Resisting Adversarial Examples via Wavelet Extension and Denoising

Qinkai Zheng[1], Han Qiu[2], Tianwei Zhang[3], Gerard Memmi[2], Meikang Qiu[4], and Jialiang Lu[1]*

[1] Shanghai Jiao Tong University, 200240, Shanghai, China.
{paristech_hill,jialiang.lu}@sjtu.edu.cn
[2] Telecom Paris, 91120, Palaiseau, France.
{han.qiu,gerard.memmi}@telecom-paris.fr
[3] Nanyang Technological University, 639798, Singapore.
tianwei.zhang@ntu.edu.sg
[4] Texas A&M University Commerce, 75428, TX, USA.
meikang.qiu@tamuc.edu

Abstract. It is well known that Deep Neural Networks are vulnerable to adversarial examples. An adversary can inject carefully-crafted perturbations on clean input to manipulate the model output. Past years have witnessed the arms race between adversarial attacks and defenses. In this paper, we propose a novel method, WED, to better resist adversarial examples. Specifically, WED adopts a wavelet transform to extend the input dimension with the image structures and basic elements. This can add significant difficulty for the adversary to calculate effective perturbations. WED further utilizes wavelet denoising to reduce the impact of adversarial perturbations on the model performance. Evaluations show that WED can resist 7 common adversarial attacks under both black-box and white-box scenarios. It outperforms two state-of-the-art wavelet-based approaches for both model accuracy and defense effectiveness.

Keywords: Adversarial Examples · Deep learning · Model Robustness · Wavelet Transform · Image Denoising

1 Introduction

With the revolutionary development of Deep Learning technology, Deep Neural Networks (DNN) has been widely adopted in many computer vision tasks and applications, e.g., image classification, objective detection, image reconstruction, etc. However, DNN models are well known to be vulnerable against Adversarial Examples (AEs) [5]. An adversary can add carefully-crafted imperceptible perturbations to the original images, which can totally alter the model results. Various methods have been proposed to generate AEs efficiently and effectively such as FGSM [5] and CW [2]. These adversarial attacks have been applied to physical scenarios [7] and real-world computer vision applications.

* Jialiang Lu is the corresponding author.

It is extremely challenging to defend AEs. First, an adversary has different approaches to generate AEs. It is hard to train a attack-agnostic DNN model without pre-knowledge of the specific attack techniques. Second, correcting the model's results on AEs can usually alter its behaviors on normal inputs, which can lead to non-negligible performance loss. Third, due to the strong transferability of the adversarial examples, the adversary can compromise the models without having access to the model structures or parameters. Such black-box techniques significantly increase the difficulty of model protection via hiding or restricting model information.

There are a variety of works attempting to address those challenges, however, they all suffer certain drawbacks or practical issues. Up to now, there are no satisfactory solutions to defeat all adversarial attacks under both black-box and white-box scenarios. Specifically, (1) adversarial training [5, 8] is a class of methods that consider generating AEs during training. Such methods require the knowledge of adversarial attacks to create AEs, which depends on the type of the attacks. (2) Defensive distillation [11] enhances the model robustness during training through the model distillation mechanisms. This approach has been shown ineffective when the adversary slightly modifies the AE generation algorithm. (3) Some approaches proposed to pre-process the input images [12] to remove the adversarial perturbations at the cost of accuracy degradation on clean images. The adversary can still easily defeat such methods by including the transformations into optimization procedure [1].

In this paper, we propose a novel approach, `WED`, to defend AEs via Wavelet Extension and Denoising. The key insight of our approach is that *by extending the input image into a higher dimensional tensor with non-differentiable wavelet transforms, it is extremely difficult for the adversary to generate perturbations that alter the model's output.* Specifically, `WED` consists of two innovations. The first innovation is Wavelet Extension. `WED` adopts wavelet transform to extend the input image into two scales: the original one and low-frequency one. These two scales are then combined as one tensor as the input, which makes it difficult to generate effective adversarial perturbations. The second innovation is to adopt wavelet denoising only at the inference step. The original image, as one scale, keeps the original visual content and elements to assist the model classification. `WED` also utilizes wavelet denoising during the inference phase to reduce the impact of the adversarial perturbations.

