	83.86	99.22	97.66	96.35	96.97	86.77	99.15	98.44
LeNet	93.86	94.11	50.69	98.63	89.06	90.00	97.03	53.50	98.65	92.19

TABLE VIII
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON THE MNIST DATASET UNDER THE APE-GAN++.

	VGG16					LeNet				
	FGSM	BIM	JSMA	DeepFool	C&W	FGSM	BIM	JSMA	DeepFool	C&W
MNIST-CNN	95.06	94.89	85.97	97.91	98.44	95.27	94.74	84.35	98.10	98.83
ResNet18	96.47	98.41	94.34	79.84	95.31	96.53	98.28	90.66	73.42	95.53
VGG16	96.50	96.74	85.92	99.12	98.05	96.64	96.91	84.86	99.13	97.66
LeNet	89.46	90.73	54.19	98.65	91.79	90.68	91.25	49.04	98.77	91.80

TABLE IX
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON THE CIFAR-10 DATASET UNDER THE APE-GAN++.

	CIFAR-CNN					ResNet18				
	FGSM	BIM	JSMA	DeepFool	C&W	FGSM	BIM	JSMA	DeepFool	C&W
CIFAR-CNN	74.20	73.76	72.99	76.48	82.03	56.95	65.83	74.47	79.61	81.64
ResNet18	68.17	73.39	73.45	77.98	80.47	57.39	69.06	78.27	81.77	85.94
VGG16	73.42	75.85	72.32	75.80	82.03	66.71	75.49	77.96	71.04	85.16
DenseNet	74.61	76.16	75.05	81.35	83.98	70.67	76.78	82.32	85.69	88.67

TABLE X
THE CLASSIFICATION ACCURACY (%) OF ADVERSARIAL EXAMPLES ON THE CIFAR-10 DATASET UNDER THE APE-GAN++.

	VGG16					DenseNet				
	FGSM	BIM	JSMA	DeepFool	C&W	FGSM	BIM	JSMA	DeepFool	C&W
CIFAR-CNN	59.23	66.44	74.28	79.03	83.59	56.93	65.74	73.79	78.78	83.98
ResNet18	51.74	68.94	77.42	81.11	85.16	51.09	68.48	77.31	80.95	84.77
VGG16	68.35	75.64	77.49	71.72	85.55	67.26	75.67	77.35	70.67	84.38
DenseNet	70.81	76.02	81.20	85.38	87.89	71.63	77.27	81.34	85.35	89.06

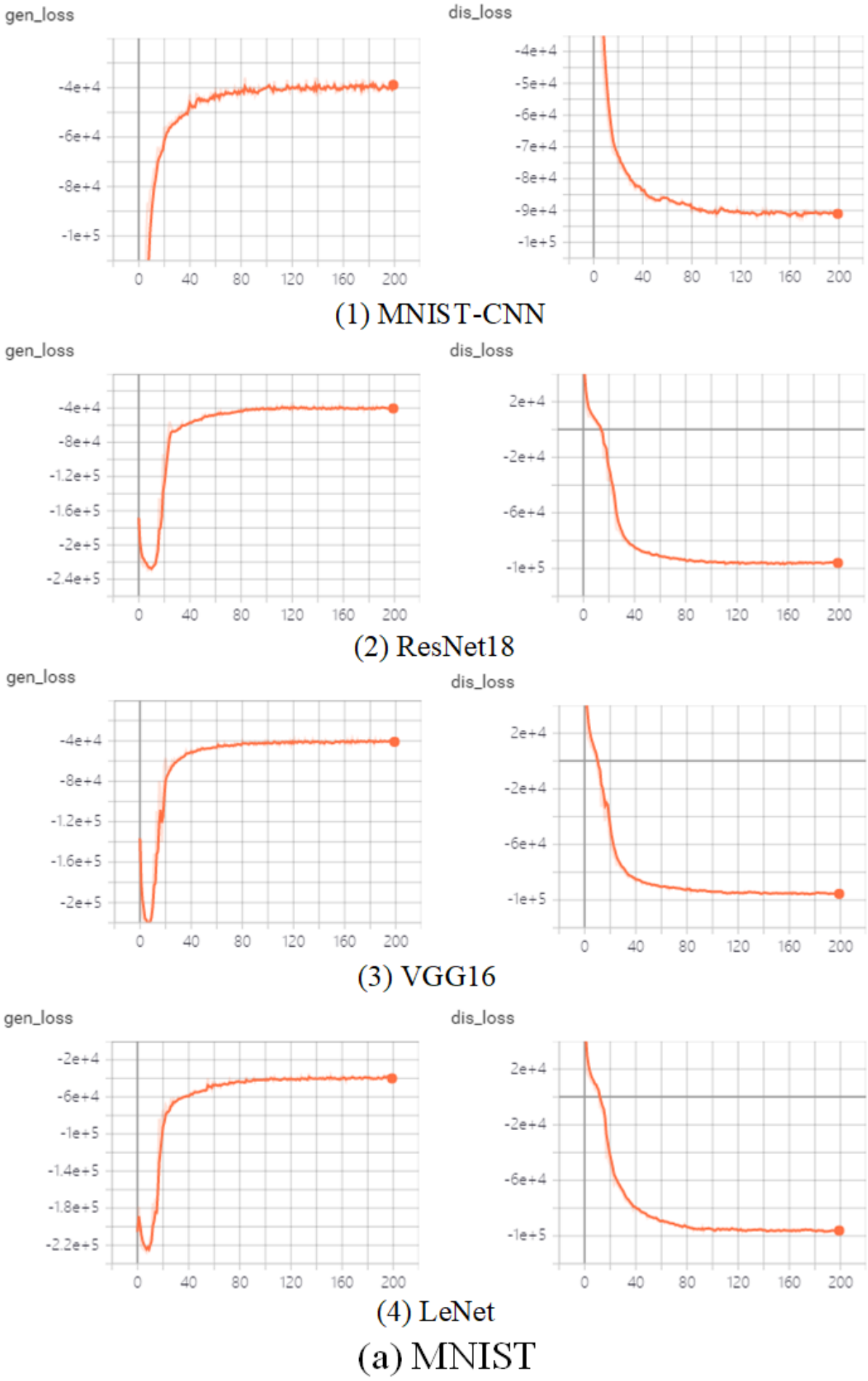
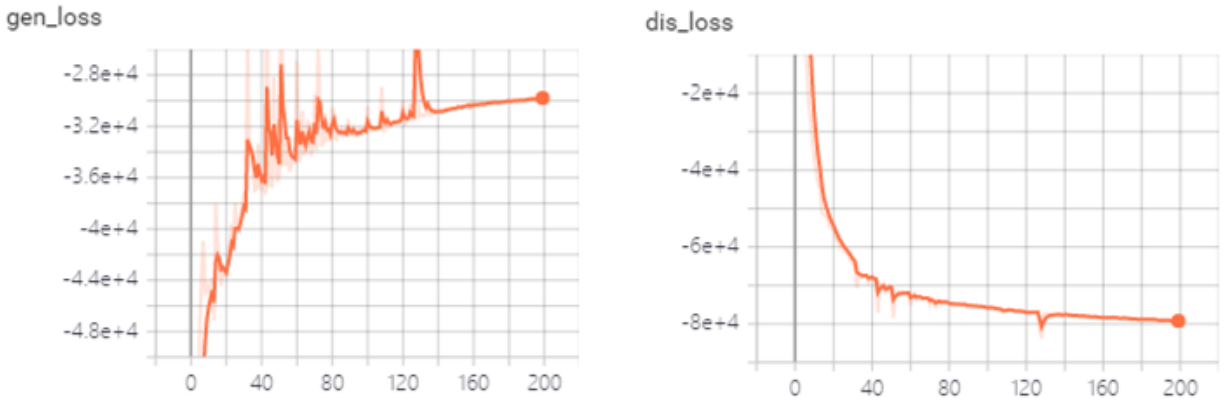
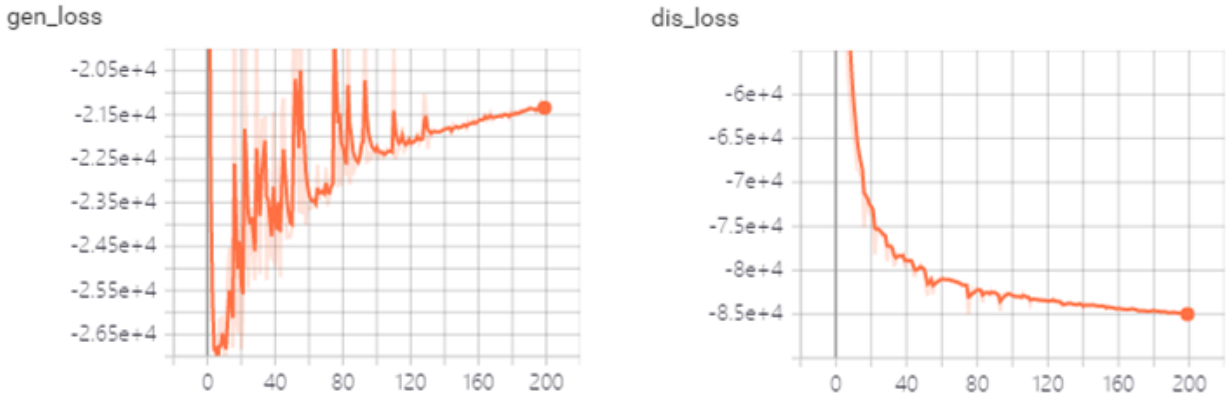


