applied sciences

MDPI

Article

Defense Against Adversarial Attacks in Deep Learning

Yuancheng Li and Yimeng Wang *

School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China; yuancheng@ncepu.cn

* Correspondence: 1162227078@ncepu.edu.cn

Received: 25 November 2018; Accepted: 21 December 2018; Published: 26 December 2018

Abstract: Neural networks are very vulnerable to adversarial examples, which threaten their application in security systems, such as face recognition, and autopilot. In response to this problem, we propose a new defensive strategy. In our strategy, we propose a new deep denoising neural network, which is called UDDN, to remove the noise on adversarial samples. The standard denoiser suffers from the amplification effect, in which the small residual adversarial noise gradually increases and leads to misclassification. The proposed denoiser overcomes this problem by using a special loss function, which is defined as the difference between the model outputs activated by the original image and denoised image. At the same time, we propose a new model training algorithm based on knowledge transfer, which can resist slight image disturbance and make the model generalize better around the training samples. Our proposed defensive strategy is robust against both white-box or black-box attacks. Meanwhile, the strategy is applicable to any deep neural network-based model. In the experiment, we apply the defensive strategy to a face recognition model. The experimental results show that our algorithm can effectively resist adversarial attacks and improve the accuracy of the model.

Keywords: adversarial attacks; deep learning; face recognition; Wasserstein generative adversarial networks (W-GAN); denoiser; knowledge transfer

1. Introduction

Deep learning occupies a central place in the development of machine learning and artificial intelligence. Neural networks are currently used in many areas to solve complex problems, such as reconstructing the brain circuit [1], analyzing mutations in DNA [2], and analyzing particle accelerator data [3]. In the field of computer vision, it has become the main force in autonomous driving [4], monitoring [5], and security application [6], etc. However, even though deep networks have demonstrated success in dealing with complex problems, recent research suggests that they are vulnerable to slight disturbances in the input [7]. In the classification problem, this slight disturbance can cause the classifier to output an erroneous result. Disturbances in pictures are often too small to be perceived by humans, but they completely fool the deep learning model. Adversarial attacks have created a series of threats to the application of deep learning in practice. For example, in face recognition, the attacker is recognized as a normal user to obtain the user's private data. In automatic driving, a roadside icon recognition error causes the control algorithm to make an incorrect judgment and generate erroneous behavior. Therefore, it is necessary to study the defense against adversarial samples.

Common adversarial attacks include limited-memory BFGS (L-BFGS) [7], fast gradient sign method (FGSM) [8], and Jacobian-based saliency map approach (JSMA) [9]. L-BFGS misleads neural networks by adding small amounts of undetectable disturbance to the image. FGSM obtains attack samples by adversarial training the model. JSMA proposes a defensive method of limiting the L_0

norm by changing only the values of a few pixels instead of disturbing the entire image. At present, there are three main directions in adversarial attack defense: (1) modifying the training process or modified input samples during the learning process; (2) modifying the network, such as adding more layers/sub-networks, changing loss/activation functions, etc.; (3) use an external model as an additional network to classify the unknown samples.

Literature [8] proposed a defensive method of brute adversarial training. The network's robustness can be improved by continuously training the model with new types of adversarial samples. To ensure effectiveness, the method requires hard adversarial samples. The network architecture must strong enough to effectively learn the image features. However, Moosavi–Dezfooli [10] pointed out that no matter how many types of anti-samples are contained, there always exists new types of adversarial samples that can deceive the network. Literature [11] used model adaptation to improve the robustness of the network. Literature [12] used the extreme learning machine technique to improve the generalization ability of the system. Considering that most training images are in Joint Photographic Experts Group (JPG) format, Dziugaite [13] used JPG image compression to reduce the impact of anti-disturbance on model performance. However, this method is only effective for partial attack algorithms, and compressing images will also reduce the accuracy of normal classification. Luo et al. proposed to use the 'foveation' mechanism [14] to defend against the adversarial attacks generated by L-BFGS and FGSM. The assumption of the method is that the image distribution is robust to the transformation variation, and the disturbance does not have this property. However, the universality of this method has not been proven. Xie et al. found that introducing random re-scaling to training images [15] can reduce the intensity of attacks. Other methods of changing images include random padding, and image enhancement. Ross and Doshi-Velez used input gradient regularization to improve the robustness against attacks [16]. This method performs well combined with brute adversarial training, but the computational complexity is too high. Literature [17] proposed a transfer learning algorithm based on bi-directional long short-term memory (BLSTM) recurrent neural networks, which can resist the disturbance to a certain extent. Nayebi and Ganguli used a highly nonlinear activation function similar to that of nonlinear dendrites in biological brains [18] to defend against adversarial attacks. Dense Associative Memory model [19] is also based on a similar mechanism. Cisse et al. used the global Lipschitz constant [20] to maintain the Lipschitz constant of each layer to remove the adversarial sample's interference. Gao et al. added a layer of network to be trained with adversarial samples before the classification layer. The theory of the method holds that the most significant features are contained in the top layers [21]. Akhtar et al. added a separately trained network to the original model [22], achieving immunity to the attack samples without adjusting parameters. But this approach increases the complexity of the network and slows down the model. There are also research works based on detection of adversarial attacks. These methods can help detect whether the input image is adversarial but cannot correctly obtain clean images and get the correct classification result. For example, SafetyNet [23], detector subnetwork [24], exploiting convolution filter statistics [25], and additional class augmentation [26].

In this paper, we propose a defense strategy that can resist against adversarial attacks. By carefully comparing the image details, we found that the adversarial sample appeared to be generated by adding noise to the original image. Therefore, to defend against this type of attack, we propose a deep denoising network based on U-Net, which consists of a contracting path and an expanding path for accurate location, to remove the noise. To train the denoiser, we use Wasserstein generative adversarial networks (W-GAN) to reconstruct the noise and augment the original dataset. To resist the nuances of the denoised image and the original image, we propose a new training method based on knowledge transfer, which will improve the robustness of the model. When the image has slight disturbances, it can still be recognized correctly. As a result, the model's resilience is further improved. Our proposed strategy does not affect the original network structure and maintains the speed of the network.