Our comprehensive experiments show that `WED` outperforms state-of-the-art defenses (e.g., Pixel Deflection (`PD`) [12] or Wavelet Approximation (`WA`) [13]) from different aspects. For robustness, `WED` shows higher robustness against almost all known adversarial attacks under black-box scenarios (e.g. 97% accuracy for AEs generated by CW) compared with `PD`. For performance, `WED` has higher classification accuracy on clean image samples (97% accuracy on the test compared with 92%from `PD`, more details see Section 4.2).

The roadmap of this paper is as follows. In Section 2, we briefly introduce the background of adversarial attacks and defenses. In Section 3, we presents the design of our proposed `WED`. In Section 4, we comprehensively evaluate the

robustness and performance of WED and compare it with state-of-the-art solutions under different scenarios. We conclude in Section 5.

2 Research Background

2.1 Adversarial Attacks

The goal of adversarial attacks is to make the DNN model give wrong predictions by adding imperceptible perturbations to the original input. Formally, for a clean image x, we denote its corresponding AE as $\widetilde{x} = x + \delta$ where δ is the adversarial perturbation. Let F be the classifier mapping function of the DNN model, the process of generating AEs can be formulated as the following problem:

$$\begin{aligned} &min \|\delta\| \\ &s.t.\ F(\widetilde{x}) = l', F(x) = l, l' \neq l \end{aligned} \tag{1}$$

where l is the correct class of clean image x, l' is the target class misled by $\widetilde{x}$. The adversary aims to find the optimal perturbation to mislead the classifier.

Various techniques have been proposed to generate AEs, and there are two main categories. (1) *Naive Gradient-based* approaches: the adversary generates AEs by calculating the model gradients with pre-set constraints of how much modification can be made on input samples. For instance, Fast Gradient Sign Method (FGSM) [5] calculates the perturbations based on the sign of the gradient of the loss function with respect to the input sample. This kind of methods, also including I-FGSM [7] and MI-FGSM [4], aim at iteratively calculating the perturbations with a small step or with momentum. (2) *Optimized Gradient-based* approaches: the adversary adopts optimization algorithms [2] to find optimal perturbations by considering the gradients of the predictions with respect to input images. This kind of solutions are very powerful under white-box scenario since the attackers can adjust the AE generation according to defense methods. Recently, there are many optimized gradient-based approaches to generate AEs including JSMA [10], DeepFool [9], LBFGS [15], CW [2].

2.2 Defense Strategies

There exists an arms race between adversarial attacks and defenses. Different solutions were proposed to defend different attacks, which are mainly following two directions. The first direction is to apply preprocessing like transformations on input images to reduce the impact of carefully-crafted adversarial perturbations. As shown in Eq. 2, the defense applies a transformation function τ on the input image. It tries to maximize the probability of classifying the adversarial examples $\widetilde{x}$ to the correct class of the original image x. The transformation function can be non-differentiable and non-invertible, which makes it difficult for adversaries to get the gradients through back-propagation. Signal processing techniques are commonly adopted in this case, e.g., wavelet transformation [13] and denoising [12]. However, there exists a trade-off between effectiveness and

performance: weak transformation fails to remove the impact of small adversarial perturbations; while intensive transformation can result in an obvious accuracy loss on the clean images. Also, under the white-box scenario, such methods are vulnerable to the optimized gradient-based adversarial attacks. If the adversary includes the preprocessing in the optimization procedure, the AEs generated can still be effective to mislead the DNN model.

$$\max_{\tau} P(F(\tau(\widetilde{x})) = F(\tau(x))) \tag{2}$$

The second direction is to modify the target model to increase its robustness. As shown in Eq. 3, F' denotes the classifier mapping function of the modified model. Typical examples include adversarial training [5, 8] and network distillation [11]. These approaches are effective under some conditions, however, there will be a high cost due to model retraining.

$$\max_{F'} P(F'(\widetilde{x}) = F'(x)) \tag{3}$$

3 Methodology

3.1 Overview

The key idea of our approach `WED` is to extend the input image x into two parts ($\tau_1(x)$, $\tau_2(x)$) with two transformation functions (τ_1 and τ_2), as shown in Eq. 4.

$$\max_{F',\tau_1,\tau_2} P(F'(\tau_1(\widetilde{x}), \tau_2(\widetilde{x})) = F'(\tau_1(x), \tau_2(x))) \tag{4}$$

Eq. 4 combines two defenses strategies in Eq. 2 and Eq. 3 at the same time by introducing transformations (τ_1 and τ_2) and modifying the mapping function of model F'. We then try to maximize the probability of classifying the AE $\widetilde{x}$ to the correct class as the original image x.

