Adversarial example defense based on image reconstruction

Yu Zhang, Huan Xu, Chengfei Pei, and Gaoming Yang

School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan, Anhui, China

Corresponding author:

Yu Zhang

Email address: yuzhang@aust.edu.cn

ABSTRACT

The rapid development of deep neural networks (DNN) has promoted the widespread application of image recognition, natural language processing, and autonomous driving. However, DNN is vulnerable to adversarial examples. Such as, an input sample with imperceptible perturbation can easily invalidate the DNN and even deliberately modify the classification results. So this paper proposes a preprocessing defense framework based on image compression reconstruction to achieve adversarial example defense. Firstly, the defense framework performs pixel depth compression on the input image based on the sensitivity of the adversarial example to eliminate adversarial perturbations. Secondly, we use the super-resolution image reconstruction network to restore the image quality and then map the adversarial example to the clean image. Therefore, there is no need to modify the network structure of the classifier model, and it can be easily combined with other defense methods. Finally, we evaluate the algorithm with MNIST, Fashion-MNIST, and CIFAR-10 datasets; the experimental results show that our approach outperforms current techniques in the task of defending against adversarial example attacks.

INTRODUCTION

Deep neural networks have been widely used in computer vision, natural language processing, speech recognition, and other fields (Karen and Andrew, 2015). However, the adversarial example proposed by Szegedy et al. (Szegedy et al., 2013), as shown in Figure 1, can easily deceive the neural network by adding a minor perturbation to the ordinary image, i.e., the deep convolutional neural network will continuously amplify this perturbation, which is sufficient to drive the model to make high confidence incorrect predictions without being detected by the human eye. As a result, the adversarial example has a minor perturbation than the normal noise. However, it brings more significant obstacles to practical applications. Researchers usually input the pictures directly into the neural network for the computer classification test when training the classifier model to solve this problem. Such as, Kurakin et al. (Kurakin et al., 2016) find that a significant fraction of adversarial images crafted using the original network are misclassified even when fed to the classifier through the camera. Nowadays, the research and implementation of autonomous driving (Deng et al., 2020) and person detection (Thys et al., 2019) rely heavily on deep learning technology. In addition to making the target model random errors, an adversarial example can also conduct targeted attacks according to the attacker's wishes and generate specified results. Such as Eykholt et al. (Eykholt et al., 2018) shows that adversarial examples bring substantial security risks to the application of related technologies. Furthermore, by adding adversarial perturbation to a road sign, the intelligent system may recognize the deceleration sign as an acceleration sign, which will bring substantial hidden dangers to traffic safety.

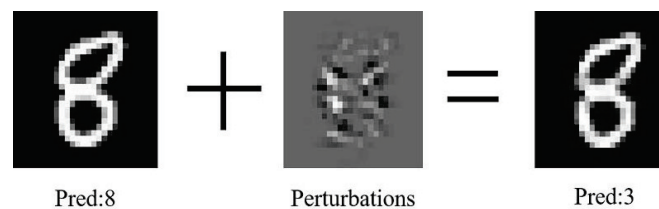



Figure 1. The generation process of adversarial example.

Currently, the reasons for the adversarial examples are still controversial. Such as Szegedy et al. (Szegedy et al., 2013) believe that it is caused by the nonlinearity of the model, while Kurakin et al. (Kurakin et al., 2016) propose that the high-dimensional space's linearity is sufficient to generate adversarial examples. If the input samples have sufficiently large dimensions for linear models, they are also attacked by adversarial examples. Adversarial attacks can be divided into single-step attacks, which perform only one step of gradient calculation, such as the FGSM (Goodfellow et al., 2015), and iterative attacks, which perform multiple steps to obtain better adversarial examples, such as BIM (Ren et al., 2020), CW (Carlini and Wagner, 2017). At the same time, adversarial example attacks can be categorized into white-box, gray-box, and black-box attacks based on the attacker's knowledge. A white-box attack means that the attacker knows all the information, including models, parameters, and training data. We can use it to calculate the attack distance and generate adversarial examples. A gray-box attack means that the attacker knows limited target model information. A black-box attack means that an attacker uses a similar model to generate adversarial examples. The generated adversarial examples have a certain degree of transferability, which can carry out transfer attacks on the model without knowing the relevant information of the model, and it has a high success rate.

Furthermore, extreme samples can even deceive multiple different models. Generally, adversarial examples not only exist in images, but also in speech and text (Xu et al., 2020), which makes the application of deep learning technology have huge uncertainty and diversity, and there are potential threats at the same time. Therefore, it is urgent to defend, which makes the application of deep learning technology have huge uncertainty and diversity, as well as many potential threats.

With the endless emergence of attack methods, the defense of adversarial examples has become a significant challenge. Many defense methods (Dong et al., 2018; Zhang and Wang, 2019; Hameed et al., 2020; Singla and Feizi, 2020; Jin et al., 2021) have been proposed, such as adversarial training (Goodfellow et al., 2015), which increases model robustness by adding adversarial examples to the training process. Some other defenses mainly rely on preprocessing methods to detect or transform the input image before the target network without modifying the target model. For example, Xu et al. (Xu et al., 2017) propose that the input's adversarial perturbation can be eliminated by reducing the color bit depth of each pixel and spatial smoothing, and they create a defense framework to detect adversarial examples in the input. Jia et al. (Jia et al., 2019) introduce the ComDefend defense model, which constructs two deep convolutional neural networks: the one for compressing images and retaining valid information; the other for reconstructing images. However, this method does not perform well under the attack of BIM.

On the other hand, if you only perform detection without other measures when defending against adversarial examples, it will not be able to meet actual needs. For example, in an autonomous driving application scenario, the defense system recognizes a road sign and detects that it is an adversarial example. At this time, the defense system refuses to input the image, which will seriously affect its normal operation. In addition, convolutional neural networks are used to extract image features and compress images. If the compression rate is too low, the uncorrupted adversarial perturbation in the reconstruction network will continue to expand, thereby significantly reducing the classifier's accuracy.

To solve the above problems, we propose a defense framework based on image compression reconstruction, which is a preprocessing method. Figure 2 clearly describes the defense framework of this paper. The defense model in the figure can be divided into two steps. The specific operation is to eliminate adversarial perturbations by compressing images to defend against adversarial example attacks. Simultaneously, to ensure that the standard and processed samples do not suffer from performance loss on the target model, we use the deep convolutional neural network to repair the processed images. In short, this paper makes the following contributions:

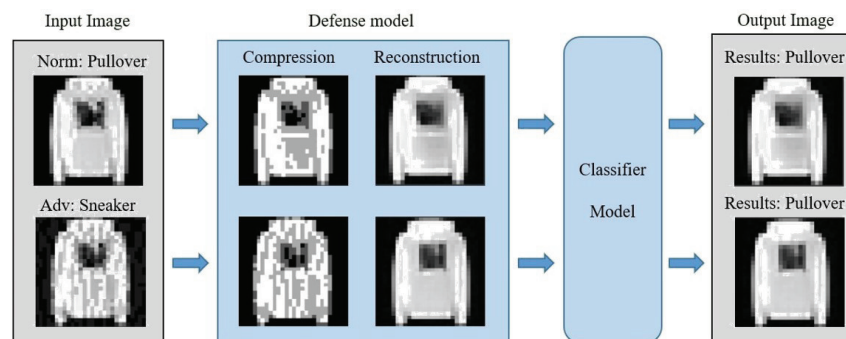



Figure 2. The defense framework uses input samples as pictures.

