Defending against adversarial examples using defense kernel network

Yuying Hao[1]
haoyy17@mails.tsinghua.edu.cn
Tuanhui Li[2]
lth17@mails.tsinghua.edu.cn
Yong Jiang[2]
jiangy@sz.tsinghua.edu.cn
Xuanye Cheng[3]
xuanyech@gmail.com
Li Li[2]
lilihitcs@gmail.com

[1] Tsinghua-Berkely Shenzhen Institute, Tsinghua University, Shenzhen, China
[2] Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[3] SenseTime Research, SenseTime, Shenzhen, China

Abstract

Deep neural networks have been widely used in recent years. Thus, the security of deep neural networks is crucial for practical applications. Most of previous defense methods are not robust for diverse adversarial perturbations and rely on some specific structure or properties of the attacked model. In this work, we propose a novel defense kernel network to convert the adversarial examples to images with evident classification features. Our method is robust to variety adversarial perturbations and can be independently applied to different attacked model. Experiments on two benchmarks demonstrate that our method has competitive defense ability compared with existing state-of-the-art defense methods.

1 Introduction

Deep neural networks have achieved remarkable performance in artificial intelligence tasks, such as image classification [20], speech recognition [4] and object detection [6, 22]. Recent works have been proved that deep neural networks are vulnerable to the well-designed perturbations [3]. These deep neural networks become the attacked model for adversarial examples. Thus, security becomes a critical aspect for deployment of artificial intelligence models. To improve the security of the neural networks, several methods have been proposed to obtain a robust model, such as obfuscated gradients [2, 17, 24], adversarial training [12, 16, 19, 23] and feature squeezing [8, 9, 25]. The obfuscated gradients and adversarial training improve the robustness by modifying the attacked model, but spend expensive computation in generating adversarial examples. As a comparison, the methods based on feature squeezing smooth which simplify the adversarial examples, but reduce the classification accuracy for exchange. These works make contributions to the research for defending against adversarial examples, but their performance shows that their methods are not robust to variety adversarial attacks. Within our knowledge, an effective way to defend against variety

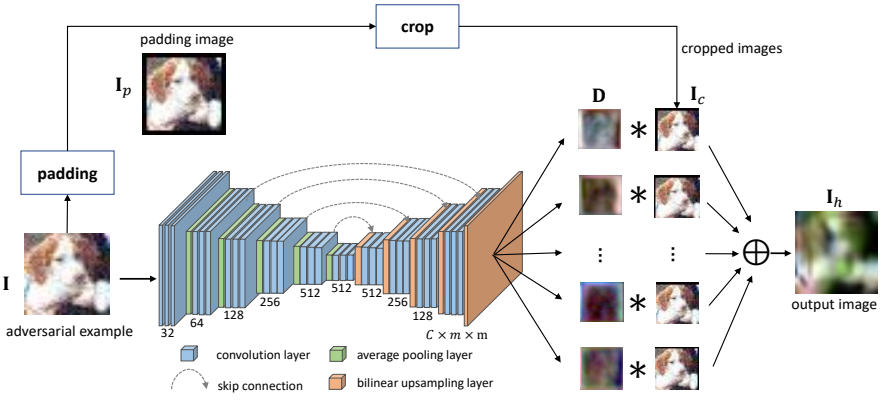

Figure 1: The framework of this proposed defense kernel structure.

adversarial examples is still not be proposed. This situation seriously affects the development of deep neural network applications.

In this work, we aim to construct a robust model to defend against variety adversarial examples, and this model does not rely on a specific attacked model. Our strategy is to learn robust defense kernel network to remove the adversarial perturbations and convert the adversarial examples to the images with evident classification features, such that the impact of the adversarial examples on attacked model can be reduced. Inspired by [15] applying a 'U-Net' structure for denoising, we adopt a similar structure to construct a robust defense kernel network. This model can convert the input adversarial examples to new images with highlighted classification features. The detailed architecture of our defense method is shown on Fig. 1. Our contribution can be summarized as follow:

(1) The proposed method introduces a general defense structure and is robust to variety attack methods. Our method does not rely on some specific architecture or the properties of attacked model and does not modify the parameters of attacked model.