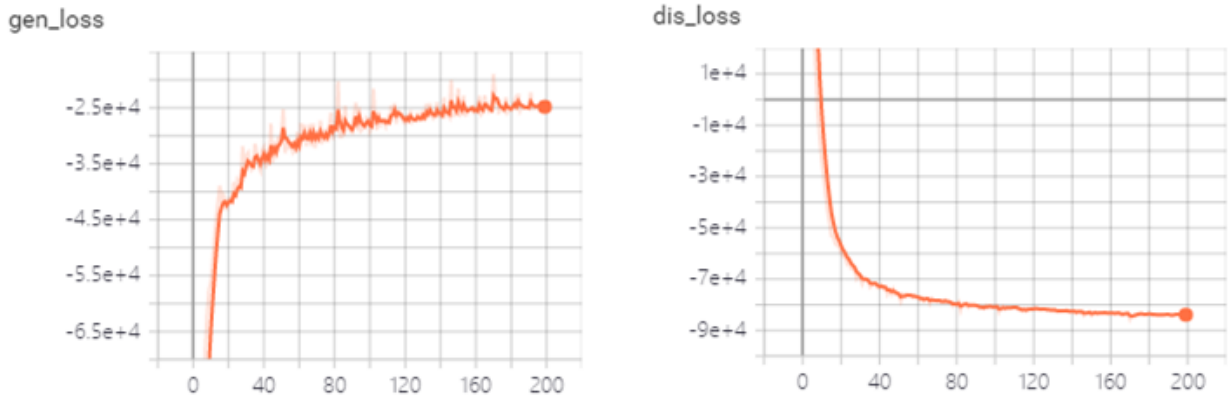
Fig.4.The loss curve of the generator and the discriminator under various third-party classification models.



(1) CIFAR-CNN



(2) ResNet18



(3) VGG16

(4) DenseNet

(b) CIFAR-10

Fig.5.The loss curve of the generator and the discriminator under various third-party classification models.

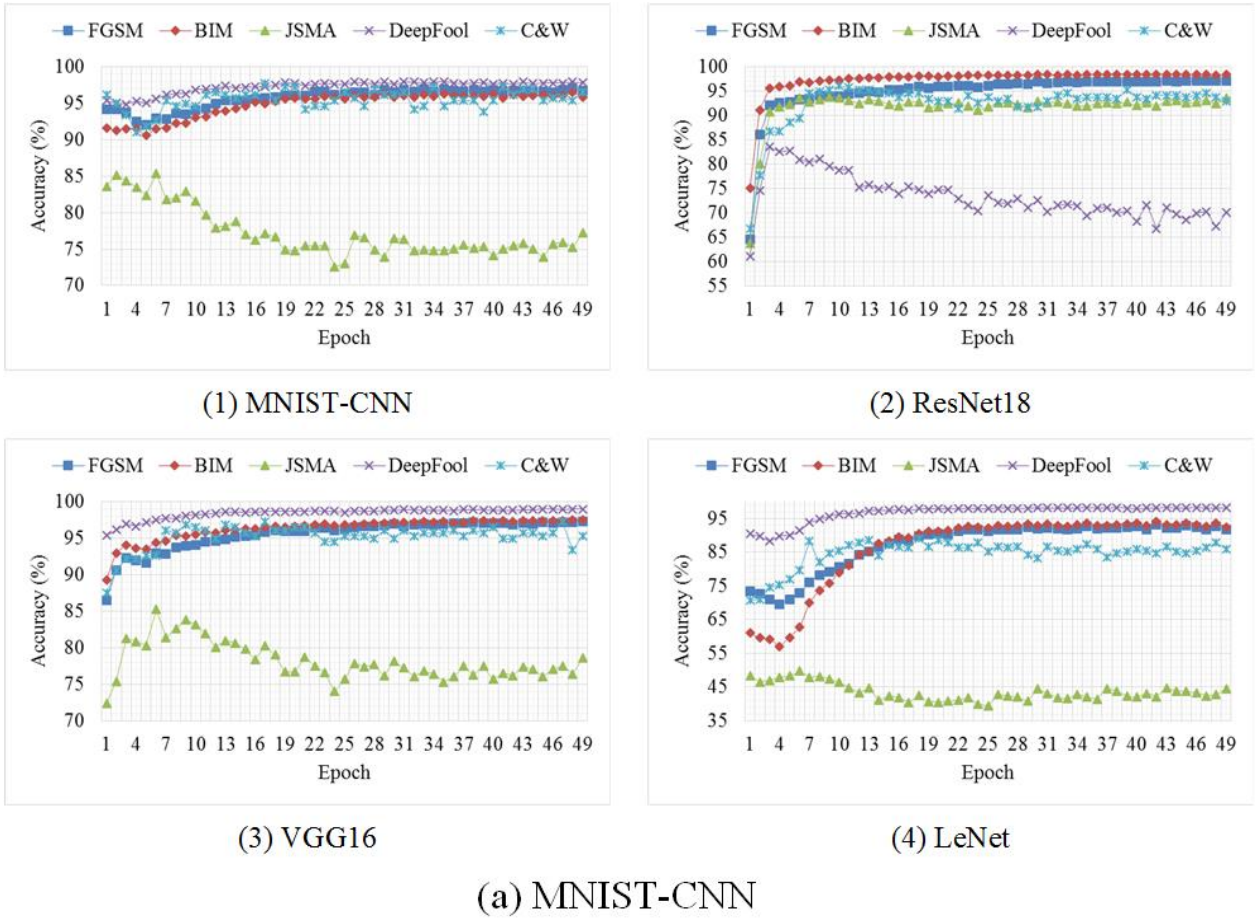


Fig.6.The classification accuracy curves of adversarial examples on the MNIST dataset under the APE-GAN++.