- To defend against various adversarial example attacks, we propose a defense framework based on image compression and reconstruction with super-resolution. This framework eliminates adversarial perturbations by compressing the input samples and then reconstructs the compressed images using super-resolution methods to alleviate the performance degradation caused by compression.

- As a preprocessing method, there is no need to modify the target model during the defense process, i. e., our method has good performance for single-step and iterative attacks and has a small calculation compared with other adversarial training methods. In addition, it can be combined with different target models to have a protective effect still.

- To verify the effectiveness, applicability, and transferability of the method, extensive experiments of defense tests are carried out on three real data sets and multiple attack methods. The results show that our approach can achieve better defense performance for different adversarial example attacks and significantly reduce image loss.

The rest of this paper is organized as follows: Section 2 briefly introduces an background of the existing attack and defense methods. Section 3 discusses the methodology and defense framework proposed in this paper in detail, followed by many experiments to demonstrate the feasibility of this method in Section 4. Finally, the conclusion is given in Section 5.

BACKGROUND

In this section, we review related works from two aspects: the attack methods of generating adversarial examples and the defensive techniques of resisting adversarial examples.

Attack methods

In order to verify the versatility of the proposed method, the following four different methods are mainly used to generate adversarial examples.

Fast Gradient Sign Method (FGSM)

Goodfellow et al. propose the FGSM (Goodfellow et al., 2015), a fast and straightforward method of generating adversarial examples. Given the input image, the maximum direction of gradient change of the deep learning model is found, and adversarial perturbations are added to maximize the cost subject to a L_∞ constraint, resulting in the wrong classification result. The FGSM adds the imperceptible perturbations to the image by increasing the image classifier loss. The generated adversarial example is formulated as follows:

$$x^{adv} = x + \varepsilon \cdot sign(\bigtriangledown_x J(\theta, x, y_{true})) \tag{1}$$

where $J(\theta, x, y)$ denotes the cross entropy cost function, x is the input image, y is the true label of the input image, and ε is the hyperparameter that determines the magnitude of the perturbations.

117   ***Basic Iterative Method (BIM)***

118   The problem of adversarial examples is constantly being studied. Such as, Kurakin et al. (Kurakin et al.,
119   2016) present a more direct basic iterative method (BIM) to improve the performance of FGSM. In other
120   words, BIM is an iterative version of FGSM. It uses the basic idea of gradient descent to perform iterative
121   training with small steps. Moreover, clip the pixel values of the intermediate results after each step to
122   ensure that they are in an ε-neighborhood of the original image:

$$x_0^{adv} = x, \cdots, x_{N+1}^{adv} = clip_{x,\varepsilon}\{x_N^{adv} + \alpha \cdot sign(\bigtriangledown_x J(\theta, x, y_{true}))\} \tag{2}$$

123   Among them, x is the input image, y_{true} is the true class label, $J(\theta, x, y)$ is the loss function, and α is the
124   step size, usually $\alpha = 1$.

125   This method attempts to increase the loss value of the correct classification and does not indicate
126   which type of wrong class label the model should choose. Therefore, it is suitable for data sets with fewer
127   and different types of applications.

128   ***Carlini*** & ***Wagner*** *(C&W)*

129   Carlin and Wagner propose an optimization-based attack method called *C&W* (Carlini and Wagner, 2017).
130   *C&W* can be a targeted attack or an untargeted attack. The distortion caused by the attack is measured by
131   three metrics: (L_0, L_2, L_∞). There are three methods introduced by *C&W*, which are more efficient than
132   all previously-known methods in terms of achieving the attack success rate with the smallest amount of
133   imperceptible perturbation. A successful *C&W* attack usually needs to meet two conditions. First, the
134   difference between the adversarial examples and the corresponding clean samples should be as slight as
135   possible. Second, the adversarial examples should make the model classification error rate as high as
136   possible. The details are shown in (3).

$$\min \|\frac{1}{2}(\tanh(x_n + 1) - X_n)\|_2^2 + c \cdot f(\frac{1}{2}\tanh(x_n) + 1)$$
$$Where f(x^{'}) = \max(\max\{Z(x^{'})_i : i \neq t\} - Z(x_t^{'}), -k) \tag{3}$$

137   where the Z is the softmax function, the k is a constant used to control the confidence, the t is the target
138   label of misclassification, and c is constant chosen with binary search. In the above formula, $\tanh(x)$ refers
139   to the mapping of adversarial examples to tanh space. After transformation, x belongs to $(-inf, +inf)$,
140   which is more conducive to optimization.

141   ***DeepFool***

142   The DeepFool algorithm is proposed by Moosavi-Dezfooli et al. (Moosavi-Dezfooli et al., 2016),
143   which generates an adversarial perturbation of the minimum norm of the input sample through iterative
144   calculation. In each iteration, the DeepFool algorithm interferes with the image through a small vector. It
145   gradually pushes the images located within the classification boundary to outside the decision boundary
146   until a misclassification occurs. In addition, DeepFool aggregates the perturbations added in each iteration
147   to calculate the total perturbations. Its perturbations are minor than FGSM, and at the same time, the
148   classifier has a higher rate of misjudgment.

149   **Defense methods**

150   At present, the defense is mainly divided into two aspects: improving the classifier model's robustness
151   and preprocessing the input without changing the classifier model. Adversarial training (Goodfellow
152   et al., 2015) is currently a more effective defense method proposed by Goodfellow et al. They use
153   adversarial examples to expand the training set and train with the original samples to increase the model's
154   fit to the adversarial examples, thereby improving the robustness of the model. However, this increases
155   the calculation cost and complexity, and adversarial training has excellent limitations. When facing
156   adversarial attacks generated by different methods, the performance varies significantly.