(2) Our method utilizes defense kernel network to convert the adversarial examples to images with evident classification features. The learned kernels contains specific per-pixel factors for enhancing the important features for classification. Compared to the common convolution methods of sharing weight parameters, the defense kernel network can deal with complex and variable adversarial attacks.

(3) Extensive experiments on two benchmarks demonstrate that our method has competitive defense ability compared to state-of-the-art defense methods. Besides, our method can be applied to unknown attacked models without being fine-tuned and has stable and robust performance on defending against adversarial examples.

2 Relate work

Recent years, a lot of methods have been proposed to defend against the adversarial examples, which can be roughly divided into 4 categories: obfuscated gradients, adversarial training, projection, preprocessing.

Obfuscated gradients. These methods attempt to apply the gradient masks for preventing the attack methods from correctly calculating the gradient. Tramèr *et al*. [24] destroyed the original gradient information and pointed to the wrong gradient direction to the adversarial attacks. Prakash *et al*. [17] utilized the randomized defenses strategy to achieve the defense goal. However, Athalye *et al*. [2] proposed that their attack method can successfully

circumvent obfuscated gradients and attack these models successfully.

Adversarial training. These methods aim to train a robust classifier by adding adversarial examples in training set [16, 23]. Shaham *et al*. [19] utilized the adversarial examples and the benign images as the inputs to train the model. Goodfellow *et al*. [7] used the adversarial examples to generate additional regularization losses. Madry *et al*. [12] demonstrated that adversarial training based on 'PGD-attack' could increase the robustness of attacked model. Adversarial training based methods perform excellently for white-box adversarial attacks. However, they are not robust to black-box attacks and cost expensive computation for generating adversarial examples.

Projection. The projection methods aim to project the inputs onto a low dimension manifold. Meng *et al*. [14] introduced the 'magnet' to defend adversarial examples. They applied the auto-encoder to training the detector and reformer, and moved the adversarial examples to the closer manifold of benign images. Samangouei *et al*. [18] adopted the generative models to reconstruct the inputs, and determined adversarial examples based on the distance metric between the original inputs and corresponding reconstructed images. However, these strategies reduce the classification accuracy of benign images and are required to lots of adversarial examples.

Preprocessing. These strategies reduce the effect of adversarial perturbations by decreasing the sensitive effect for adversarial perturbations. Weilin *et al*. [25] introduced the feature squeezing strategy by reducing the color bit depth and smoothing the whole images, such that many subtle different input values are projected into the same value. Zoubin *et al*. [5] improved the robustness of model by compressing the inputs based on JPEG criterion. Guo *et al*. [9] defended against adversarial examples by minimizing total variance. However, the methods based on preprocesing only increase the defense ability slightly and can not guarantee the classification accuracy of the benign images.

The previous works are not robust to different attacked models and attack methods. Different from these methods, our method converts the inputs to the images with evident classification features and we do not change the attacked model or reduce the classification accuracy. Our method is robust to different attacked models and attack methods and has powerful defense ability.

3 The proposed method

3.1 Background

In this paper, we denote $\mathbf{X}$ as the benign image. The untargeted attack goal is to add an imperceptible small perturbations δ to $\mathbf{X}$ and generate an adversarial example $\mathbf{X}'$. $\mathbf{X}'$ can lead to misclassification for the classifier, *e.g.* $\arg\max f(\mathbf{X}) \neq \arg\max f(\mathbf{X}')$, where $f(\cdot)$ denotes the output of classifier. In the training process, we mainly consider two types of adversarial examples: random perturbed image $\mathbf{X}^{rand}$, FGSM perturbed image $\mathbf{X}^{fgsm}$.

Random perturbation. This method is the simplest attack method for classifier. We utilize $\mathbf{X}^{rand}$ in the training process to increase the robustness of defense kernel network for random noise. The definition of $\mathbf{X}^{rand}$ can be described as:

$$\mathbf{X}^{rand} = \mathbf{X} + \mathcal{U}(-\varepsilon, \varepsilon), \tag{1}$$

where $\mathcal{U}(-\varepsilon, \varepsilon)$ is denoted as the uniform distribution between $[-\varepsilon, \varepsilon]$.