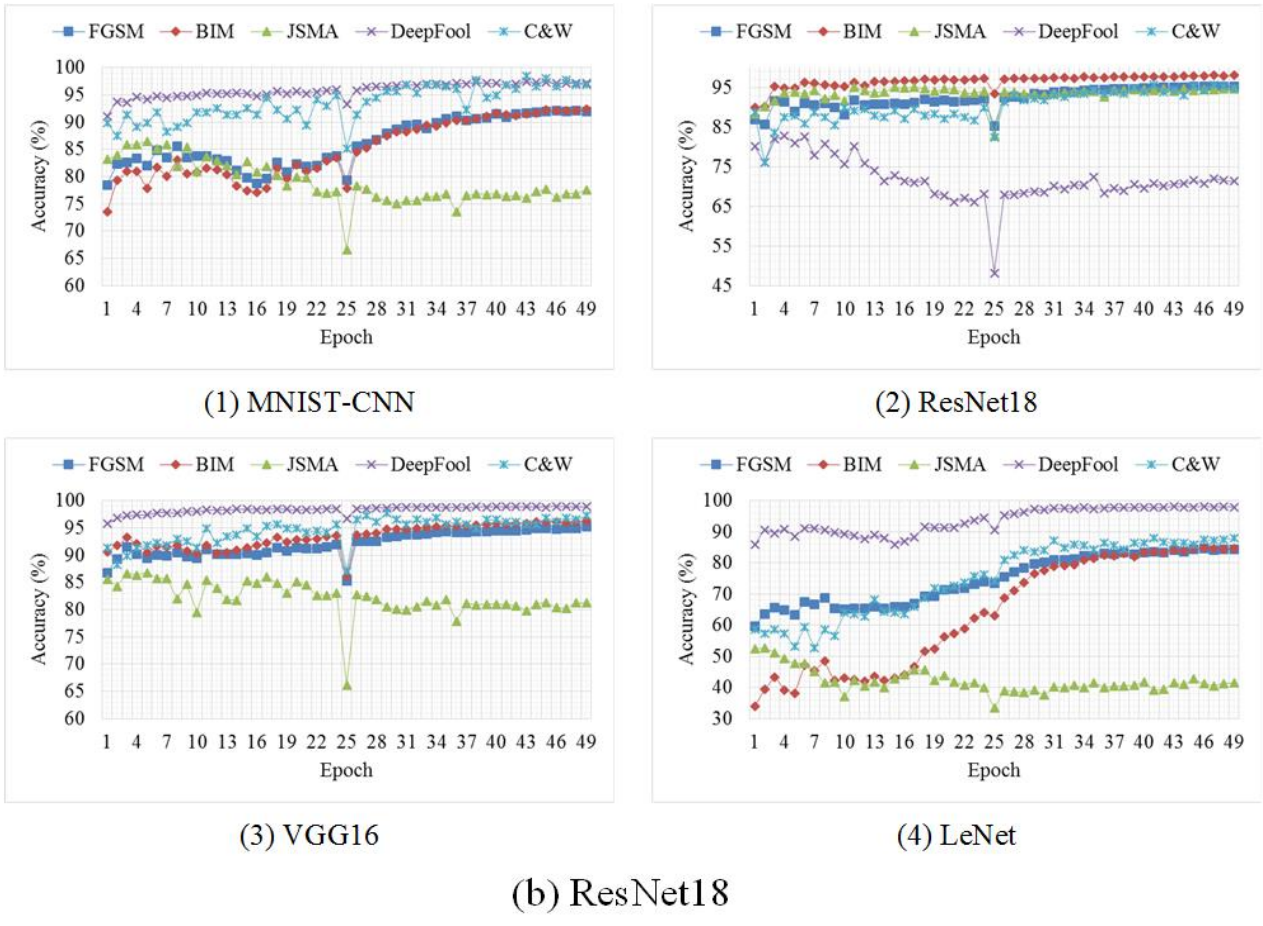


Fig.7.The classification accuracy curves of adversarial examples on the MNIST dataset under the APE-GAN++.

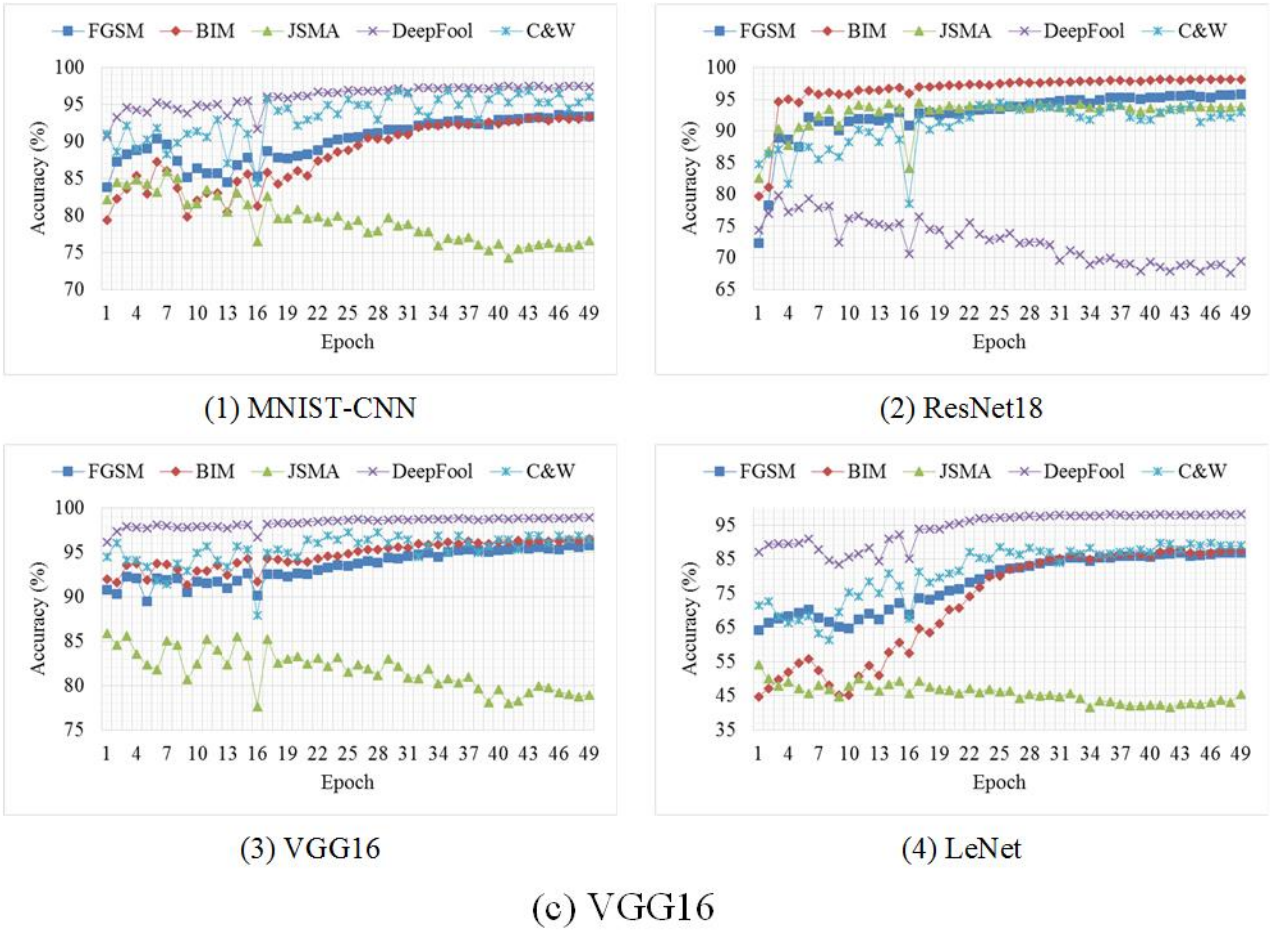


Fig.8.The classification accuracy curves of adversarial examples on the MNIST dataset under the APE-GAN++.