The main contributions of this paper:

(1) A noise reconstruction method based on W-GAN is proposed to reconstruct the noise data delicately. The trained W-GAN is used to extend the adversarial sample dataset for training the deep denoising network.

(2) A deep denoising network based on U-Net is proposed. The attack samples can obtain "clean" images through the denoising network, which greatly improves the ability to resist against the attacks.

(3) A model training method based on knowledge transfer is proposed. The method can improve the robustness of the model and further improve the resilience and accuracy of the model.

(4) We apply the overall defensive strategy to the face recognition problem. Through experiments, it can be shown that our proposed method can greatly reduce the success rate of attacks and improve the accuracy of the model.

The rest of our paper is organized as follows: in Section 2, we propose a noise reconstruction algorithm GNR based on a generative adversarial network (GAN); in Section 3 we propose a deep denoising network UDDN based on U-Net; in Section 4 we propose a training method based on knowledge transfer; in Section 5 we conduct experiments and analyze experimental results; and in Section 6 we conclude our paper.

2. Noise Reconstruction Algorithm Based on GAN: GNR

By observing the local part of the attack image, we found that the attack samples appeared to be generated by adding noise to the original samples. To defend against this form of attack, we usually need sufficient data to train the model to remove noise. However, it is very complicated and time-consuming to generate the adversarial image for each image. To augment the dataset with only a few image pairs, we need to train the model to reconstruct noise. The noise can be obtained by subtracting the adversarial image and the clean image. Based on the constructed real noise dataset, we train GAN to model the noise and learn the distribution of noise. The trained GAN generates enough noise data, which is added to the clean image to obtain new adversarial images.

2.1. W-GAN

GAN is a deep generation network, the essence of which is to make the two distributions as close as possible. Therefore, the distribution of the generated data is as close as possible to the original data distribution. However, GAN has many problems, such as training difficulties, loss function of generators and discriminators unable to indicate training processes, and lack of diversity in generating samples. W-GAN has solved these problems to some extent. Compared to the original GAN algorithm, W-GAN solves these problems by introducing the Wasserstein distance. The mathematical representation of the Wasserstein Distance is shown in Equation (1).

$$\mathrm{W}(P_{data}, P_G) = \max_{\mathrm{D}\in 1-\mathrm{Lipschitz}} \left\{ E_{x\sim P_{data}}[\mathrm{D(x)}] - E_{x\sim P_G}[\mathrm{D(x)}] \right\} \tag{1}$$

In Equation (1), P_{data} represents raw data, and P_G represents data generated by the generator. D(x) represents the output of the discriminator and satisfies 1-Lipschitz. The role of 1-Lipschitz is to limit the change of D to be more gradual, so that the generator tends to generate more diverse images, increasing the diversity of generated images.

To facilitate the training of the actual model, W-GAN adopts a method of increasing the penalty to optimize the following goal, as shown in formula (2).

$$\mathrm{W}(P_{data}, P_G) = \max_{\mathrm{D}} \{ E_{x\sim P_{data}}[\mathrm{D(x)}] - E_{x\sim P_G}[\mathrm{D(x)}] - \lambda E_{x\sim P_{penalty}}[(\parallel \nabla_x D(x) \parallel -1)^2] \} \tag{2}$$

where $P_{penalty}$ represents the distribution of the input x. The points sampled in the P_{data} and P_G data are connected, and the point randomly sampled on the line is taken as the point of $P_{penalty}$. In this way, P_G is pulled to P_{data}, and the added penalty ensures that D is gently changing. The ideal D should be as close as possible to P_{data} and as far as possible from P_G. Therefore, the closer $\parallel \nabla_x D(x) \parallel$ is to one,

the better it is. Compared to the JS divergence in the original GAN, the Wasserstein distance is a better measure of distance that can ultimately be translated into an optimization problem.

2.2. Network Structure

For defending the adversarial attack, we use the generative adversarial network to model the noise. We first combine the original image and the attack image into a one-to-one sample pair. Then, based on the sample pair, we subtract the original image from the original image to get true noise data. Finally, by training GAN with true noise dataset, we obtain the distribution of the noise samples. As a framework of the generated model, GAN has the ability to learn complex distribution. More importantly, the GAN can generate noise by forward propagation in the generator network without involving another component. In our algorithm, the specific network structure of GAN is shown in Figure 1.