Fast gradient sign method. This method is an one-step attack proposed by Goodfellow *et al*. [7]. It aims to add perturbations based on the direction of gradient ascend. $\mathbf{X}^{fgsm}$

can be described as:

$$\mathbf{X}^{fgsm} = \mathbf{X} + \varepsilon \text{Sign}(\nabla_{\mathbf{X}} \mathcal{L}(\mathbf{X}, \mathbf{y})), \tag{2}$$

where $\mathbf{y}$ is the ground-truth of $\mathbf{X}$ and $\mathcal{L}(\mathbf{X}, \mathbf{y})$ is the loss function for the attacked model. We select this attack method since it cost little computation while maintaining the gradient information.

3.2 Defense kernel network

Adversarial perturbation is a kind of well-designed noise, which can destroy the classification features of the benign image. However, due to the diversity and imperceptibility of the perturbations, it is difficult to detect the adversarial examples by the normal trained classifiers. Inspired by [13], we propose a novel defense kernel network to restore the destroyed features of the adversarial examples,as shown on Fig. 1.

We define $\mathbf{I} \in \mathbb{R}^{C \times W \times H}$ as the input, where C, W and H are denoted as the input channels, the width and height, respectively. During training process, the input $\mathbf{I}$ can be the benign image $\mathbf{X}$, random perturbed image $\mathbf{X}^{rand}$ and FGSM perturbed image $\mathbf{X}^{fgsm}$. The output of the defense kernel network contains $K = C \times m \times m$ channels , which can be split into m^2 per-pixel kernels, we denote the i-th per-pixel kernel as $\mathbf{D}^i \in \mathbb{R}^{C \times W \times H}$. Per-pixel kernels can efficiently defend against dense adversarial perturbations, while for sparse adversarial perturbations with pixel value bounce, we need to take neighbor pixel-wise information to smooth these outliers. To increase the robustness for sparse adversarial attacks and flexibly generate images with neighbor pixel information, we make subtle positional changes to the input image. To obtain $\mathbf{I}_p \in \mathbb{R}^{C \times (W + 2\lfloor m/2 \rfloor) \times (H + 2\lfloor m/2 \rfloor)}$, we expand the size of the input image $\mathbf{I}$ with padding zeros. To conduct per-pixel convolution with the per-pixel kernels, we crop $\mathbf{I}_p$ into m^2 images and denote the i-th image as: $\mathbf{I}_c^i \in \mathbb{R}^{C \times W \times H}$ $(i \in [1, 2, 3, ..., m^2])$. We apply convolution operation to the per-pixel kernels and the cropped images to obtain the highlighted image $\mathbf{I}_h$, which can be expressed as:

$$\mathbf{I}_h = \frac{1}{m^2} \sum_{i=1}^{m^2} \mathbf{D}^i * \mathbf{I}_c^i, \tag{3}$$

where $*$ denotes the per-pixel convolution operation.

3.3 Training the defense kernel network

The benign images contain critical information for correct classification. To get the pixel-wise information, ℓ_p-norm $(p = 1, 2, \infty)$ is always adopted in previous works. In this work, we use weighted ℓ_1-norm, which can be described as:

$$\ell_1 = ||\mathbf{X} - \mathbf{I}_h||_1, \tag{4}$$

where $\mathbf{X}$ and $\mathbf{I}_h$ are denoted as the benign image and generated image, respectively. Besides, to reduce the differences of high-level representation between $\mathbf{X}$ and $\mathbf{I}_h$, we utilize the perceptual loss to guide the training process. Inspired by [11], we define the perceptual loss ℓ_p as follows:

$$\ell_p = \frac{1}{n} \sum_{l=1}^{n} \frac{1}{N^l W^l H^l} ||\psi(\mathbf{X}^l) - \psi(\mathbf{I}_h^l)||^2, \tag{5}$$

Algorithm 1 Training process of the defense kernel network for one iteration

1: **Input:** randomly selected benign image $\mathbf{X}$
2: **Output:** generated highlighted image $\mathbf{I}_h$
3: generate $\mathbf{X}^{rand}$ based on (1);
4: generate $\mathbf{X}^{fgsm}$ based on (2);
5: randomly select an input $\mathbf{I} \in \{\mathbf{X}^{rand}, \mathbf{X}^{fgsm}, \mathbf{X}\}$;
6: calculate $\mathbf{I}_h$ based on (3);
7: take $\mathbf{I}_h$ as the input of attacked model and calculate $\mathcal{L}$ based on (7);
8: update parameters of the defense kernel network.