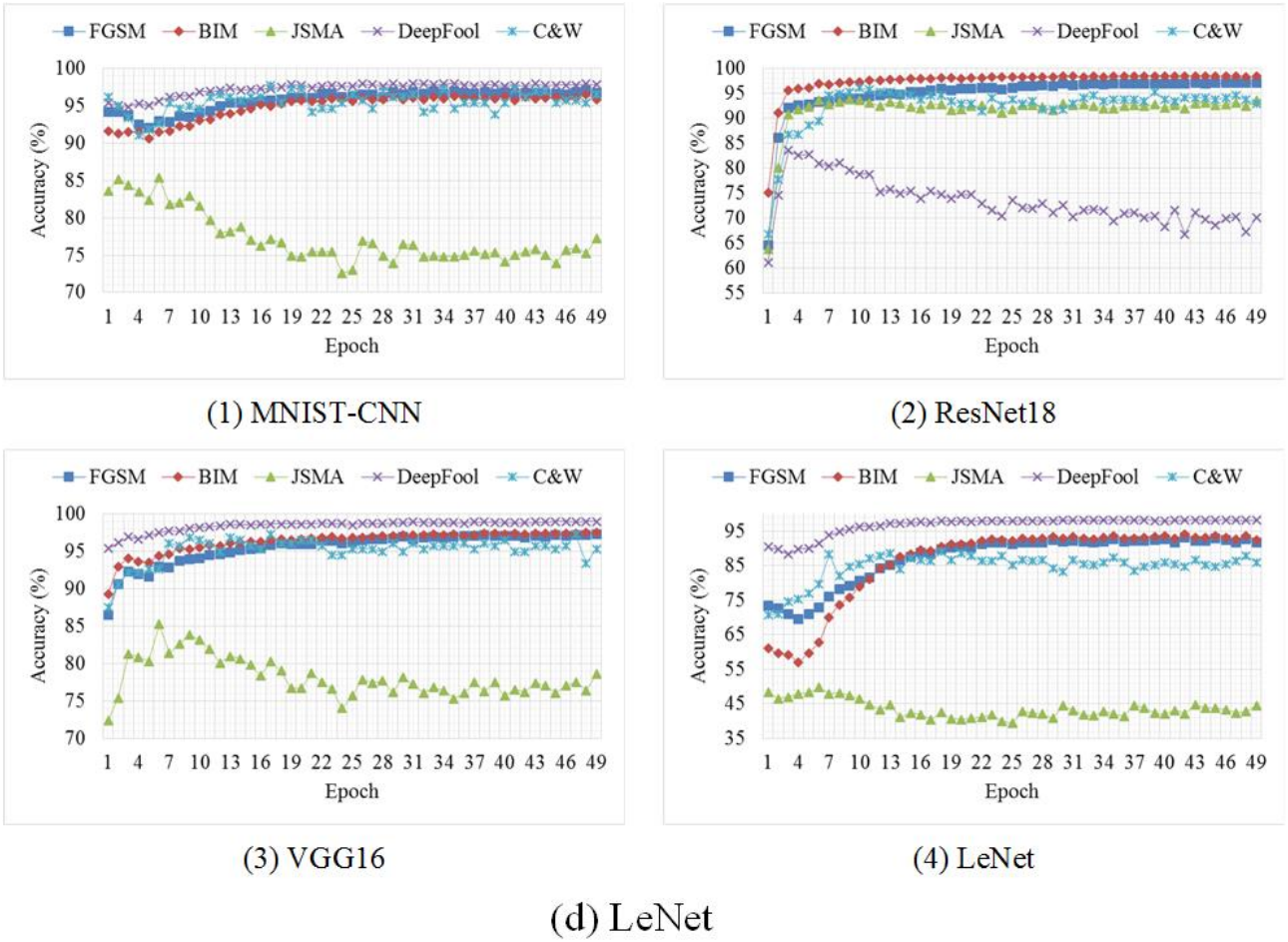


Fig.9.The classification accuracy curves of adversarial examples on the MNIST dataset under the APE-GAN++.

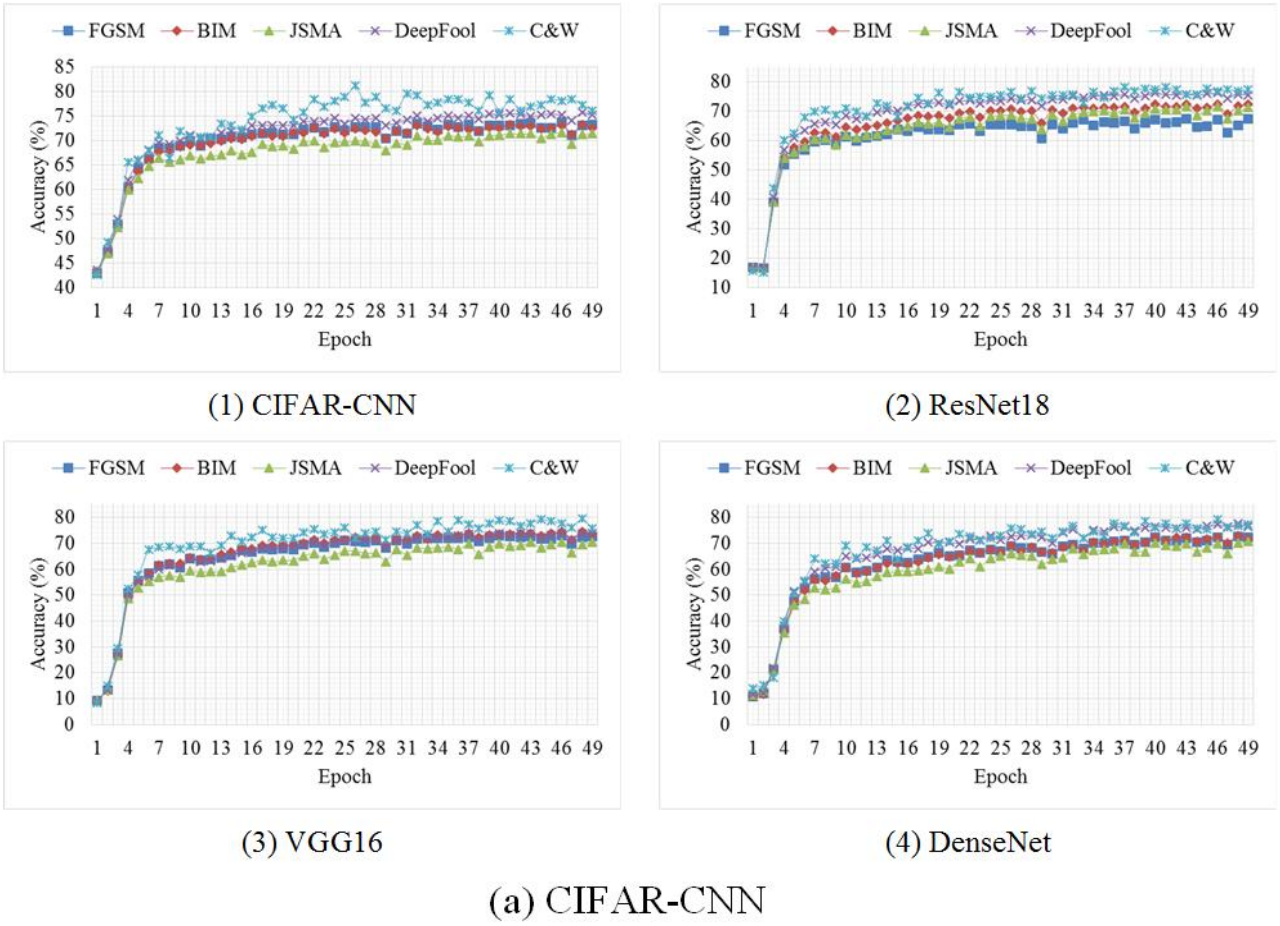


Fig.10.The classification accuracy curves of adversarial examples on the CIFAR-10 dataset under the APE-GAN++.