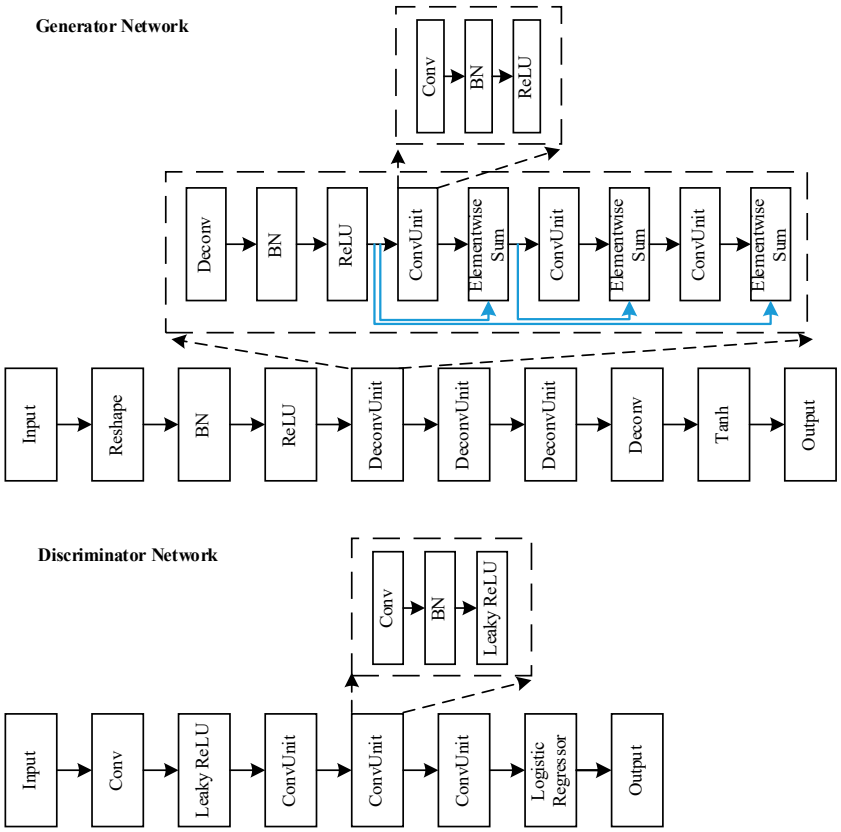

Figure 1. Generative adversarial network (GAN) structure. The whole structure consists of a generator network and a discriminator network. The generator consists of three DeconvUnits and the discriminator has three ConvUnits. The output of the generator is the input of the discriminator.

In the generator network, the size of the second to fourth filters are set to 5×5, which are 256, 128, and 64. The number of filters is equal to the number of channel outputs. The generator network consists of three DenconvUnit modules, each containing one deconvolution block and three convolution blocks. Each convolution or deconvolution layer is followed by a batch normalization layer and a ReLU layer. The connection in the generator network prevents the problem of gradient dispersion and degradation during network training. The generator's input is a random noise that satisfies the distribution. The output of the generator will be passed to the authentication network as input. In the discriminator network, the first to fourth filters are 64, 128, 256, and 512, respectively. The discriminator network consists of three ConvUnits, each of which includes a convolutional layer, followed with a batch normalization layer and a Leaky ReLU layer. The output of the discriminator is

the probability that the noise is true. The Wasserstein distance is used to define the loss function of the model, which is shown in Equation (3).

$$L_{GAN} = \underset{\tilde{x}\sim P_g}{E}[F(\tilde{x})] - \underset{x\sim P_r}{E}[F(x)] + \lambda \underset{\bar{x}\sim P_{\bar{x}}}{E}[(\| \nabla_{\bar{x}} F(\bar{x}) \|_2 - 1)^2] \quad (3)$$

In Equation (3), $\tilde{x}$ represents the generated noise, x represents the real noise, and $\bar{x}$ represents the noise randomly selected on the line between the generated noise and the real noise. $F(\cdot)$ represents the output of the discriminator. P_g represents the distribution of the generator, P_r represents the distribution of the real noise samples, and $P_{\bar{x}}$ represents the distribution of the samples of the random sampling between the real noise and the generated noise. The added penalty can help generate noise that tends to be true to noise and helps the generator generate more diverse noise. As the training progresses, the noise distribution generated by the generator will continue to approach the true noise distribution. We can directly generate noise through the trained generator network and realize the reconstructing of the noise in the adversarial sample.

3. Deep Denoising Network Based on U-Net: UDDN

To remove the noise on the adversarial image, we train a deep denoising network UDDN using the expanded adversarial sample dataset. The input of the network is adversarial images and the output is clean images that have already removed the noise. In this section, we improved the original denoiser DAE and proposed a new denoiser based on U-Net.

3.1. U-Net

U-Net is a variant of convolutional neural network and is an improved algorithm based on Fully Convolutional Networks (FCN). The U-Net's entire network consists of two parts: a contracting path and an expanding path. The contracting path is used to capture the context information in the image by gradually reducing the spatial dimension of the pooling layer, while the expanding path is to accurately locate the segmented image by expanding spatial dimensions and gradually recovering the details. To reconstruct the details of the target, the U-Net combines the high-pixel features extracted from the contracting path with the new feature map during the upsampling process. This feature fusion method can preserve important feature information in the process of downsampling. The basic module of a U-Net is shown in Figure 2.

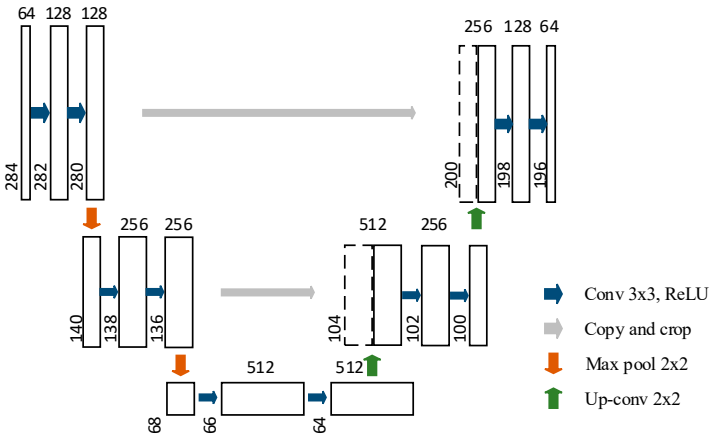

Figure 2. U-Net block. The U-Net consists of a contracting path and an expanding path. The horizontal connection combines the features of the upsampling and downsampling processes.

In the contracting path, every two 3×3 of the unpadded convolutional layers are followed by a 2×2 maximum pooling layer with a step size of 2. Each convolutional layer is followed by a ReLU

layer, which is used to downsample the original image. In addition, each downsampling doubles the number of feature channels. In the deconvolution of the expending path, each block consists of a 2 × 2 convolution layer and two 3×3 convolution layers. The feature map of each block in the expanding path is added to by the output of corresponding block in contracting path. An important feature of U-Net is that it can basically convolve images of any size, especially arbitrarily large ones.