where $\psi(\mathbf{X}^l)$ and $\psi(\mathbf{I}_h^l)$ are the output feature maps of $\mathbf{X}$ and $\mathbf{I}_h$ of layer l, respectively. N^l,W^l and H^l are the output channels, the kernel width and height of layer l, respectively. n is the number of selected layers for extracting high-level representations. Moreover, to keep up the classification accuracy of attacked model when adding defense kernel network, we adopt the cross-entropy loss in the training process, which can be described as follows:

$$\ell_c = -\sum \mathbf{y} \log \hat{\mathbf{y}}, \tag{6}$$

where $\mathbf{y}$ is the one-hot label of $\mathbf{X}$ and $\hat{\mathbf{y}}$ is the model output. The total loss $\mathcal{L}$ for training the defense kernel network in our method can be expressed as follows:

$$\mathcal{L} = \lambda_1 \ell_1 + \lambda_2 \ell_p + \lambda_3 \ell_c, \tag{7}$$

where λ_1, λ_2 and λ_3 are the hyperparameters. The training algorithm of our method is shown on Algorithm 1.

4 Experiment

4.1 Experiment setting

Datasets. Our experiments are based on two benchmarks, MNIST and CIFAR-10. MNIST contains 60000 images in the training set, 10000 images in testing set, and can be divided into 10 categories. CIFAR-10 includes 50000 images in the training set, 10000 images in the testing set, and can be divided into 10 categories. In our experiments, we randomly generate 1000 untargeted adversarial examples from two testing set for each attack method.

Models and comparison methods. For white-box attack on MNIST, we apply model A, including four convolutional layers, two maxpooling layers and three full connected layers, as the attacked model. For black-box attack on MNIST, we generate adversarial examples on model A and transfer the adversarial examples for attacking model B. Model B has four convolutional layers, four maxpooling layers and one full connected layer. For white-box attack on CIFAR-10, we select VGG-19 [21] as attacked model. For black-box attack on CIFAR-10, we generate adversarial examples on VGG-19 and attack ResNet-18 [10]. In this work, we take the Cleverhans [1] to complete the attack methods including FGSM [7], PGD [13] and CW-ℓ_2 [3] and defense methods including Featuresqueezing [25], Jpegcompression [5], Totalvariancemin [8] and Spatialsmoothing [25]. Every defense methods take the same adversarial examples as the inputs and are tested on the same attacked model.

Implementation details. In the experiment, we normalize the image into [0,1]. The learning rate η, λ_1, λ_2, λ_3 are set to $0.0001, 1, 1, 1$, respectively. The maximum training epochs are set

Table 1: Classification accuracy of different attacked models on MNIST. The column of 'White-box' expresses the accuracy on white-box attack. The column of 'Black-box' expresses the accuracy on black-box attack. Higher accuracy indicates better defense ability.

Attack method	Defense method	Setting	White-box	Black-box
FGSM [9]	Featuresqueezing [25]	depth=1	58.27	76.74
		depth=2	4.59	69.13
		depth=5	2.00	70.91
	Spatialsmoothing [25]	window_size=2	40.09	66.79
		window_size=3	16.00	71.47
	Jpegcompression [8]	quality=70%	15.17	71.32
		quality=80%	13.17	71.51
		quality=90%	10.31	71.50
	Totalvariencemin [8]	-	41.67	64.82
	Ours	-	**99.33**	**88.30**
PGD [13]	Featuresqueezing [25]	depth=1	**93.34**	81.31
		depth=2	5.23	71.62
		depth=5	0.35	77.11
	Spatialsmoothing [25]	window_size=2	15.21	77.42
		window_size=3	11.83	71.42
	Jpegcompression [8]	quality=70%	7.62	77.80
		quality=80%	6.74	75.00
		quality=90%	3.38	76.43
	Totalvariencemin [8]	-	33.32	71.45
	Ours	-	92.95	**82.37**
CW-ℓ_2 [3]	Featuresqueezing [25]	depth=1	75.00	87.77
		depth=2	77.07	90.09
		depth=5	1.78	89.86
	Spatialsmoothing [25]	window_size=2	88.48	89.09
		window_size=3	76.11	88.62
	Jpegcompression [8]	quality=70%	91.09	90.19
		quality=80%	90.62	89.87
		quality=90%	88.54	90.19
	Totalvariencemin [8]	-	82.70	87.94
	Ours	-	**95.88**	**90.62**