157   Generally, the preprocessing process does not need to modify the target model, compared with
158   adversarial training and other methods, which is more convenient to implement. Moreover, it has a smaller
159   amount of calculation and can be used in combination with different models. For instance, Xie et al.
160   (Xie et al., 2017) propose to enlarge and fill the input image randomly. The entire defense process does

not need to be retrained and is easy to use. However, the results show that this method is only effective for iterative attacks such as *C&W* and DeepFool (Moosavi-Dezfooli et al., 2016), while for FGSM, the defensive effect of this single-step attack is inferior. They believe that this is due to the iterative attack to fitting the target model, resulting in low-level image transformation that can destroy the fixed structure of the adversarial disturbance. In addition, Liao et al. (Liao et al., 2018) regard adversarial perturbation as a kind of noise, and they designe a high-level representation guided denoiser (HGD) model to eliminate the adversarial disturbance of the input species. Das et al. (Das et al., 2017) use JPEG compression to destroy adversarial examples.

Similarly, Pixel Defend (Song et al., 2017) is a new method that purifies the image by moving the maliciously perturbed image back to the training data to view the distribution. Feature squeezing (Xu et al., 2017) is both attack-agnostic and model-agnostic. It can reduce the image range from [0, 255] to a smaller value, merge the samples corresponding to many different feature vectors in the initial space, and reduce the search space available to the opponent. Similar methods also include label smoothing (Warde-Farley and Goodfellow, 2016), which converts one-hot labels to soft targets. Besides, Zhang et al. (Zhang et al., 2021) propose a domain adaptation method, which gradually aligns the features extracted from the adversarial example domain with the clean domain features, making DNN more robust and less susceptible to spoofing by diverse adversarial examples.

OUR APPROACH

Motivation

The essence of adversarial examples is to deliberately add high-frequency perturbations to clean input samples and amplify the noise through deep neural networks so that the model gives the wrong output with high confidence. For example, when we input a clean image of a cat, add an imperceptible perturbation, the classifier will misclassify it as a leopard with high confidence. Through previous research, we have also learned that the classifier is robust to ordinary noise. Simultaneously, the adversarial perturbation in the adversarial example is very unstable and can be destroyed by some simple image transformation methods. According to the currently known image characteristics, we use image processing methods to eliminate the fixed structure of the adversarial perturbation before the adversarial example is input to the target. At the same time, to ensure the system's normal operation and the performance of the adversarial examples after converting the original image and the target model, we combine image compression and image restoration neural networks to form the entire defense model. This model can convert adversarial examples into clean images to resist adversarial example attacks without significantly reducing the quality of ordinary images.

Pixel depth reduce

An array of pixels represents a standard digital image in a computer, and each pixel is usually represented as a number with a specific color. Since two common representations are used in the test data set, they are 8-bit grayscale and 24-bit color. Grayscale images provide $2^8 = 256$ possible values for each pixel; we use k to represent the maximum range of pixel values. An 8-bit value represents a pixel's intensity, where 0 is black, 255 is white, and the average number represents different shades of gray. The 8-bit ratio can be expanded to display color images with separate red, green and blue channels and provides 24 bits for each pixel, representing $2^{24} \approx 16$ million different colors. The redundancy of the image itself offers many opportunities for attackers to create adversarial examples.

Compressed pixel bit depth can reduce image redundancy and destroy the fixed structure of adversarial examples in the input while retaining image information without affecting the image's accuracy on the classifier model. As shown in Figure 3, the defense capability is tested on the MNIST, Fashion-MNIST, and CIFAR-10 datasets. In the sub-pictures (a), (b), and (c), k refers to the maximum range of pixel value color depth. When ε is small, the attack intensity is low, and reducing each pixel's color depth can have an excellent defense effect. On the contrary, as the attack intensity continues to increase, the defense effect is also declining. At the same time, the situation becomes more complicated in the face of more complex data sets (such as Cifar-10). Although a higher compression rate can improve the defensive performance to a certain extent, it will also cause the loss of ordinary image information and reduce the prediction accuracy of the classifier model. Therefore, we need to repair the damaged image after compression.

(a) results on the MNIST dataset.

(b) results on the F-MNIST dataset.

(c) results on the CIFAR-10 dataset.

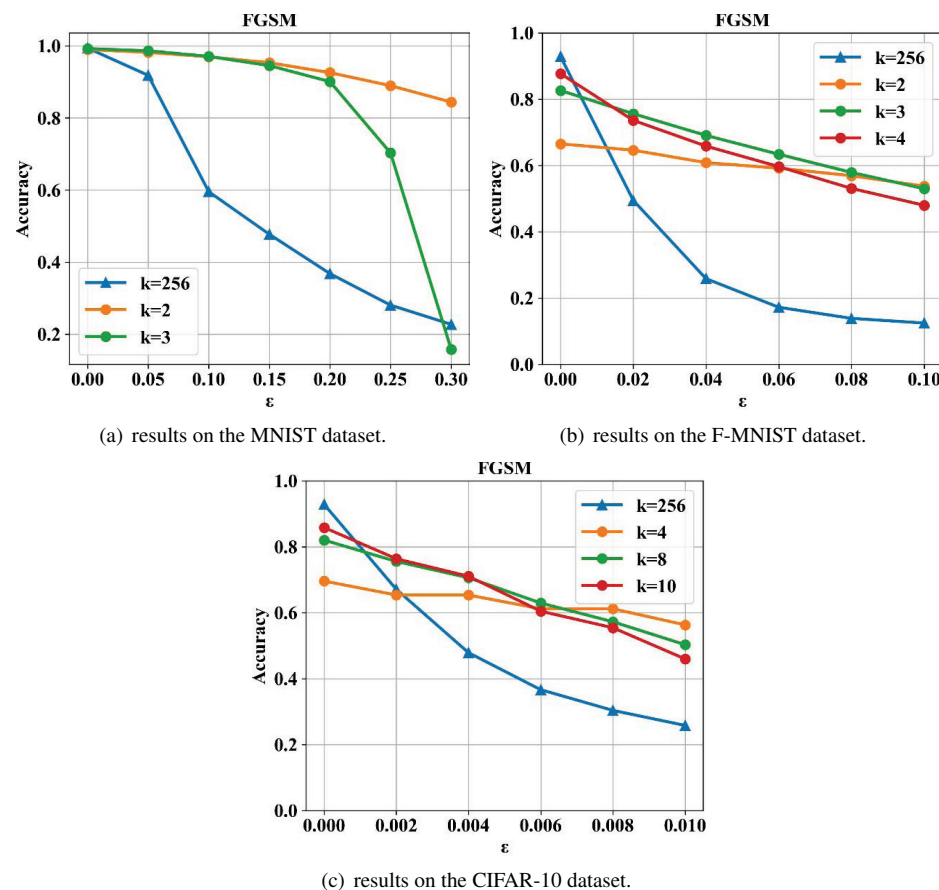



Figure 3. Changes in the defense effect of pixel compression on the three data sets.