3.2. Network Structure

The denoising autoencoder (DAE) is one of the popular denoising models. A DAE was used as a multi-layered perceptual network to defend against adversarial attacks. Experiments of defensive strategies are often performed on some relatively simple datasets, such as the MNIST dataset. For high resolution images, the strategy does not work well. DAE has a bottleneck structure between the encoder and the decoder. The network structure may limit the propagation of fine-scale information required to reconstruct high-resolution images. To overcome this problem, we incorporated the idea of U-Net in the DAE. In our method, the decoder layer can fuse the feature map of the encoder layer, which can better locate and recover the details of the image. Unlike U-Net, the decoder layer fuses feature images of the same scale at the encoder layer. The learning target of UDDN is to obtain noise instead of reconstructing the entire image, which is different from DAE. This residual learning method can help to train the model more easily and obtain more accurate noise data. By subtracting the noise from the adversarial image, the clean image can be obtained. The specific structure of UDDN is shown in Figure 3.

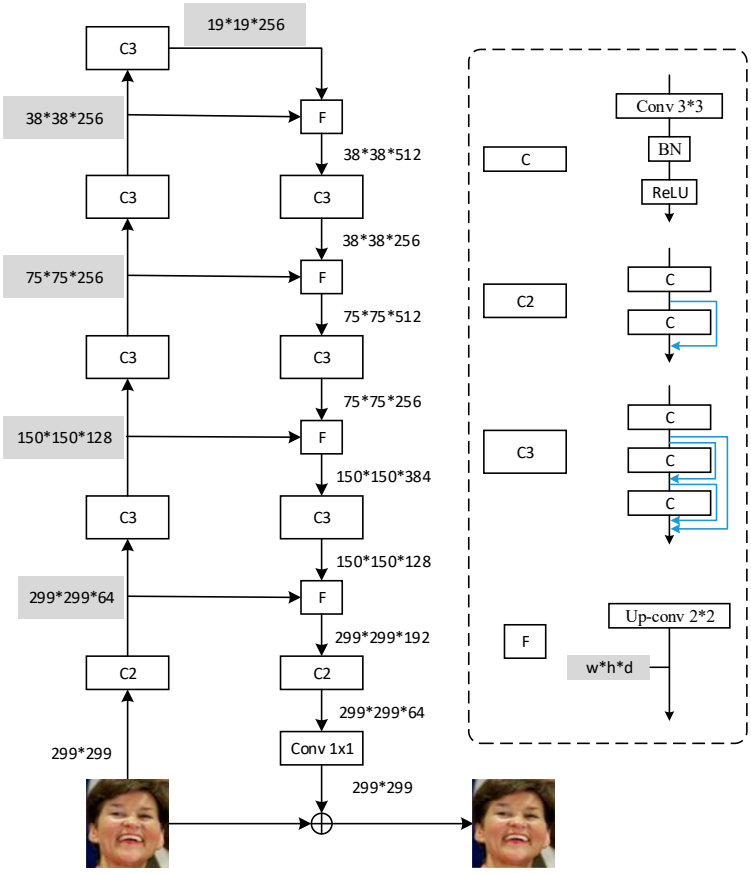

Figure 3. Deep denoising neural network (UDDN) network structure. The structure consists of an encoder and a decoder. The output of the decoder is added to the adversarial images to obtain the denoised image. The whole model is composed of C2, C3, and F units, the structures of which are shown on the right side of the figure.

As can be seen from Figure 3, a DAE can be obtained by removing the horizontal connection in UDDN. C stands for a basic module that consists of a three-layer network. The first layer is a 3×3 convolution followed by a BN layer and a ReLU activation function layer. Ck stands for k consecutive Cs. In C2, we added a connection at the second C, forming a residual structure. In C3, we added numerous connections at the second and third C positions, forming a structure similar to the densenet block. The F module is the same fusion module as U-Net, which combines the same-scale feature map of the encoder layer and the decoder layer into a new feature map. The overall network also consists of a contracting path and an expanding path. The encoder consists of five modules, one C2 and four C3. The step size of the first convolutional layer in C3 is two, and step size in other layers is one. The decoder consists of four modules and a 1*1 convolutional layer. Each module receives a feedback input from the decoder and a lateral input from the encoder. It first upsamples the feedback input to the same size as the horizontal input. Then, Ck processes the concatenation of the feedback input and the lateral input. From top to bottom, three C3 and one C2 were used. The output of the last module is converted to negative noise $-\mathrm{d}\hat{x}$ by a 1×1 convolutional layer. x^* represents the noise image, and $\hat{x}$ represents the output of the network. Then, the final output is the sum of the negative noise and the input image, as shown in Equation (4).

$$\hat{x} = x^* - d\hat{x} \tag{4}$$

The difference between the adversarial image and the clean image is negligible. However, this small perturbation is gradually amplified by the deep neural network and produces erroneous predictions. Even though the noise reducer can significantly suppress pixel-level noise, the residual noise can still distort the response of the target model. To overcome this problem, we replaced the pixel-level loss function with the reconstruction loss of the model output. More specifically, given a target neural network, we extracted the representation of x and x from the Lth layer, and defined the loss function as the L1 norm of their differences:

$$\mathrm{L} = \| f_l(\hat{x}) - f_l(x) \| \tag{5}$$

The supervisory signals come from certain top layers of the model, as well as guidance information related to image classification. Here, we define $l = -1$, which represents the layer index of the layer before the last softmax layer, which is the logits layer. This kind of denoiser can be referred to the one guided by logits. Therefore, the loss function calculates the difference between the two logits respectively activated by x and $\hat{x}$. In addition to this, l can be set to -2, which represents the index of the last convolutional layer. The output of this layer will be directly input to the linear classification layer, so it is more relevant to the classification target than other convolutional layers. This kind of denoiser can be referred to the one guided by feature. Then, the corresponding loss function is called perceptual loss or feature matching loss. In both ways, high-level information of the image can be obtained. Feature mapping provides more supervised information, while logits directly represents classification results. The model we propose is an unsupervised model, which does not need real labels during the training process.