to 100 and 600 on MNIST and CIFAR-10, respectively. To fit the structure of defense kernel network, we preprocess MNIST from 28 to 32 by filling zeros and take them as the inputs of defense kernel network. For perceptual loss, we utilize VGG-19 as the perceptual module and extract high-level representations on 12-th and 16-th pooling layers. For defense kernel parameter, we set $m = 5$ and generate 25 kernels in our experiments. We select classification accuracy to evaluate defense ability, higher accuracy indicates better defense ability.

4.2 Experiment results

4.2.1 Results on MNIST

The results of MNIST are shown on table 1. For white-box attack, we generate adversarial examples on model A with accuracy 98.30%. When we add defense kernel network in front of model A, the accuracy on benign images is 98.55%. The defense kernel network does not decrease the classification accuracy of attacked model. Compared with other methods, our method achieves competitive performance on different attacks. For example, on FGSM attack, the accuracy of our method achieves 99.33%, compared with the best results of 58.27% on Featuresqueezing [25], 40.09% on Spatialsmoothing [25] and 15.17% on Jpegcompression [8]. Besides, our method defends against different attack methods with stable perfor-

Table 2: Classification accuracy of different attacked models on CIFAR-10.

Attack method	Defense method	Setting	White-box	Black-box
FGSM [7]	Featuresqueezing [25]	depth=1	8.17	14.96
		depth=5	0.18	9.64
		depth=10	0	9.64
	Spatialsmoothing [25]	window_size=2	8.33	11.51
		window_size=3	9.93	12.07
	Jpegcompression [5]	quality=80%	1.52	9.73
		quality=90%	0	9.43
		quality=95%	1.37	9.64
	Totalvariancemin [8]	-	15.01	10.14
	Ours	-	**42.84**	**51.30**
PGD [13]	Featuresqueezing [25]	depth=1	3.63	15.83
		depth=5	0.14	0.14
		depth=10	0	0.87
	Spatialsmoothing [25]	window_size=2	0	7.11
		window_size=3	0	5.53
	Jpegcompression [5]	quality=80%	0.81	4.32
		quality=90%	1.69	2.50
		quality=95%	1.24	1.47
	Totalvariancemin [8]	-	13.66	16.30
	Ours	-	**82.37**	**49.18**
CW-ℓ_2 [3]	Featuresqueezing [25]	depth=1	46.31	27.18
		depth=5	62.12	**87.12**
		depth=10	0.95	87.10
	Spatialsmoothing [25]	window_size=2	68.20	82.82
		window_size=3	62.97	80.01
	Jpegcompression [5]	quality=80%	**79.36**	79.03
		quality=90%	74.53	83.56
		quality=95%	67.15	86.80
	Totalvariancemin [8]	-	47.30	51.00
	Ours	-	71.72	86.60

mance, for the accuracy of 3 different attack methods achieves above 90%. In comparison, Featuresqueezing performs well on PGD [13] attack when depth=1, but decrease sharply when changing the depth value. Jpegcompression performs excellently on CW-ℓ_2 [3] attack, but not stable on PGD and FGSM [7] attacks. Our method has the impressive performance on white-box attack on MNIST.

For black-box attack, we generate adversarial examples on model A and attack model B. The accuracy of model B on benign images is 99.42%. On black-box attack, our method outperforms all other defense methods in comparison. For instance, best result of Featuresqueezing, Spatialsmoothing, Jpegcompression and Totalvariancemin on FGSM is 76.74%, 71.47%, 71.50%, 64.82%, respectively. As a comparison, our method achieves 88.30% on FGSM attack, and greatly improves the performance. Our method neither relies on the structure nor the specific properties of the attacked model. It aims to convert the adversarial examples to images with evident classification features, and learn the flexible per-pixel kernels by the distribution of the inputs. Thus, our method performs best for black-box attack.