Image reconstruction

Super-resolution (Mustafa et al., 2019) image reconstruction is a typical application in computer vision and deep learning development. Recently, super-resolution has made significant progress, and this technology is often used to reconstruct high-resolution images or repair damaged images. We also hope to use this deep neural network to repair compressed images. Generally, the low-quality images used in the training process of the image reconstruction network are obtained through down-sampling, blurring, or other degradation methods. In this paper, we first collect low-quality images by compressing the pixel depth and input them into the reconstruction network; then, we train the deep neural network learning ability to restore low-quality images to high-quality images.

Without loss of generality, the deeper the network and the more parameters, the better the performance for deep convolutional neural networks. Figure 4 shows the main process of applying the input image to reconstruct the defense model based on image compression. For reconstructing network structure, we refer to the excellent EDSR structure in the super-resolution image reconstruction network (Lim et al., 2017) to build a very deep neural network to ensure the recovery performance of the image. The chain-hopping structure in the network (He et al., 2016) can help us build a deeper network to obtain better performance without worrying about gradients' disappearance. The entire defense model is used to complete the conversion from adversarial examples to clean samples and ensure the quality of the reconstructed image. Without considering the disappearance of the gradient, to build a deeper network, we add the ResNet structure to the reconstruction network, and use the ReLU activation function and a 3×3 filter. In the reconstruction training process, we first train the low-multiple up-sampling model and then initialize the high-multiple up-sampling model with the parameters obtained in training. This will make the training time of the high-multiple upsampling model shorter and the training result better. Finally, we get a picture SR_img that eliminates the perturbation of adversarial examples.

In the experiment, we find that after compressing the high-strength adversarial example, the clas-

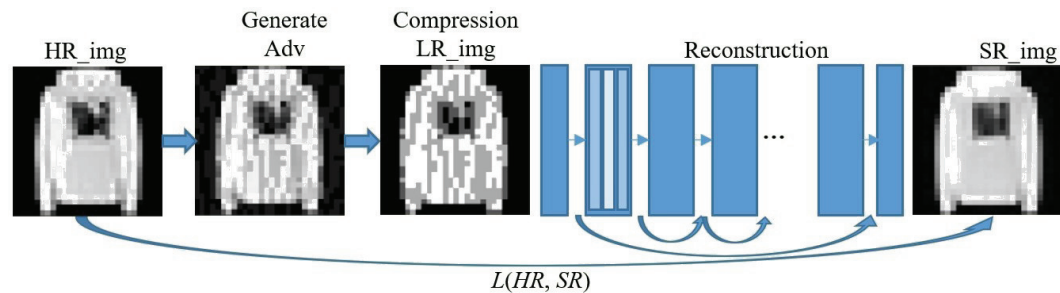



Figure 4. Implementation framework of our defense model.

sification result is different from the original image and the adversarial example. Because the high compression rate destroys the fixed structure of the adversarial perturbation, it also causes a certain amount of information loss. Adding adversarial examples in the training process can solve this problem well and improve the neural network's ability to repair compressed images. We use clean samples to generate adversarial examples during the training process and keep the total number of training sets unchanged. Then we use adversarial samples for training and use clean data sets as labels to narrow the gap between adversarial samples and clean samples. To prevent the network from overfitting and repair the compressed image of the adversarial example, we reduce the repair effect of the ordinary sample after compression to a certain extent.

To better reconstruct clean samples, we minimize the distance between the reconstructed *SR_img* and the original image *HR_img*. We use Mean Squared Error(MSE) to define the loss function of the CNN:

$$L(\theta) = \frac{1}{2N} \sum \|F(X,\theta) - Y\|^2 \tag{4}$$

where F is the image restoration network, X is the compressed image, θ is the network parameter, and Y is the original image.

After the training is completed, the reconstructed network has the ability to filter and fight noise. We add the reconstructed network model before the classifier that needs to be protected. When a batch of samples are input, they first pass through our reconstructed network model. If these input images include adversarial examples, their adversarial features will be destroyed, while normal samples will not be affected. In this way, we can turn the input into a clean sample to defend against adversarial attacks.

EXPERIMENT

In this section, we use experiments to verify the effectiveness of the proposed algorithm. The basic process of the experiment includes generating adversarial examples on different datasets and training multiple classifier models to test the performance and transferability of the defense model. In addition, we conduct a comprehensive theoretical analysis of the experimental results.

Experimental setup

In our experiments, we use three different image datasets: MNIST (LeCun et al., 1998), Fashion-MNIST (F-MNIST) (Xiao et al., 2017) and CIFAR-10 (Xiao et al., 2018). The MNIST and F-MNIST datasets both contain 60,000 training images and 10,000 test images. Each example is a 28×28 grayscale image associated with one label in 10 categories. The difference is that MNIST is a classification of handwritten numbers $0-9$, while F-MNIST is no longer an abstract symbol but a more concrete clothing classification. The CIFAR-10 dataset is a 32×32 color image associated with 10 category labels, including 50,000 training images and 10,000 test images. To prevent over-fitting, both the defense model and the classifier target model in this paper are trained by the training set. The classifier model's accuracy and the defense model's performance experiment are conducted in the test set.

To verify the generalization effect of the defense framework, this paper chooses FGSM, BIM, Deep-Fool and *C&W* four methods to generate different types of adversarial examples for defense testing. We preprocess the defense model and then input it into the classifier model to get the experimental results.

272 For FGSM and BIM, we use the L_∞ norm to control the perturbation's intensity by changing the size of ε.
273 Differently, we use the L_2 norm to implement the *C&W* model, and adjust the degree of perturbation by
274 controlling the maximum number of iterations. To preserve the original image information and eliminate
275 adversarial perturbations as much as possible, we set $k = 2$ (k denotes the maximum range of pixel value
276 color depths) on the MNIST dataset and $k = 4$ on the F-MNIST dataset.

Experiment results

278 In this section, the adversarial examples generated by FGSM, BIM, DeepFool, and *C&W* on different
279 datasets are applied to the defense framework of this paper. Simultaneously, in the training process, to
280 make the reconstructed network have selective noise reduction and generalization capabilities, we use
281 the FGSM with the most perturbation to generate adversarial examples and input them into the neural
282 network. Generally, simple images need to add a significant perturbation to be effective. In this paper, for
283 the MNIST dataset, the value of ε is up to 0.3; for the F-MNIST dataset, the value of ε is from 0 to 0.1;
284 for the Cifar-10 dataset, the value of ε is taken from 0 to 0.01. When ε is equal to 0.01, the adversarial
285 example is enough to produce a higher error rate on the target classifier model for the CIFAR-10 data set.
286 The results of each step of the defense experiment process are shown in the figure below.