4. Defense Mechanism Based on Knowledge Transfer

The adversarial image processing by denoising network as the input of the model can greatly decrease the success rate of the adversarial attack. However, the image after denoising is still different from the original image. This difference may still cause the model to make a wrong judgment. To improve the model's ability to tolerate this part of the disturbance, we propose a new strategy for training models. This strategy can help the model improve its resilience against sample attacks and improve the accuracy of the model. At the last, the overall defense mechanism is introduced in detail.

4.1. Knowledge Distillation

Knowledge distillation is one of the methods of model compression. It can train a small high-precision model under existing external conditions. The main idea is to use a teacher model to train a student model. The output of the teacher model often comes with some additional information, which is referred to dark knowledge. Compared with the ground truth, the result of the teacher model refers to the probability distribution, which represents correlation of samples with other types of samples. This additional information reflects a certain law learned by the classifier, through which the similarity of the sample to other types of samples can be calculated. The output of the teacher model is easier learned by student model than the ground truth, and the learning target is closer to the real situation. In the process of training the student model, the cross-entropy loss function is used to calculate the loss of the model, as shown in Equation (6).

$$\mathrm{L} = \mathrm{KL}(P_t||P_s) + \lambda \mathrm{KL}(P_t^T||P_s^T) \tag{6}$$

$$P^T(x) = softmax^T(x) = \frac{e^{\frac{Z_i(x)}{T}}}{\sum_j e^{\frac{Z_i(x)}{T}}} \tag{7}$$

where P_t represents the output of the teacher model, P_s represents the output of the student model, KL $(\cdot)$ represents the cross-entropy loss function, and λ is the harmonic parameter. T is an additional parameter in softmax and $\mathrm{T} \geq 1$. As T increase, the distribution of the output is more scattered and smoother. When T = 1, it degenerates into softmax.

4.2. Training Method Based on Distillation

By observing the denoised image and the original image, we found there are still subtle differences between the two images. The differences cause the denoised image to be recognized incorrectly. To further improve the ability to resist adversarial samples, we need to improve the robustness of the model to denoised pictures. For improving the applicability of the defense strategy, our proposed strategy should minimize the modification of the model network structure and maintain the speed of the network. After applying the defense strategy, the model could accurately identify the denoised image that is very close to the original image and maintain the accuracy of the recognition. We propose a training strategy based on Distillation. The specific strategy is shown in Figure 4.

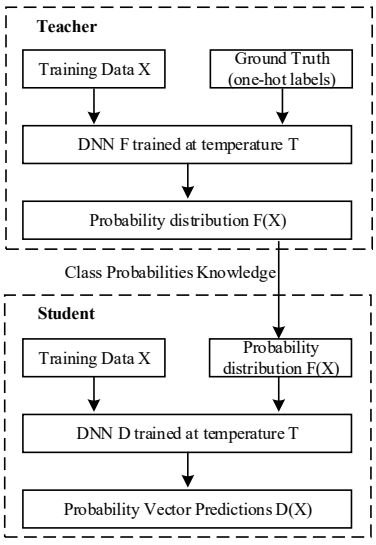

Figure 4. Defense mechanism based on Distillation. The mechanism consists of a teacher model and a student model. The student is trained with the output classification probability of the teacher.

In our proposed defense strategy, we trained the student model by using the teacher model. Teacher and Student model have the same network structure. We used the denoised image and the corresponding ground truth to train the teacher model, where the ground truth refers to the one-hot label. The Student model was then trained using the image and the corresponding output of the Teacher model, which represents the correlation of the image with other types of images and finer image information. The output probability was used to train student model, which can improve the robustness and accuracy of the model. After the training, the student model is used as a model for external access. The denoised image recognition result can be obtained by processing with the student model. Therefore, our defense strategy can maintain the speed of the model. The method prevents the model from fitting closely to the data and helps to generalize the training data. The parameter T controls the knowledge extracted by the action of Distillation. The larger T is, the larger the output probability value for recognition class is. The overall defense strategy is shown in Figure 5. The overall defense mechanism is divided into three phases. In the first stage, noise images are first obtained from clean pictures and corresponding attack images. The noise samples are used to train the GNR to achieve the reconstruction of the noise. The trained GAN is used to generate the corresponding adversarial image for the clean image. The more image pairs are obtained, and the trained dataset is expanded. In the second stage, the already expanded dataset is used to train the deep denoising network UDDN, and the denoised image can be obtained through accessing the trained UDDN. In the classification model, relatively clean images can be correctly identified with a high probability. To further improve the resilience of the model, in the third stage, the training strategy based on knowledge transfer is proposed to train the model. The teacher model is trained with the denoised images. After that, the student model is trained with the denoised images and the output of the teacher model. Finally, the student model is used for external access. The training method can enhance the robustness of the model and improve the resilience of the model. Our proposed defensive strategy applies to any deep neural network model, and the overall strategy does not significantly increase the computation of the network and maintains the speed of the network.

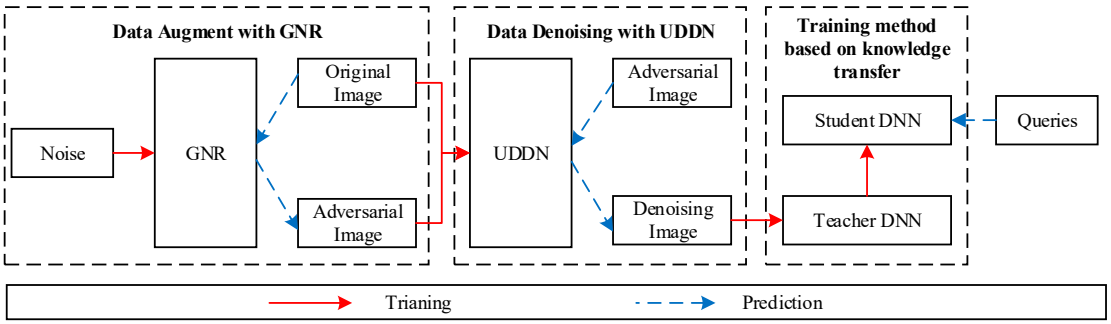

Figure 5. Overview of the defending approach. The approach consists of three stages. The first stage is expending adversarial images with GNR. The second stage is training the UDDN to denoising the attack images. The third stage is training the published model based on knowledge transfer.