4.2.2 Results on CIFAR-10

Table 2 shows the classification accuracy on CIFAR-10. The accuracy of attacked model (VGG-19) on benign images is 89.32%. When we add defense kernel network in front of the attacked model, the accuracy of attacked model is 89.79%. Adding defense kernel network does not decrease the classification accuracy of attacked model. For white-box attack, our

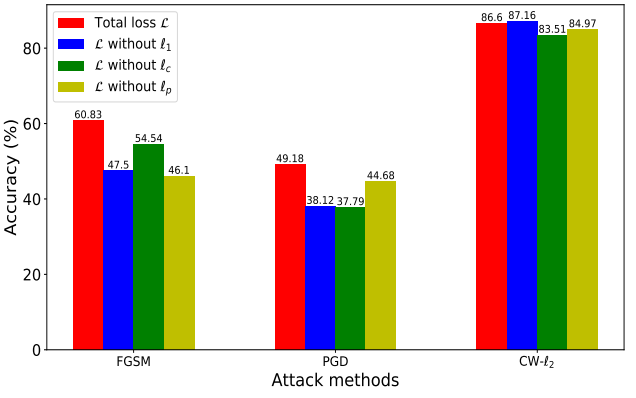

Figure 2: The effects of different loss for training the defense kernel network on CIFAR-10. we generate adversarial examples on the 'VGG-19' and attack 'ResNet-18'.

method achieves the best results on FGSM and PGD attacks. For example, the accuracy of Totalvariencemin [8] on PGD attack is 13.66%, our method is 82.37%, the accuracy has been greatly improved. For CW-ℓ_2, our method do not achieves the best performance. We analyse and conclude that the adversarial examples generated on CW-ℓ_2 are similar to the benign images. The perturbations of CW-ℓ_2 are dense and subtle, while our method performs excellently on perceptible noise. Our defense kernel network still performs competitively on dense and subtle adversarial perturbations, compared with Featuresqueezing [25], Spatialsmoothing [25] and Totalvariencemin [8].

For black-box attack, we generate adversarial examples on VGG-19 and attack ResNet-18. The accuracy of ResNet-18 on benign images is 91.73%. Compared to other defense methods, our method achieves outstanding results. For example, when adversarial examples generated on FGSM, the best result of Featuresqueezing is 14.96%, but ours achieves 51.30%. Besides, the adversarial examples of PGD attack generated on VGG-19 also have the strong attack abilities on ResNet-18. For instance, accuracy of Jpegcompression [5] under PGD black-box attack is below 5%, while our method achieves 49.18%. Our method has a significant improvement on defending against adversarial examples.

Above experiments demonstrate that our defense method is more robust than other compared defense methods. Adopting the defense kernel network can improve the classification accuracy of the attacked model. Our defense method does not modify the parameters of the attacked model and shows the excellent generalization ability on different models. The novel defense kernel network shows impressive abilities to defend against adversarial examples.

4.3 Analysis of the training loss

To evaluate the effects of loss terms on robustness individually, we compare the accuracy of different loss term cases. As shown in Fig. 2 with 4 cases: total loss $\mathcal{L}$, $\mathcal{L}$ without ℓ_1, $\mathcal{L}$ without ℓ_c and $\mathcal{L}$ without ℓ_p. In this experiment, we generate adversarial examples on CIFAR-10 via 3 attack methods: FGSM, PGD and CW-ℓ_2, respectively. Then we use different loss terms to train our defense kernel network. Different loss terms can affect the performance of the defense kernel network directly. For example, on FGSM attack, the accuracy of total loss $\mathcal{L}$ is 60.83%. When we abandon the ℓ_1, ℓ_c, ℓ_p in the training process, the accuracy drops to 47.5%, 54.54% and 46.1%, respectively. It illustrates that for one-step attack method like FGSM, ℓ_p with high-level representations takes important defense infor-