287   From left to right, the different subgraphs in Figure 5 are the adversarial examples generated by
288 FGSM, BIM, DeepFool, and *C&W* attack methods, respectively; from top to bottom are normal examples,
289 adversarial examples, compression examples, and reconstructed examples. Figure 5(a) is the result of
290 working on the MNIST data set. It can be seen that only the pixel compression operation can eliminate
291 most of the adversarial perturbations, and the adversarial examples restore the accuracy of the classifier
292 model. In addition, the adversarial examples generated by different methods have different perturbation
293 levels to the image, and FGSM has the most massive perturbation level. When ε is 1.5, it has already had
294 a more significant impact on the image, and the human eye can already detect it, but our method can still
295 restore it to a clean sample. A few extreme adversarial examples become other classification results after
296 processing, as shown in the first column of Figure 5(a). Still, after the reconstruction of the network, the
297 recognition accuracy is also restored.

298   Figure 5(b) and (c) show the experimental results of the relatively complex of F-MNIST and CIFAR-10
299 data sets. Since pixel depth reduction is a lossy compression, choosing an appropriate compression level
300 can eliminate the adversarial perturbation of the input sample as much as possible while retaining the
301 necessary information. Generally, a slight loss of details does not affect the classifier model's correct
302 recognition of the image. The following experiment will specifically show the defense effect of different
303 data sets after processed by our defense framework under different attack intensities.

304   Figure 6(a), (b), (c) and (d) are the recognition accuracy rates of the model ResNet-50 with and without
305 defensive measures for different attack strengths (ε, iteration) on the MNIST dataset. Our algorithm is
306 compared with four different types of adversarial samples in defensive and non-defensive situations. After
307 the defense model processes the data set in this paper, the accuracy of the original image has almost no
308 change. Furthermore, in the face of different types of attacks from FGSM, BIM, DeepFool, and *C&W*,
309 the operation can eliminate adversarial perturbations in the input image. This is because the defense
310 model has certain image recovery capabilities, the MNIST image structure is relatively simple, and the
311 information is not easily damaged. For FGSM attacks, we can see that the accuracy can be restored from
312 20% to 97% under high attack intensity, BIM can be restored from 5% to 98%, DeepFool can be restored
313 from 0% to 98%, and *C&W* can be restored from 0% to 98%.

314   Figure 7(a), (b), (c) and (d) are the recognition accuracy rates of the model ResNet-50 with and
315 without defensive measures for different attack strengths (ε, iteration) on the Fashion-MNIST dataset. It
316 can be seen from Figure 7 that we have also achieved good results in the face of a slightly complicated
317 Fashion-MNIST defense model, i. e., the original image recognition accuracy rate drops by 4% after
318 preprocessing. For FGSM attacks, it recovery from 13% to 81%; for BIM, it recovery from 1% to 82%;
319 for DeepFool, it recovery from 0% to 88%; and for *C&W*, it recovery from 0% to 88%.

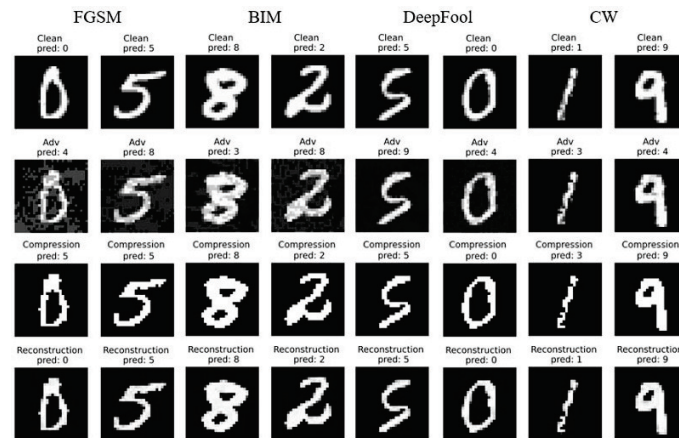
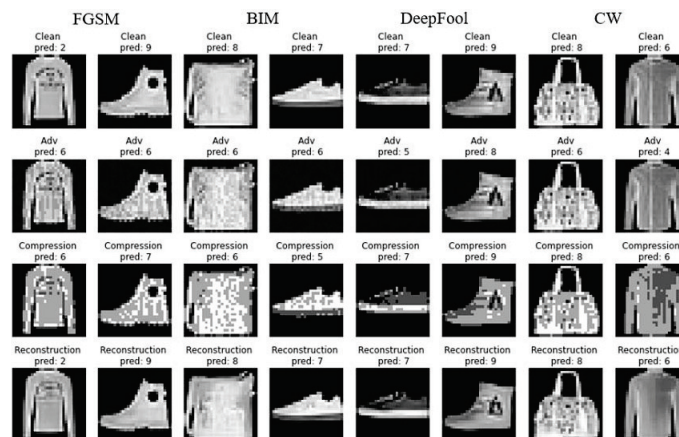
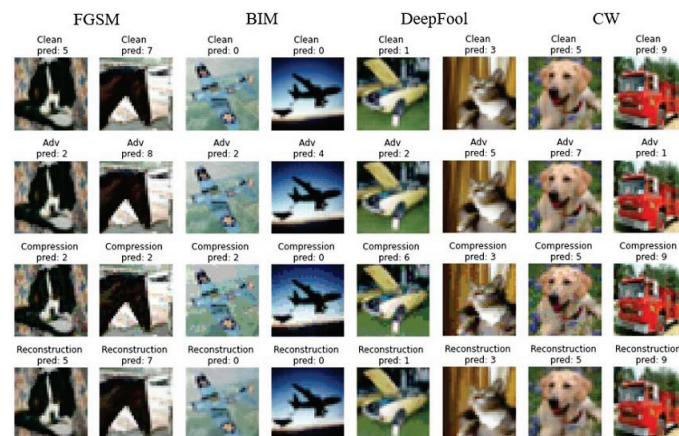
320   Figure 8(a), (b), (c) and (d) are the recognition accuracy rates of the model ResNet-50 with and
321 without defensive measures for different attack strengths (ε, iteration) on the Cifar-10 dataset. When
322 processing the three-channel color dataset CIFAR-10, we find that it is more complicated than the first two
323 single-channel grayscale image data sets. Mainly because it is difficult to balance the pixel compression
324 rate and the defense rate, which makes the defense effect appear to be reduced to a certain extent. It
325 can be seen from Figure 8, the ordinary sample has a loss close to 5% in accuracy after compressed and

(a) experimental results on MNIST dataset.

(b) experimental results on Fashion-MNIST datasets.

(c) experimental results on CIFAR-10 datasets.

Figure 5. Performance of defense models against multiple adversarial example attacks on different datasets.