5. Experimental Results and Analysis

In this section, we verify the effects of our proposed defense against adversarial attack and compare the results with other methods. We apply our defense mechanism to face recognition, which proves that our defense method can greatly reduce the success rate of attacks and improve the recognition accuracy of the model.

5.1. The Evaluation of UDDN

We apply our defensive strategy to existing face recognition models. We use the UDDN network to denoise the attack samples. The result of denoising is shown in Figure 6. x represents the original image of the face, x^* represents the image of the adversarial attack, which is generated by the FGSM

algorithm, and x^* represents the image denoised by our proposed UDNN. The next line is the difference between the adversarial image and the original image, and the difference between the denoised image and the original image. We can see that in the details, the image denoised by UDDN is clearer than the image before denoising, and more consistent with the original image.

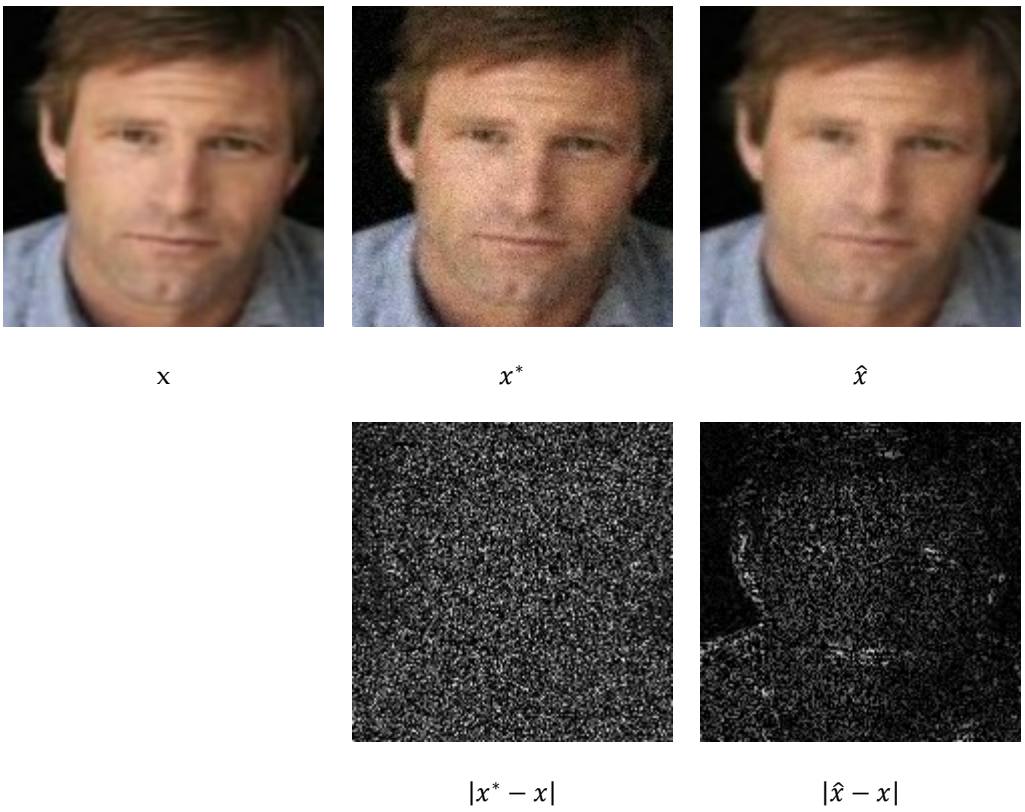

Figure 6. Denoising results with UDNN. x represents the clean image, x^* represents the attack image and $\hat{x}$ represents the denoised image. The images below them are the difference images between the images to the clean image.

To prove that our proposed UDDN can effectively resist against adversarial attacks, we compared it with the existing DAE algorithm and PGD algorithm. We experimented on the Labeled Faces in the Wild dataset (LFW) face dataset and used FaceNet as the face recognition model.

We first extract 10 k face images from the LFW (10 per class) and use a series of attack methods to distort the images and get the corresponding adversarial images. The attack methods include FGSM and IFGSM2, IFGSM4, and IFSGM8. The perturbation level of each sample is uniformly sampled from [0, 1] to obtain 40 k images, which will be used to train the GNR. Then, we continued to extract 20 k face data from the LFW (10 per class) and used the trained GNR to generate 80 k attack images. Finally, we collected a total of 120 k images in the training set. To prepare the validation set, we extract 5 k personal face images (five per class) from LFW. The same method was used to obtain 20 k attack images. Two different test sets were built, one for white-box attacks and the other for black-box attacks. We extracted 5 k personal face images (five per class) from LFW. The white-box attack test set uses two attack methods FGSM and FGSM4 based on FaceNet to obtain 10 k test images. The black-box attack test set uses two attack methods FGSM and FGSM4 based on pre-trained Inception V3 to obtain 10 K test images. When training UDDN, we set the learning rate initially to 0.001 and decay to 0.0001 when the training loss converges. The model is trained on 4 GPUs, the batch size is 32, and the number of trained epochs is between 30 and 40, depending on the convergence speed of the model.

We also performed white box attacks and black box attacks on the models. We calculated the L1 distance of the image generated by different denoising algorithms and the original image.

The result is shown in Table 1. The recognition accuracy of denoised image is also shown in the Table 2. NA represents a primitive face recognition model that does not defend by any protective strategy.

Table 1. The L1 distance between the input image and the denoised image.

Defense	Clean	WhiteTestSet	BlackTestSet
NA	0.0000	0.0373	0.0157
DAE	0.0153	0.0359	0.0161
PGD	0.0138	0.0178	0.0145
UDDN	**0.0125**	**0.0167**	**0.0134**

Table 2. The accuracy of model with different denoising methods on the test datasets.