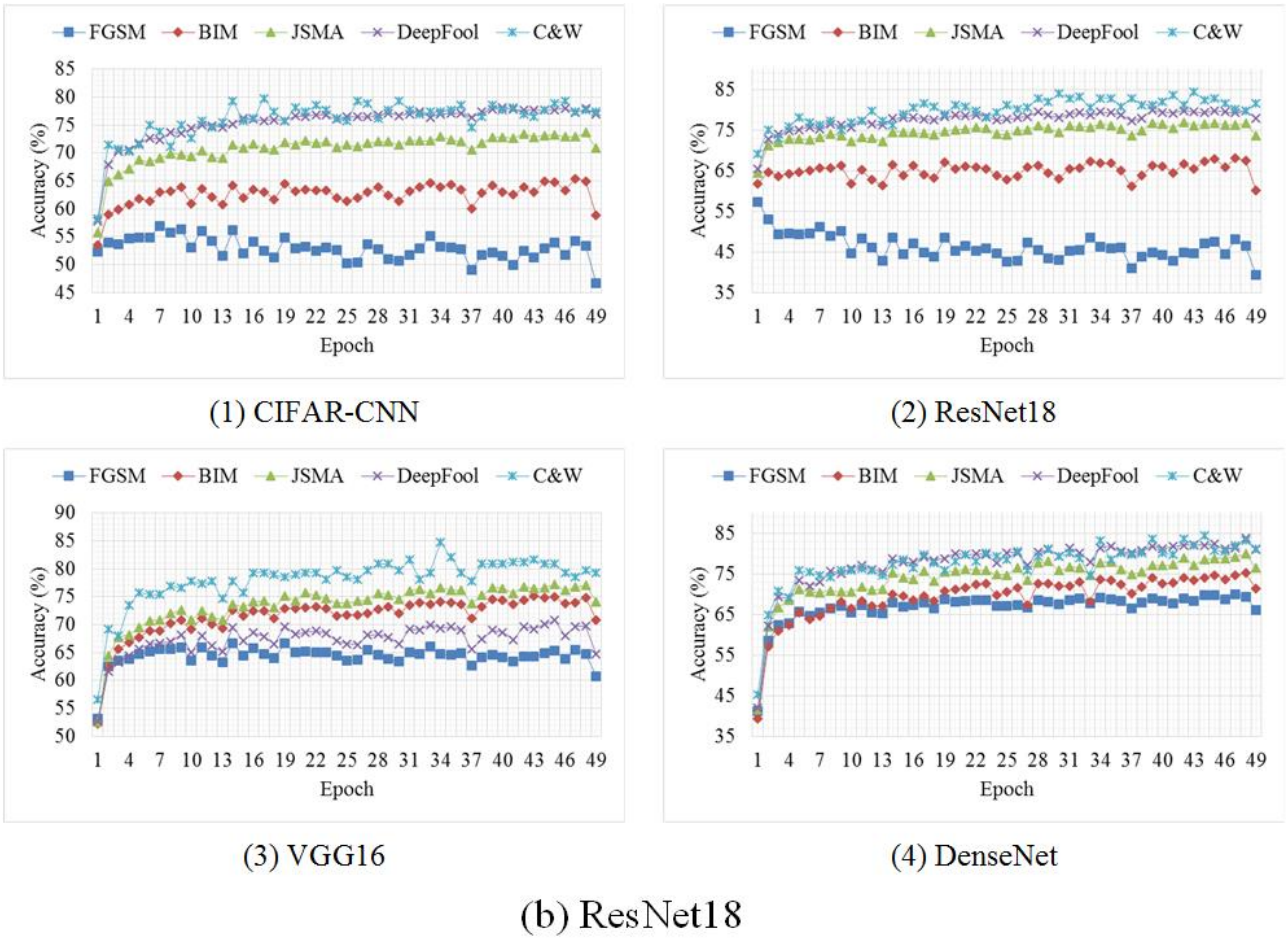


Fig.11.The classification accuracy curves of adversarial examples on the CIFAR-10 dataset under the APE-GAN++.

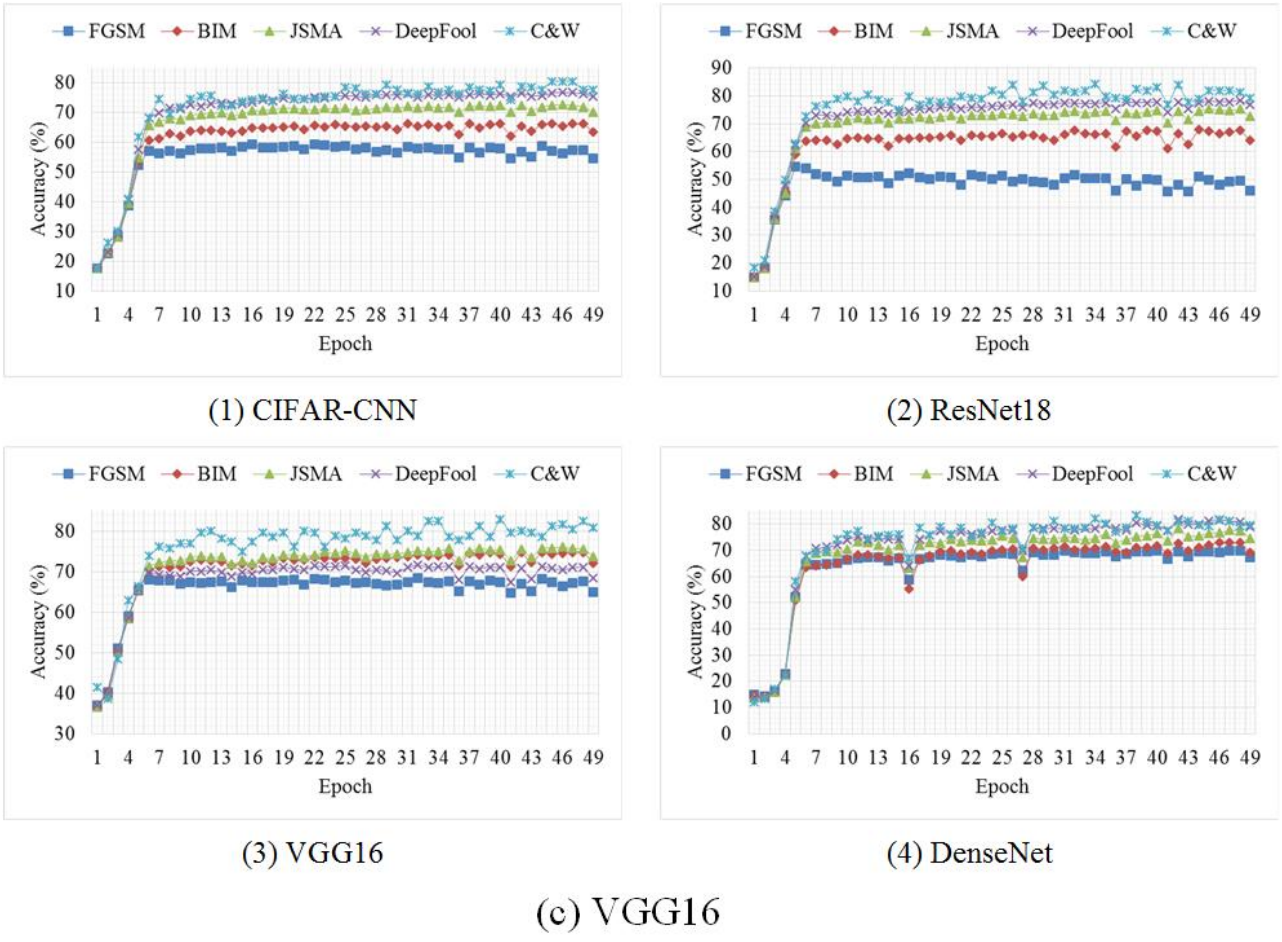


Fig.12.The classification accuracy curves of adversarial examples on the CIFAR-10 dataset under the APE-GAN++.