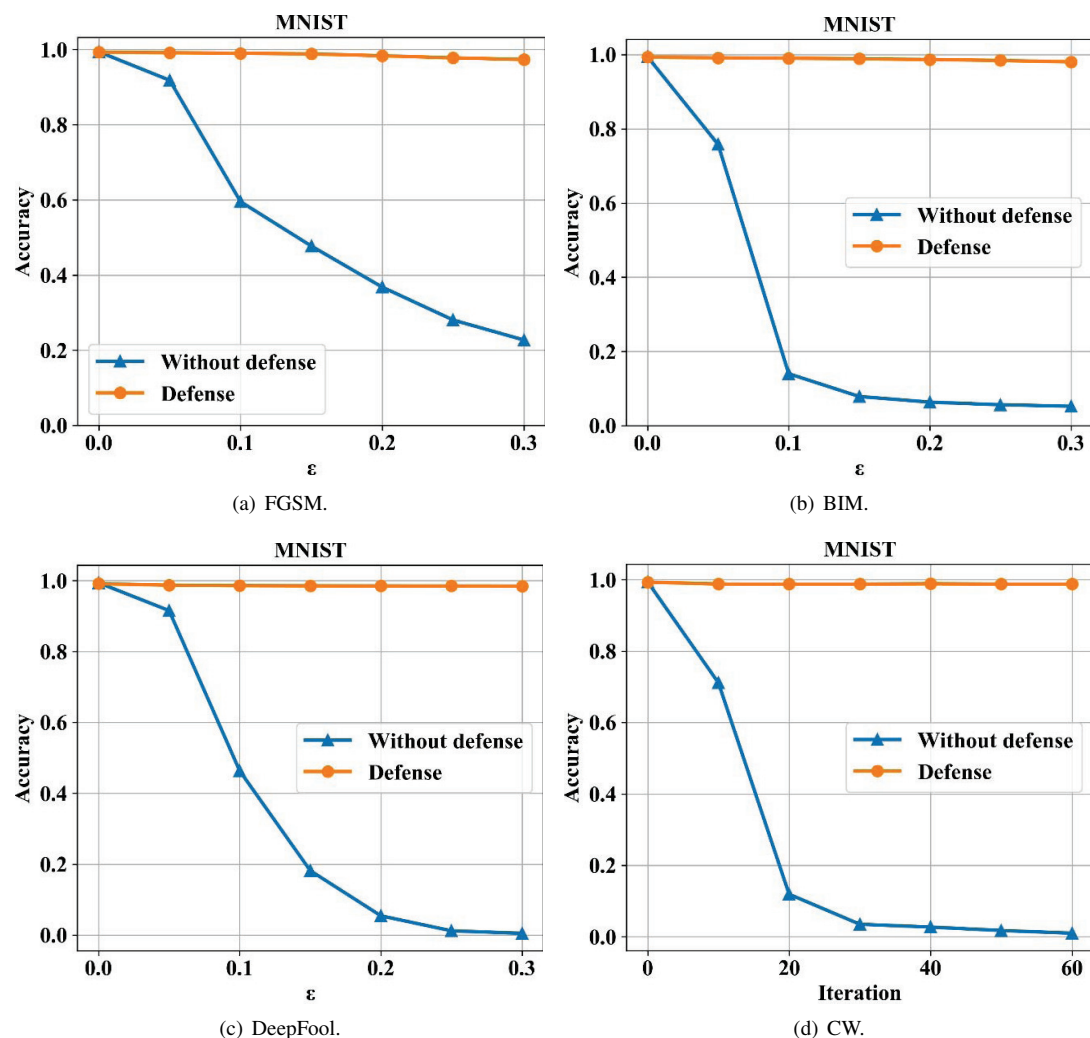



Figure 6. Performance of the defense model on the MNIST dataset.

reconstructed. For FGSM attacks, the defense model can restore the accuracy from 23% to 71%, BIM from 2% to 70%, DeepFool from 18% to 87%, and CW from 0% to 87%.

Defense Transferability

As a preprocessing method, we can combine different target models without modifying them. To verify the defense model's portability, we train three classifier models from weak to strong performance. They are: LeNet (LeCun et al., 1998), GoogLeNet (Szegedy et al., 2015), and ResNet101 (He et al., 2016). Besides, we combine the defense model trained with these three classifier models to test the defense performance.

Table 1 and Table 2 show in detail the experimental results of the transferability of the defense model. On the MNIST and Fashion-MNIST datasets, we take the median value of 0.15 and 0.05 for ε, respectively. Due to the performance difference of the target model, the effect will be slightly reduced when the defense model is combined with different models. However, it can still defend well against adversarial example attacks. Table 3 shows the transferability performance of our defense model combined with ResNet 50, ResNet101, and GoogLeNet on the data set Cifar-10. We take the median value of 0.005 for ε. The classification accuracy of the overall network model on the Cifar-10 data set has been reduced compared to the performance of the MNIST and F-MNIST data sets. This is because the Cifar-10 data set is relatively complex. In short, the classification accuracy of the network model with defense is much higher than the network model without defense when facing different attacks.