Defense	Clean	WhiteTestSet	BlackTestSet
NA	84.5%	22.3%	69.0%
DAE	66.1%	28.8%	62.7%
PGD	81.9%	58.2%	72.2%
UDDN	**83.1%**	**60.2%**	**75.3%**

From Table 1, we can observe that our proposed deep denoising network UDDN can restore the closest image to the original image. It can be intuitively observed that the image denoised by our proposed UDDN is very close to the real image. To quantitatively analyze the performance of denoising method, we feed the denoised images into the classification model. The result is shown in Table 2. It can be seen from Table 2 that the recognition accuracy of the denoised images, which are generated by our proposed denoising network, achieved the highest accuracy. Under the white box attack, the accuracy of our proposed algorithm is nearly 2 times higher than DAE and 2% higher than PGD. Under the black box attack, our algorithm is 13% more accurate than DAE and 3% higher than PGD. The results of the model accuracy show that the probability of misclassification is greatly reduced after the images are denoised by the proposed UDDN. Compared with other denoising models, our method achieved higher accuracy, and it is more resistant to attack against adversarial attacks. Therefore, no matter what kind of adversarial attack, our proposed denoising algorithm is better than the existing algorithm and can recover the image closest to the original image.

5.2. The Evaluation of Defending Strategy

In our overall defense strategy, to improve the robustness of the model and improve the accuracy of the model, we also used a model training method based on knowledge transfer. We first used the deep denoising network to remove the noise of the attack image, and then used the relatively clean image to train the model. To verify that the overall strategy was effective, we compared our proposed defense strategy with other existing defense methods. The results are shown in Table 3 and Figure 7.

Table 3. The accuracy of model under different defending methods.

Defense	Clean	WhiteTestSet	BlackTestSet
NA	84.5%	22.3%	69.0%
Data Compression [13]	82.2%	53.4%	55.8%
PGD [27]	83.3%	51.4%	70.0%
DCN [28]	81.9%	58.2%	72.2%
Distillation [29]	83.6%	56.3%	70.8%
Our method	**83.9%**	**61.2%**	**76.5%**

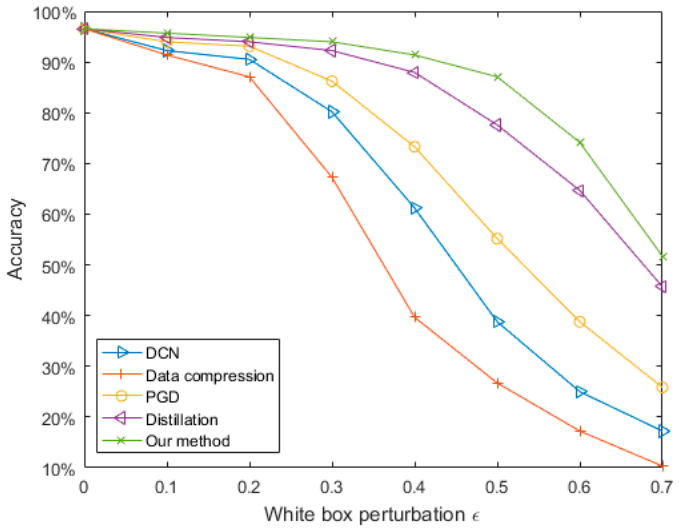

Figure 7. The accuracy of model under different levels of perturbation.

As we can see from Table 3, these defensive algorithms have achieved certain effects in resisting the adversarial samples. When the model does not have any defensive strategies, the accuracy rates obtained under white box attacks and black box attacks are 22% and 69%, respectively. Under the white box attack, Data Compression and DCN improved the accuracy of the model by 30%, PGD and Distillation increased the accuracy by about 35%, and our algorithm improved the model by about 40%. Under the black box attack, Data Compression has no obvious defense effect. DCN, PGD and Distillation improve the accuracy of the model by about 2%. Our method improves accuracy by about 6%, which is three times than that of other methods. The experimental results of the model accuracy show that the model achieved the highest accuracy under the protection of our proposed defense mechanism, which proves that we have the strongest defense ability against the adversarial attacks. Therefore, we can see that no matter what kind of attack, under our proposed algorithm, the accuracy of the model is the highest, and the best defense effect is achieved.

To prove the robustness of the strategy, the experiments were performed on different mainstream deep learning models and different datasets. Except for the LFW dataset, we performed experiments on the Youtube Face dataset (YTF) and Social Face Classification dataset (SFC) face datasets respectively. The results of the experiment are shown in Table 4.

Table 4. Accuracy of model using different datasets.

Dataset	Clean	WhiteTestSet	BlackTestSet
NA	83.9%	22.3%	69.0%
LFW	83.9%	61.2%	76.5%
YTF	81.2%	60.3%	74.4%
SFC	82.7%	60.9%	75.6%

From Table 4, we can see that compared with the model without the defense algorithm, our defense strategy has achieved good defensive effects on the three face datasets. In the white box attack, our defense strategy improved accuracy by approximately 40% on three datasets. In the black box attack, our model improved the accuracy by about 5%. Therefore, our proposed strategy is valid across multiple datasets. Our proposed defense strategy does not apply to a special face recognition model. To prove that our proposed algorithm has good generalization, we have selected several popular networks and used the face dataset LFW for training. The results of the experiment are shown in Table 5.

We can see from Table 5 that our proposed strategy has achieved good results on these models. In the white-box attack, compared with the unprotected model, our proposed defensive strategy can improve the accuracy of the model by about 40%. In the black box attack, we can increase the accuracy of the model by 10%. The results prove that our proposed algorithm applies to all DNN-based network models, and compared with other defense strategies, our model achieves the highest accuracy. Therefore, our proposed strategy is more resistant against adversarial attacks.

Table 5. Accuracy of model using our proposed strategy.