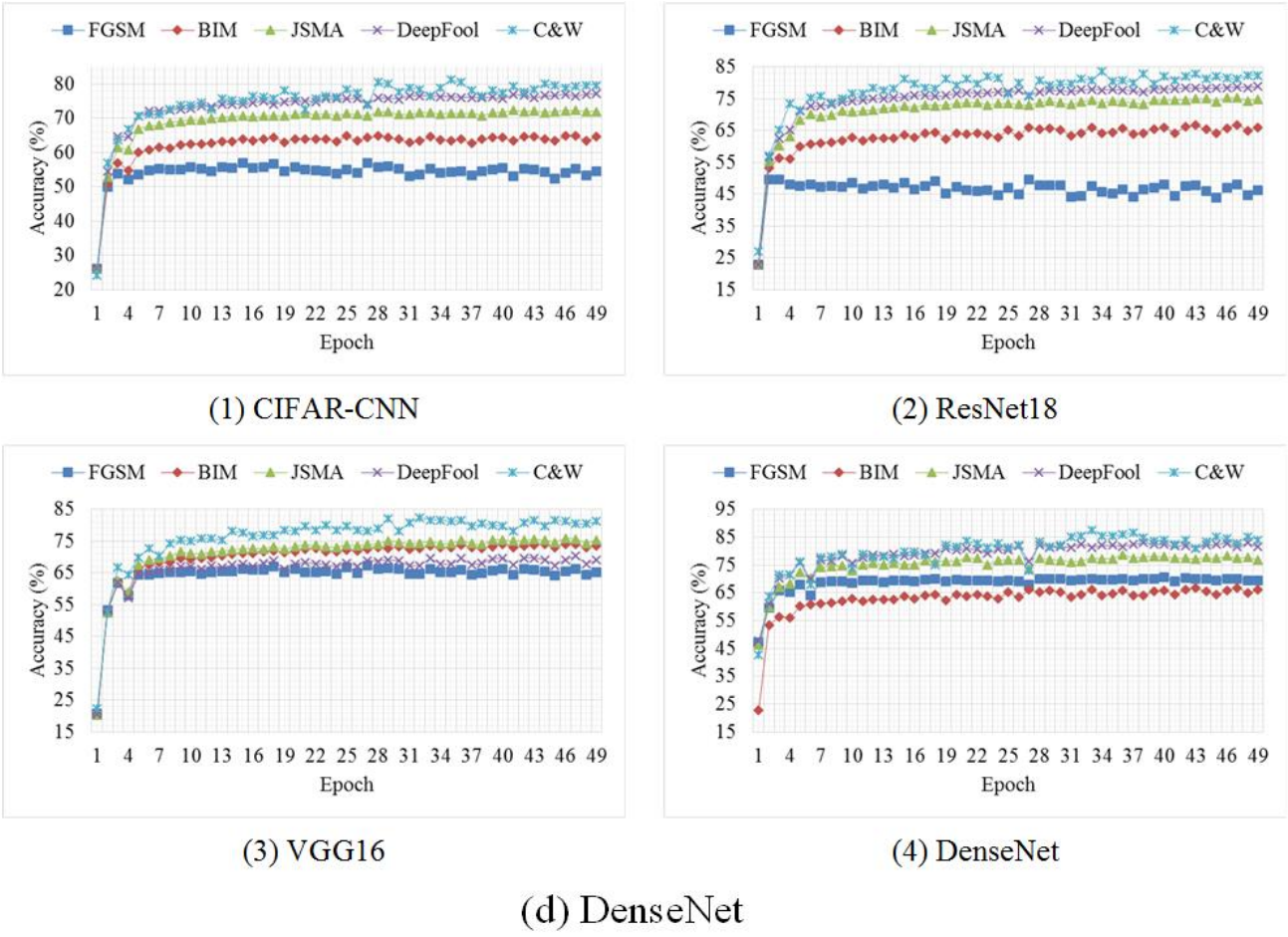


Fig.13.The classification accuracy curves of adversarial examples on the CIFAR-10 dataset under the APE-GAN++.

Fig.14.The visualization of adversarial examples on the MNIST dataset recovering from the APE-GAN++.

Fig.15.The visualization of adversarial examples on the MNIST dataset recovering from the APE-GAN++.

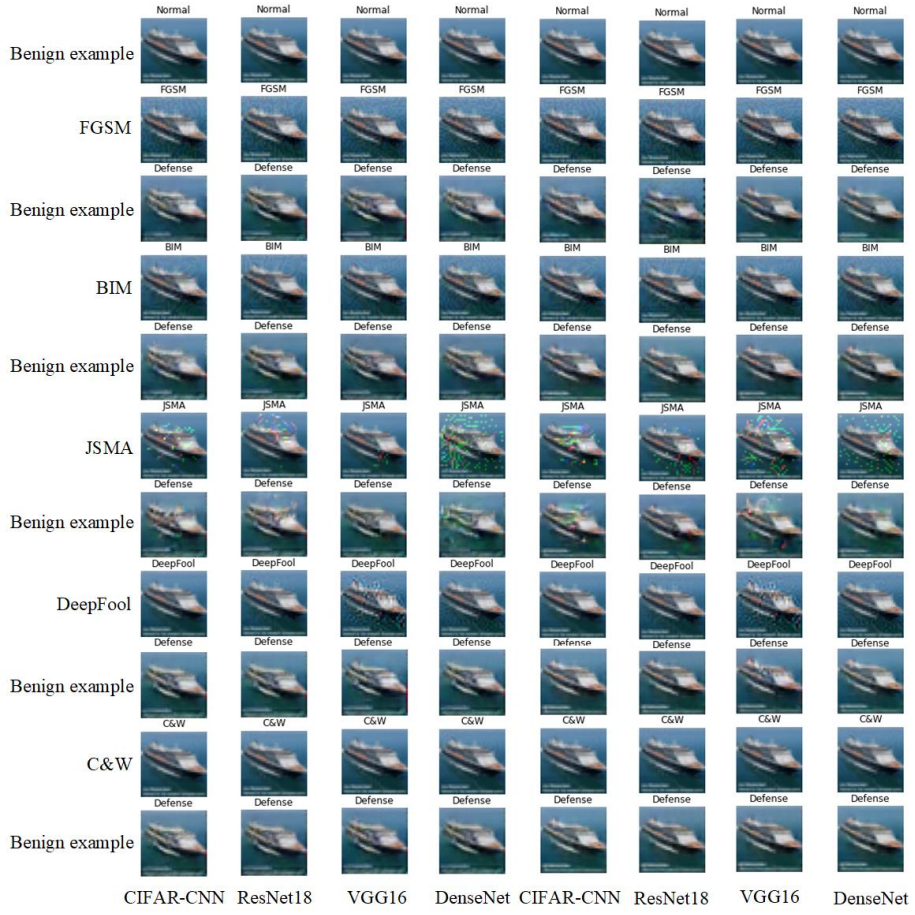


Fig.16.The visualization of adversarial examples on the CIFAR-10 dataset recovering from the APE-GAN++.

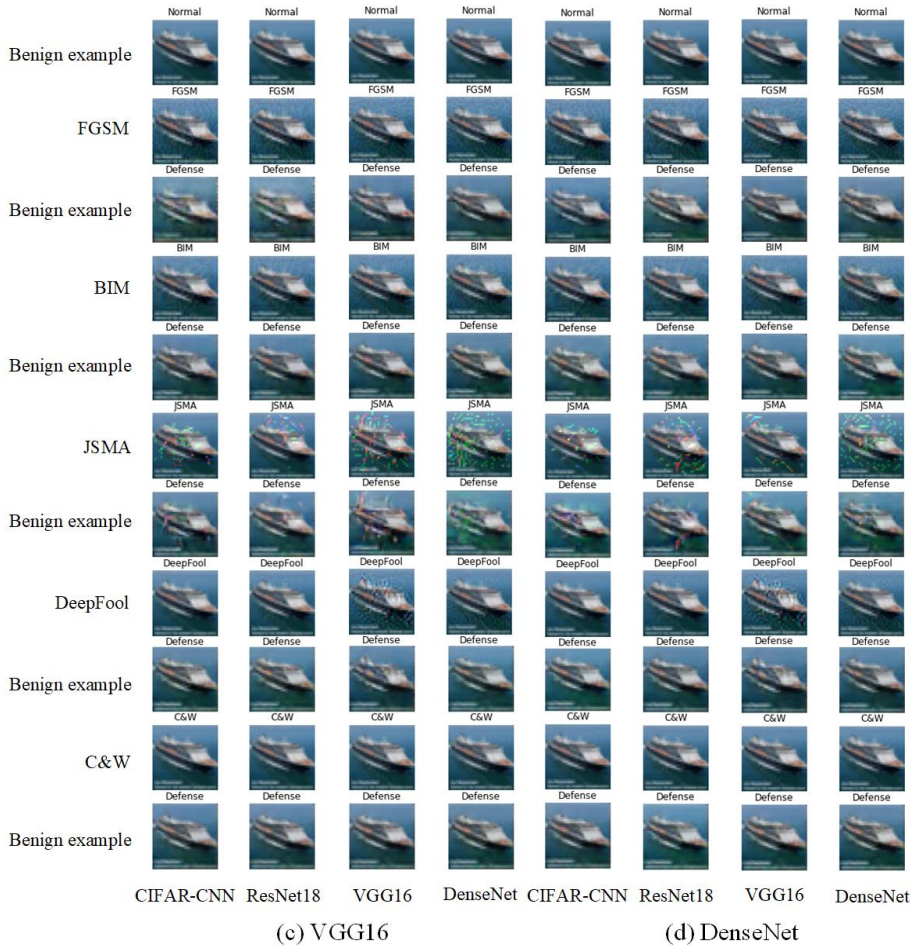
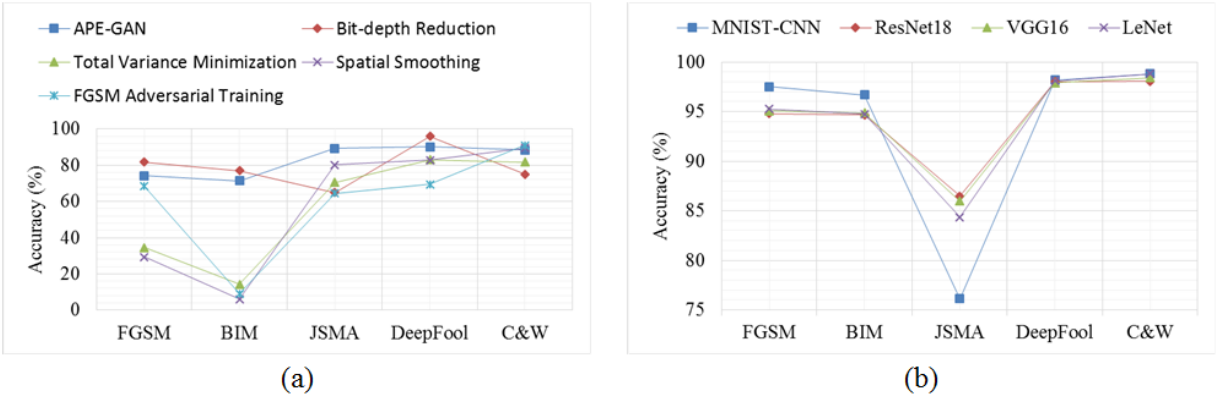


Fig.17The visualization of adversarial examples on the CIFAR-10 dataset recovering from the APE-GAN++.