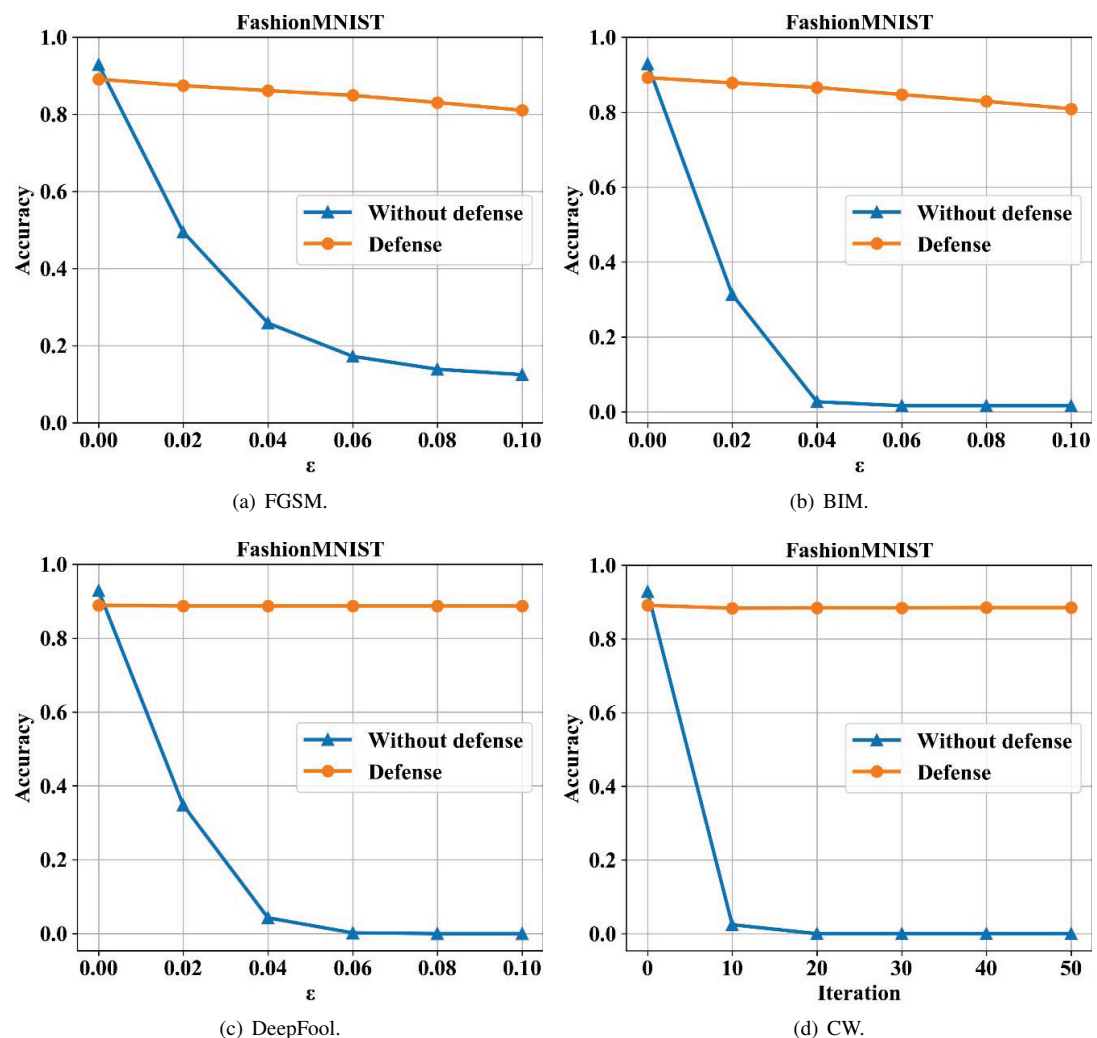



Figure 7. Performance of the defense model on the F-MNIST dataset.

Performance comparison between similar defense models

This section uses four methods (FGSM, BIM, DeepFool, and *C&W*) on the Fashion-MNIST dataset to generate two adversarial examples of different strengths for the ResNet50 target model and conduct defense tests. To better compare with other classic methods and verify the effectiveness of our approach, all experiments use the same dataset, target model, and related parameter settings as other methods. As shown in Table 4, our method performs best compared with other methods under attack models such as BIM, DeepFool, and *C&W*.

Although the ComDefend method is better at preserving the original image information, it adds Gaussian noise during training to improve the network's ability to resist noise. The defense effect of some attacks, such as BIM, is not ideal. The impact of adding an FGSM attack is only acceptable in the case of FGSM adversarial examples, and it performs poorly for adversarial examples generated by other methods. In general, although the direct pixel depth reduction has made a certain sacrifice in image information preservation, the confrontation samples generated in the face of different attacks in the above experiments can all play a good defense effect. Therefore, to the best of our knowledge, our method can effectively defend against adversarial example attacks.

CONCLUSION

Finding a robust defense method for adversarial examples is an open problem, and many researchers have carried out work in this area. This paper proposes an image compression and reconstruction defense

Table 1. The performance of the defense model combined with LeNet and GoogLeNet on MNIST.

Dataset	Network	Clean	FGSM	BIM	DeepFool	CW
MNIST (ε=0.15)	ResNet50(no defense)	99%	48%	9%	18%	3%
	ResNet50(defense)	99%	98%	99%	99%	99%
	LeNet(no defense)	99%	38%	2%	9%	3%
	LeNet(defense)	99%	98%	98%	98%	98%
	GoogLeNet(no defense)	99%	48%	26%	19%	15%
	GoogLeNet(defense)	99%	98%	97%	98%	98%

Table 2. The performance of the defense model combined with ResNet and GoogLeNet on F-MNIST.

Dataset	Network	Clean	FGSM	BIM	DeepFool	CW
F-MNIST (ε=0.05)	ResNet50(no defense)	93%	22%	2%	0%	0%
	ResNet50(defense)	89%	85%	85%	89%	89%
	GoogLeNet(no defense)	90%	35%	18%	2%	0%
	GoogLeNet(defense)	90%	81%	84%	88%	88%
	ResNet101(no defense)	93%	20%	2%	0%	0%
	ResNet101(defense)	89%	83%	84%	88%	88%

Table 3. The performance of the defense model combined with ResNet and GoogLeNet on Cifar-10.

Dataset	Network	Clean	FGSM	BIM	DeepFool	CW
CIFAR-10 (ε=0.005)	ResNet50(no defense)	84%	64%	23%	20%	39%
	ResNet50(defense)	79%	68%	59%	72%	72%
	GoogLeNet(no defense)	98%	36%	34%	35%	0%
	GoogLeNet(defense)	94%	51%	52%	61%	60%
	ResNet101(no defense)	84%	64%	24%	22%	43%
	ResNet101(defense)	80%	69%	63%	74%	74%

Table 4. The result of comparisons with other defensive methods(F-MNIST).

Network	Methods	Clean	FGSM	BIM	DeepFool	CW
Resnet50	Normal	**93%/93%**	38%/24%	00%/00%	06%/06%	00%/00%
	Adversarial FGSM	**93%/93%**	85%/85%	51%/00%	63%/07%	67%/21%
	Adversarial BIM	92%/91%	84%/79%	76%/63%	82%/72%	81%/70%
	Feature Squeezing	84%/84%	70%/28%	56%/25%	82%/72%	83%/83%
	Pixel Defend	89%/89%	87%/82%	85%/83%	83%/83%	88%/88%
	ComDefend	**93%/93%**	**89%/89%**	70%/60%	88%/88%	88%/89%
	Our Method	89%/89%	87%/86%	**87%/86%**	**90%/89%**	**89%/89%**