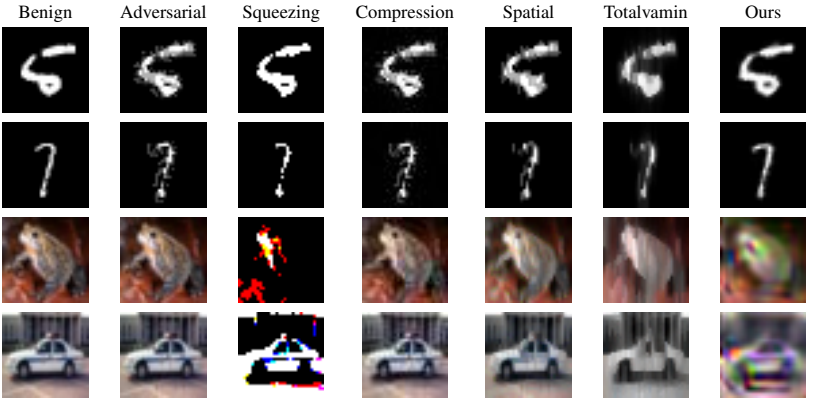

Figure 3: The visualization of different defense methods. The first and second rows are the images on MNIST. The third and forth rows are the images on CIFAR-10. The first column shows benign images, the second column shows adversarial examples, the third to sixth show the adversarial examples after being converted by Featuresqueezing [25], Jpegcompression [5], Spatialsmoothing [25] and Totalvariencemin [8], respectively. The last column shows the converted images of our method.

mation. For PGD attack, the accuracy of four cases is 49.18%, 38.12%, 37.79%, 44.68%, respectively. It shows that when we abandon ℓ_c containing classification information, the accuracy drops from 49.18% to 37.79%. This illustrates that for unknown attacked method not involved in the training process, defense kernel network relies on enhancing the classification features to achieve the defense goals. Although the added features have no semantic information for human, they can enhance the classification features for attacked model. For CW-ℓ_2, the distinction of accuracy caused by different loss terms is not obvious, and they all achieve the ideal results. CW-ℓ_2 attack is weak for our attack regardless of the forms of the loss function. Considering the robustness of defense kernel network, in this work, we choose $\mathcal{L}$ as loss function, and it can be adapted to different adversarial attacks.

4.4 Visualization of the defense results

In this section, we visualize the effects of different defense methods on adversarial examples in Fig. 3. As for Jpegcompression [5], Spatialsmoothing [25] and Totalvariencemin [8], they all try to smooth the adversarial examples in visual to reduce the effect of adversarial perturbations. Featuresqueezing [25] simplifies the bit-depth to defend against adversarial examples. Different from these methods, our method tries to convert the adversarial examples with enhanced features. Although these features are not semantic for human, they contain key information of classification for neural network.

5 Conclusion

In this work, we introduce a novel approach to defend against adversarial examples. We adopt the novel loss function to obtain the learn-based defense kernel network, and then adding defense kernel network in front of the attacked model to convert the adversarial examples. Our method neither changes the attacked model nor relies on the specific properties of the attacked model. Besides, our method is robust to different attack methods and different attacked models. The experiments demonstrate that our approach can achieve the impressive defense results, compared with state-of-the-art methods.

Acknowledgements. This work is supported by Guangdong Province Key Area RD Program under grant No. 2018B010113001, the RD Program of Shenzhen under grant No. JCYJ20170307153157440 and the Shenzhen Key Lab of Software Defined Networking under grant No. ZDSYS20140509172959989, and the National Natural Science Foundation of China under Grant No. 61802220.

References

[1] Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. pages 274–283.

[3] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[4] Dong Yu et al Geoffrey Hinton, Li Deng. Deep neural networks for acoustic modeling in speech recognition. *The Shared Views of Four Research Groups. IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[5] Zoubin Ghahramani Gintare Karolina Dziugaite and Daniel Roy. A study of the effect of jpeg compression on adversarial images. arXiv:1608.00853.

[6] Darrell T et al Girshick R, Donahue J. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society.

[7] Shlens J. Goodfellow, I. J. and C Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[8] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. 2018.

[9] Cisse M et al. Guo C, Rana M. Countering adversarial images using input transformations. 2018.

[10] Zhang X. Ren S. He, K. and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[11] Alahi A. Johnson, J. and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711, 2016.

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. 2018.

[14] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[15] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018.

[16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[17] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8571–8580, 2018.

[18] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. 2018.

[19] Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307:195–204, 2018.

[20] Rodner E Simon M. Neural activation constellations: Unsupervised part model. *Discovery with Convolutional Networks*, pages 1143–1151, 2015.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[22] Zisserman A Simonyan K. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. 2014.

[24] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. 2018a.

[25] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *In Network and Distributed Systems Security Symposium (NDSS)*, 2018.