Fig.18.The visualization of adversarial examples recovering from the APE-GAN.



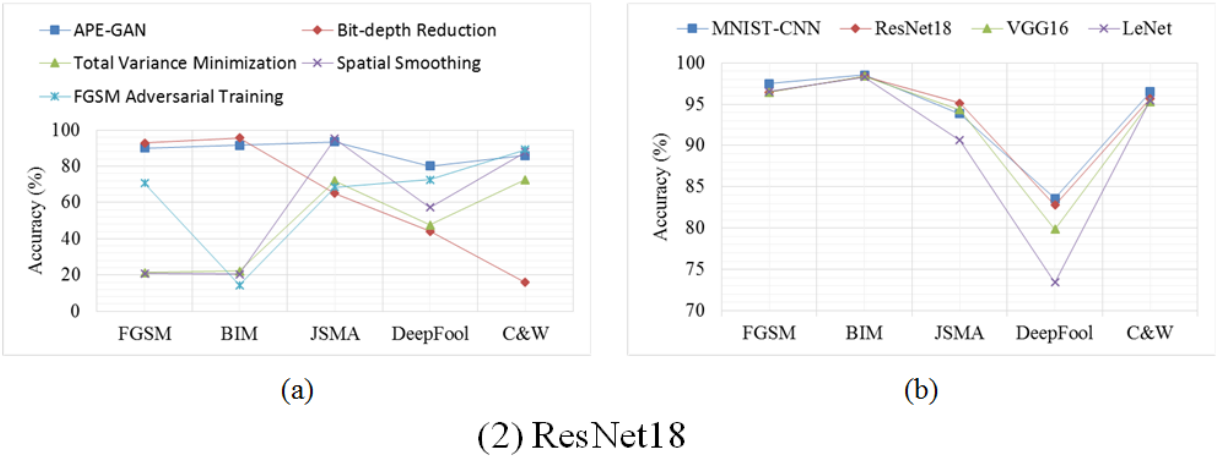


Fig.19.The classification accuracy line chart of adversarial examples on the MNIST dataset under various defense algorithms.

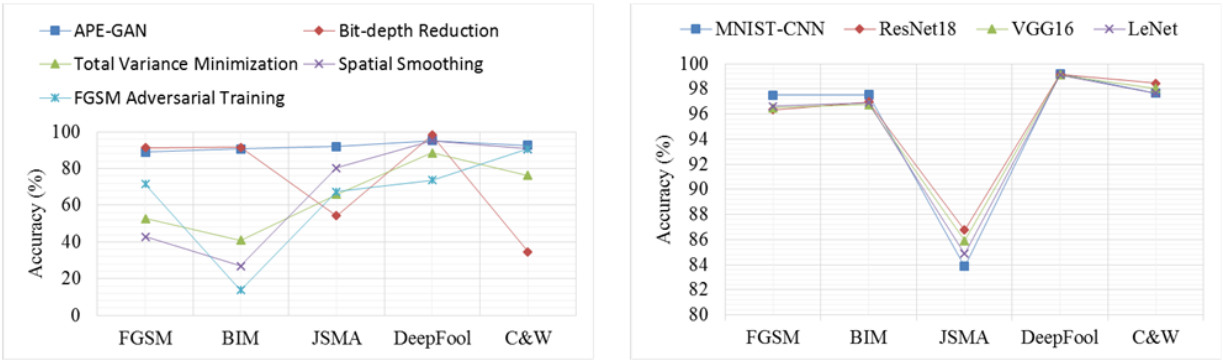


Fig.20.The classification accuracy line chart of adversarial examples on the MNIST dataset under various defense algorithms.

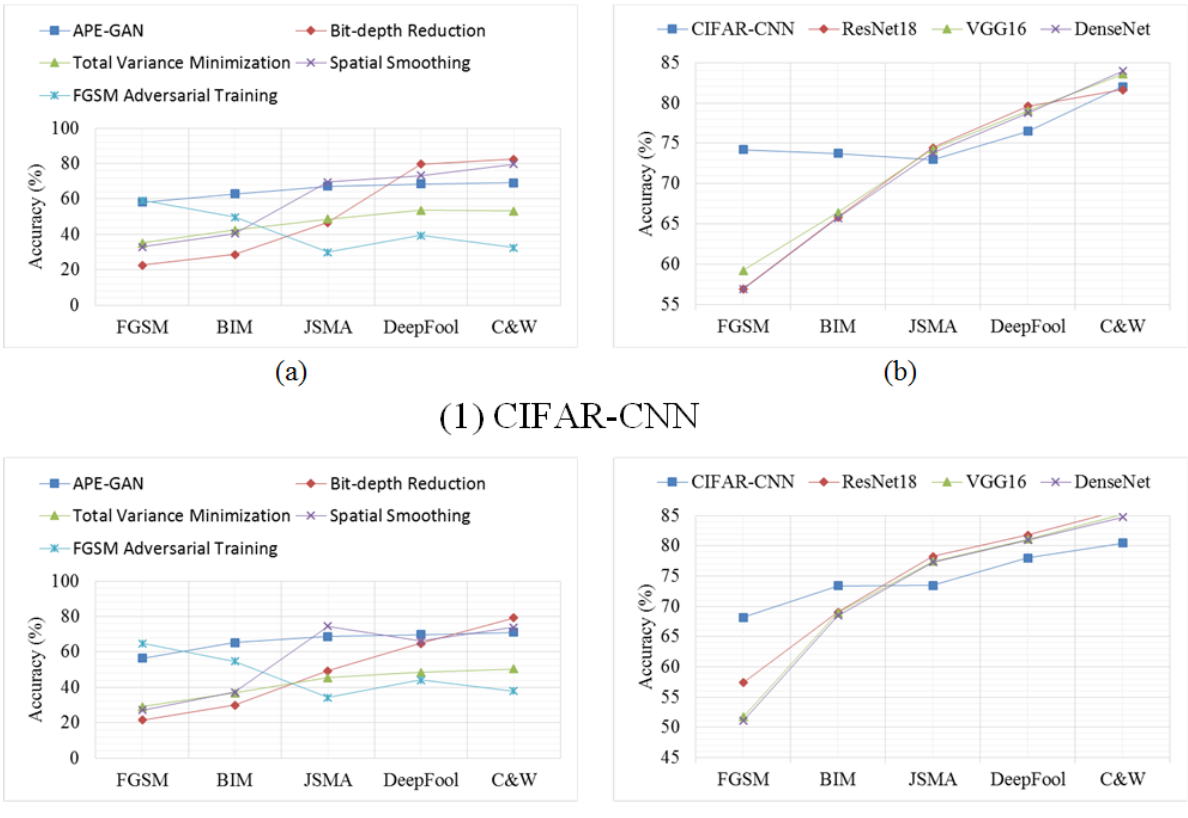


Fig.21.The classification accuracy line chart of adversarial examples on the CIFAR-10 dataset under various defense algorithms.