Model	Clean	WhiteTestSet/NA	BlackTestSet/NA
MobileNet	81.9%	60.5%/21.1%	75.7%/67.0%
FaceNet	83.9%	61.2%/22.3%	76.5%/69.0%
GoogleNet	81.7%	60.0%/21.0%	73.9%/66.3%
VGG 16	82.8%	61.0%/21.8%	76.3%/67.0%

6. Conclusions

In this work, we proposed a defense strategy that can resist against adversarial attacks. Our proposed algorithm utilizes GAN to augment the dataset. We utilized the differences at the top of the network as the loss function to guide the training of the image denoiser. At the same time, we proposed a training method based on knowledge transfer to improve the model's ability to resist subtle disturbances and improve the generalization ability of the model around the training data. Compared with other existing defense strategies, our proposed defensive strategy has achieved better resistance effect, and the protected model has achieved higher accuracy. Our algorithm is very robust against both white and black box attacks. The proposed algorithm is robust and suitable for most network structures based on DNN.

Author Contributions: Supervision, Y.L.; Writing–original draft, Y.W.

Acknowledgments: This work was supported by the Fundamental Research Funds for the Central Universities (2018ZD06).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Helmstaedter, M.; Briggman, K.L.; Turaga, S.C.; Jain, V.; Seung, H.S.; Denk, W. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature* **2013**, *500*, 168–174. [CrossRef] [PubMed]
2. Xiong, H.Y.; Alipanahi, B.; Lee, J.L.; Bretschneider, H.; Merico, D.; Yuen, R.K.; Morris, Q. The human splicing code reveals new insights into the genetic determinants of disease. *Science* **2015**, *347*, 1254806. [CrossRef] [PubMed]
3. Ciodaro, T.; Deva, D.; de Seixas, J.; Damazio, D. *Online Particle Detection with Neural Networks Based on Topological Calorimetry Information*; Journal of physics: conference series; IOP Publishing: Bristol, UK, 2012; Volume 368.
4. Ackerman, E. How Drive.ai Is Mastering Autonomous Driving with Deep Learning. Available online: https://spectrum.ieee.org/cars-that-think/transportation/self-driving/how-driveai-is-mastering-autonomous-driving-with-deep-learning (accessed on 10 March 2017).
5. Najafabadi, M.M.; Villanustre, F.; Khoshgoftaar, T.M.; Seliya, N.; Wald, R.; Muharemagic, E. Deep learning applications and challenges in big data analytics. *J. Big Data* **2015**, *2*, 1. [CrossRef]
6. Middlehurst, C. China Unveils World's First Facial Recognition ATM. 2015. Available online: http://www.telegraph.co.uk/news/worldnews/asia/china/11643314/China-unveils-worlds-first-facial-recognition-ATM.html (accessed on 1 June 2017).
7. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2014**, arXiv:1312.6199.
8. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572.

9. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the IEEE European Symposium on Security and Privacy, Saarbrucken, Germany, 21–24 March 2016.
10. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
11. Siniscalchi, S.M.; Salerno, V.M. Adaptation to new microphones using artificial neural networks with trainable activation functions. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 1959–1965. [CrossRef]
12. Salerno, V.M.; Rabbeni, G. An extreme learning machine approach to effective energy disaggregation. *Electronics* **2018**, *7*, 235. [CrossRef]
13. Dziugaite, G.K.; Ghahramani, Z.; Roy, D.M. A study of the effect of JPG compression on adversarial images. *arXiv* **2016**, arXiv:1608.00853.
14. Luo, Y.; Boix, X.; Roig, G.; Poggio, T.; Zhao, Q. Foveation-based mechanisms alleviate adversarial examples. *arXiv* **2015**, arXiv:1511.06292.
15. Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; Yuille, A. Adversarial Examples for Semantic Segmentation and Object Detection. *arXiv* **2017**, arXiv:1703.08603.
16. Ross, A.S.; Doshi-Velez, F. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients. *arXiv* **2017**, arXiv:1711.09404.
17. Zhang, A.; Wang, H.; Li, S.; Cui, Y.; Liu, Z.; Yang, G.; Hu, J. Transfer Learning with Deep Recurrent Neural Networks for Remaining Useful Life Estimation. *Appl. Sci.* **2018**, *8*, 2416. [CrossRef]
18. Nayebi, A.; Ganguli, S. Biologically inspired protection of deep networks from adversarial attacks. *arXiv* **2017**, arXiv:1703.09202.
19. Krotov, D.; Hopfield, J.J. Dense Associative Memory is Robust to Adversarial Inputs. *arXiv* **2017**, arXiv:1701.00939. [CrossRef] [PubMed]
20. Cisse, M.; Adi, Y.; Neverova, N.; Keshet, J. Houdini: Fooling deep structured prediction models. *arXiv* **2017**, arXiv:1707.05373.
21. Gao, J.; Wang, B.; Lin, Z.; Xu, W.; Qi, Y. DeepCloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples. *arXiv* **2017**, arXiv:1702.06763.
22. Akhtar, N.; Liu, J.; Mian, A. Defense against Universal Adversarial Perturbations. *arXiv* **2017**, arXiv:1711.05929.
23. Lu, J.; Issaranon, T.; Forsyth, D. SafetyNet: Detecting and Rejecting Adversarial Examples Robustly. *arXiv* **2017**, arXiv:1704.00103.
24. Metzen, J.H.; Genewein, T.; Fischer, V.; Bischoff, B. On Detecting Adversarial Perturbations. *arXiv* **2017**, arXiv:1702.04267.
25. Li, X.; Li, F. Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics. In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
26. Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; McDaniel, P. On the (Statistical) Detection of Adversarial Examples. *arXiv* **2017**, arXiv:1702.06280.
27. Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Zhu, J.; Hu, X. Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser. *arXiv* **2017**, arXiv:1712.02976.
28. Gu, S.; Rigazio, L. Towards Deep Neural Network Architectures Robust to Adversarial Examples. *arXiv* **2015**, arXiv:1412.5068.
29. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 582–597.