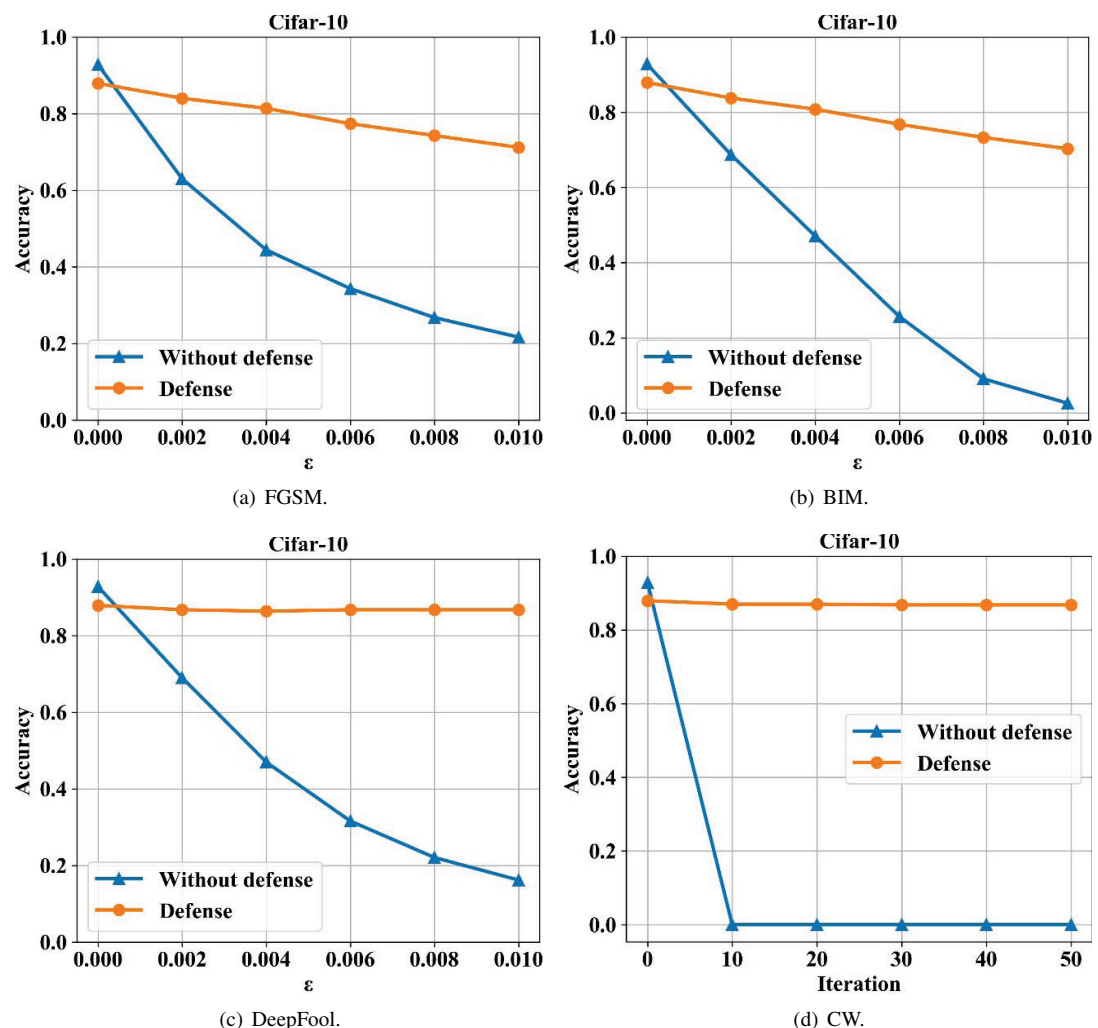



Figure 8. Performance of the defense model on the CIFAR-10 dataset.

framework to defend against adversarial example attacks based on the redundancy of images and the sensitivity of adversarial examples. We compress the pixel bit depth in the image to destroy the adversarial perturbation of the image and then use DNN to repair the image. On the premise of ensuring the image quality, the adversarial examples are converted into clean samples to achieve the purpose of defense. In addition, this method can be easily combined with other defense methods without modifying the target classifier model. Extensive experiments have been applied to the three real datasets of MNIST, F-MNIST, and CIFAR-10, showing the superiority of the proposed method over some classic techniques to defend against adversarial examples, i.e., the defensive framework we designed can resist different attacks. However, due to limited knowledge and personal abilities, many issues need further research. We will study how to better balance the compression rate of complex images and preserve adequate information and verify the method's effectiveness on more complex datasets.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant 61572034, by the Major Science and Technology Projects in Anhui Province under Grant 18030901025, by the Natural Science Foundation of Anhui Province of China under Grant 2008085MF220.

REFERENCES

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE.

Das, N., Shanbhogue, M., Chen, S., Hohman, F., Chen, L., Kounavis, M. E., and Chau, D. H. (2017). Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*.

Deng, Y., Zheng, X., Zhang, T., Chen, C., Lou, G., and Kim, M. (2020). An analysis of adversarial attacks and defenses on autonomous driving models. In *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10. IEEE.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. (2018). Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples.

Hameed, M. Z., György, A., and Gündüz, D. (2020). The best defense is a good offense: Adversarial attacks to avoid modulation detection. *IEEE Transactions on Information Forensics and Security*, 16:1074–1087.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Jia, X., Wei, X., Cao, X., and Foroosh, H. (2019). Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6084–6092.

Jin, W., Li, Y., Xu, H., Wang, Y., Ji, S., Aggarwal, C., and Tang, J. (2021). Adversarial attacks and defenses on graphs. *ACM SIGKDD Explorations Newsletter*, 22(2):19–34.

Karen, S. and Andrew, Z. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, Conference Track Proceedings*.

Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial examples in the physical world.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., and Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787.

Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144.

Moosavi-Dezfooli, S., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582.

Mustafa, A., Khan, S. H., Hayat, M., Shen, J., and Shao, L. (2019). Image super-resolution as a defense against adversarial attacks. *IEEE Transactions on Image Processing*, 29:1711–1724.

Ren, K., Zheng, T., Qin, Z., and Liu, X. (2020). Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360.

Singla, S. and Feizi, S. (2020). Second-order provable defenses against adversarial attacks. In *International Conference on Machine Learning*, pages 8981–8991. PMLR.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. (2017). Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *Computer Science*.

Thys, S., Van Ranst, W., and Goedemé, T. (2019). Fooling automated surveillance cameras: adversarial

432   patches to attack person detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision*
433   *and Pattern Recognition Workshops*, pages 0–0.

434   Warde-Farley, D. and Goodfellow, I. J. (2016). 11 adversarial perturbations of deep neural networks.
435   *Perturbations, Optimization, and Statistics*, 311.

436   Xiao, C., Li, B., Zhu, J., He, W., Liu, M., and Song, D. (2018). Generating adversarial examples with
437   adversarial networks. pages 3905–3911.

438   Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking
439   machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

440   Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. (2017). Mitigating adversarial effects through
441   randomization. *arXiv preprint arXiv:1711.01991*.

442   Xu, H., Ma, Y., Liu, H., Deb, D., Liu, H., Tang, J., and Jain, A. K. (2020). Adversarial attacks and
443   defenses in images, graphs and text: A review. *International Journal of Automation and Computing*,
444   17(2):151–178.

445   Xu, W., Evans, D., and Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural
446   networks. *arXiv preprint arXiv:1704.01155*.

447   Zhang, H. and Wang, J. (2019). Defense against adversarial attacks using feature scattering-based
448   adversarial training. *Advances in Neural Information Processing Systems*, 32:1831–1841.

449   Zhang, X., Wang, J., Wang, T., Jiang, R., Xu, J., and Zhao, L. (2021). Robust feature learning for
450   adversarial defense via hierarchical feature alignment. *Information Sciences*, 560:256–270.