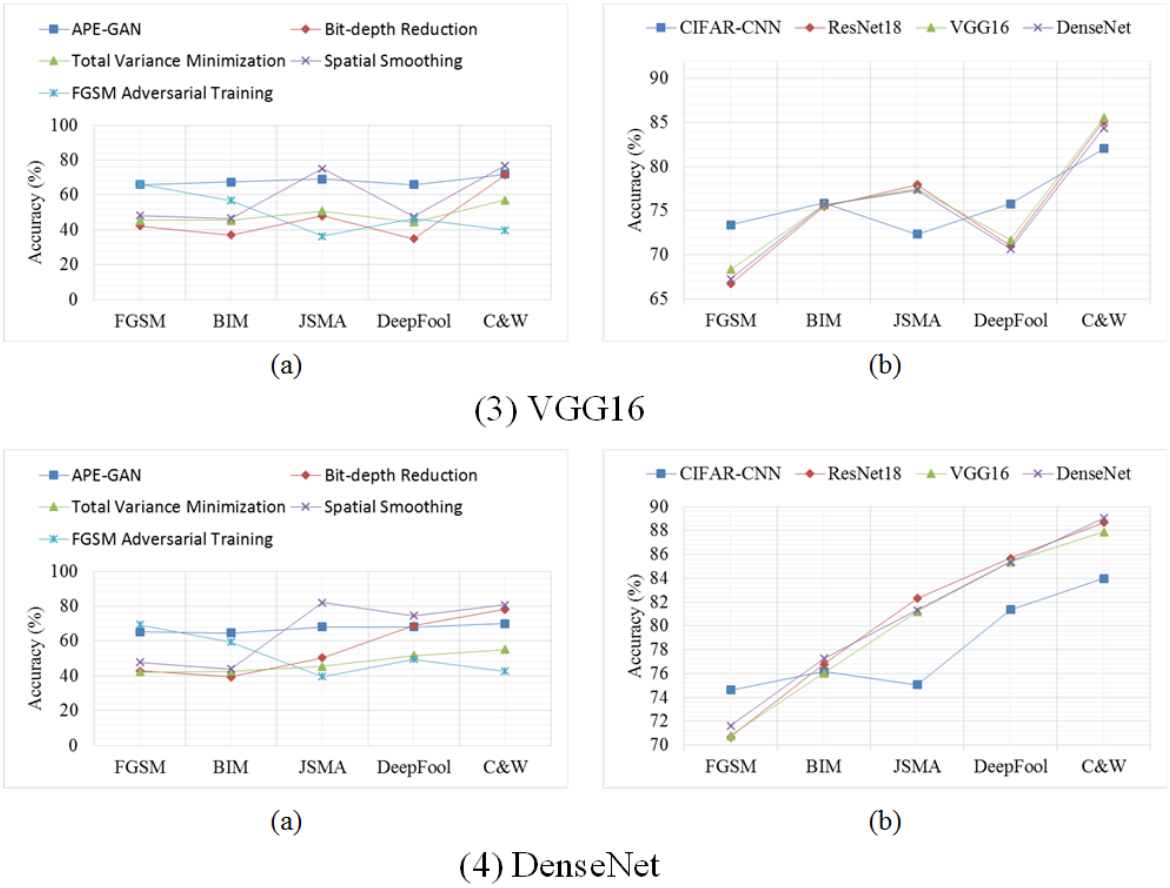


Fig.22.The classification accuracy line chart of adversarial examples on the CIFAR-10 dataset under various defense algorithms.

REFERENCES

[1] Szegedy C, Zaremba W, Sutskever I, "Intriguing properties of neural networks." Computer Science,2013.

[2] Metzen J H, Genewein T, Fischer V, "On detecting adversarial perturbations," 2017, arXiv:1702.04267. [Online]. Available: https://arxiv.org/abs/1702.04267.

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

[4] Feinman R, Curtin R R, Shintre S, "Detecting adversarial samples from artifacts,"2017, arXiv:1703.00410. [Online]. Available: https://arxiv.org/abs/1703.00410.

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

[6] Xu W, Evans D, Qi Y, "Feature squeezing: detecting adversarial examples in deep neural networks," 2017, arXiv:1704.01155. [Online]. Available: https://arxiv.org/abs/1704.01155.

[7] Carlini N, Wagner D, "Adversarial examples are not easily detected: Bypassing ten detection methods," Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp.3-14, 2017.

[8] Tramèr F, Kurakin A, Papernot N, "Ensemble adversarial training: Attacks and defenses," 2017, arXiv:1705.07204. [Online]. Available: https://arxiv.org/abs/1705.07204.

[9] Hinton G, Vinyals O, Dean J, "Distilling the knowledge in a neural network" 2015, arXiv:1503.02531. [Online]. Available: https://arxiv.org/abs/1503.02531.

[10] Dhillon G S, Azizzadenesheli K, Lipton Z C, "Stochastic activation pruning for robust adversarial defense," 2018, arXiv:1803.01442. [Online]. Available: https://arxiv.org/abs/1803.01442.

[11] Song Y, Kim T, Nowozin S, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," 2017, arXiv:1710.10766. [Online]. Available: https://arxiv.org/abs/1710.10766.

[12] Shen S, Jin G, Gao K, "Ape-gan: adversarial perturbation elimination with gan," 2017, arXiv:1707.05474. [Online]. Available: https://arxiv.org/abs/1707.05474.

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

[14] Guo C, Rana M, Cisse M, "Countering adversarial images using input transformations," 2017, arXiv:1711.00117. [Online]. Available: https://arxiv.org/abs/1711.00117.

[15] Buckman J, Roy A, Raffel C, "Thermometer encoding: One hot way to resist adversarial examples," International Conference on Learning Representations, 2018.

[16] Yang Y, Zhang G, Katabi D, et al, "Me-net: Towards effective adversarial robustness with matrix estimation," 2019, arXiv:1905.11971. [Online]. Available: https://arxiv.org/abs/1905.11971.

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

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

[19] Lindqvist B, Sugrim S, Izmailov R, "AutoGAN: Robust classifier against adversarial attacks," 2018, arXiv:1812.03405. [Online]. Available: https://arxiv.org/abs/1812.03405.

[20] Liu G, Khalil I, Khreishah A, "Gandef: A gan based adversarial training defense for neural network classifier," IFIP International Conference on ICT Systems Security and Privacy Protection. Springer, Cham, pp.19-32, 2019.

[21] Hashemi A S, Mozaffari S, "Secure deep neural networks using adversarial image generation and training with Noise-GAN," Computers & Security, vol.86, pp.372-387, 2019.

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

[23] Kurakin A, Goodfellow I, Bengio S, "Adversarial machine learning at scale," 2017 ICLR, arXiv:1611.01236. [Online]. Available: https://arxiv.org/abs/1611.01236.

[24] Moosavi-Dezfooli S M, Fawzi A, Frossard P, "Deepfool: a simple and accurate method to fool deep neural networks," Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016.

[25] Papernot N, McDaniel P, Jha S, "The limitations of deep learning in adversarial settings," 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016.
[26] Carlini N, Wagner D, "Towards evaluating the robustness of neural networks," 2017 ieee symposium on security and privacy (sp). IEEE, pp.39-57, 2017.
[27] Goodfellow I J, Pouget-Abadie J, Mirza M, "Generative adversarial networks," 2014, arXiv:1406.2661. [Online]. Available: https://arxiv.org/abs/1406.2661.
[28] Gulrajani I, Ahmed F, Arjovsky M, "Improved training of wasserstein gans," 2017, arXiv:1704.00028. [Online]. Available: https://arxiv.org/abs/1704.00028.
[29] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[30] Simonyan K, Zisserman A, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv:1409.1556. [Online]. Available: https://arxiv.org/abs/1409.1556.
[31] El-Sawy A, Hazem E L B, Loey M. CNN for handwritten arabic digits recognition based on LeNet-5[C]//International conference on advanced intelligent systems and informatics. Springer, Cham, 2016: 566-575.
[32] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

Rui Yang is studying for a master's degree in the School of Computer Science and Technology at China University of Mining and Technology. He received the B.S. degree from Xuzhou Medical University in 2019. His research interests include adversarial machine learning and information security.

Xiu-Qing Chen is a lecturer at the School of Medical Information and Engineering, Xuzhou Medical University. She received her bachelor's degree and master's degree from the China University of Mining and Technology. She has been a Ph.D. degree candidate in applied computer Technology from the China University of Mining and Technology. Her research interests include security protocols and network security.

Tian-Jie Cao is a professor at the School of Computer Science and Technology, China University of Mining and Technology. He received the BS and MS degree in mathematics from Nankai University, Tianjin, China, and the Ph.D. degree in computer software and theory from State Key Laboratory of Information Security of Institute of Software, Chinese Academy of Sciences, Beijing, China. His research interests are in security protocols and network security.