These two transformations must be carefully designed to satisfy two requirements: (1) it should prevent the adversary from affecting the processed images by perturbing the input; (2) it should maintain high prediction accuracy on clean images. To achieve those goals, we adopt the wavelet transformations (extension and denoising) for τ_1 and τ_2 respectively. Fig. 1 shows the overview of our approach. During the training phase (Fig. 1(a)), each image is extended to a higher dimensional tensor of two scales, τ_1: an identity mapping of the original image; τ_2: the low-frequency scale extracted by wavelet decomposition. During the inference phase (Fig. 1(b)), the input image is pre-processed in a similar way, except that τ_1 is now a wavelet denoising function applied to the original image.

Both of these two transformations play critical roles in increasing robustness. In order to fool the target model, the two parts must be affected in a sophisticated and collaborative way at the same time. However, since the adversary can only make changes to the original image x, it is difficult to affect the model output by small perturbations. Below, we elaborate and validate each technique.

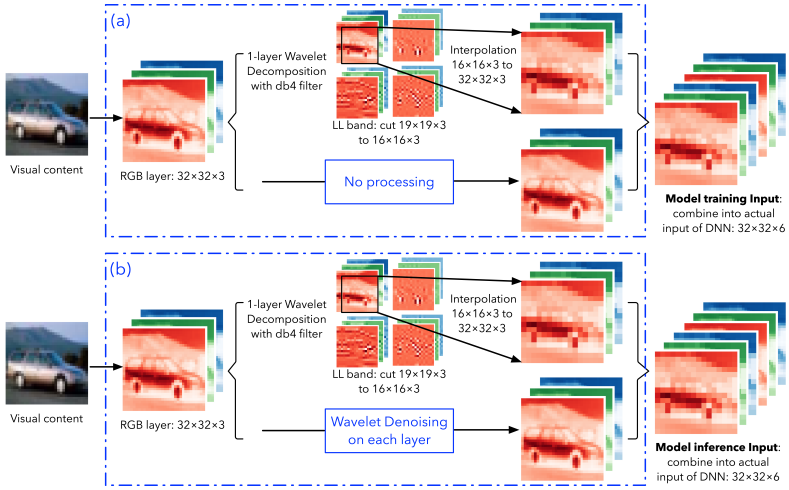

Fig. 1. Methodology Overview. (a) Training phase; (b) Inference phase.

3.2 Wavelet Extension

As shown in Fig. 1, `WED` adopts Wavelet Extension (WavExt) to extract low-frequency information and extend the input. This process can build an image extension that represents the basic visual structures, which can be hardly influenced by adversarial perturbations and can assist the model prediction. This is done by extending the RGB image from $N \times N \times 3$ into $N \times N \times 6$ by adding an image obtained by a reconstruction algorithm based on wavelet transform.

Generally, the wavelet transform represents any arbitrary signal as a superposition of wavelets. Discrete Wavelet Transformation (DWT) decomposes a signal into different levels. DWT decomposes one signal into a low-frequency band (L band) and a high-frequency band (H band) with equal size. For image processing, the DWT is normally processed in a two-dimensional manner as 2D-DWT in two directions: vertical and horizontal. There will be four sub-bands generated as shown in Fig. 2: LL band, LH band, HL band, and HH band. With the proper choice of the wavelet filter, the image can be decomposed into different frequency bands representing various elements such as basic structures, details, etc. As we can observe in Fig. 2, the LL band is an abstract of the basic image structures while the rest three bands represent the image details.

Algorithm 1 shows the detailed steps to process the images. After one level of two-dimensional DWT (Line 1), there are four sub-bands generated each of size $floor(\frac{N-1}{2}) + n$, with $n = 4$ as the filter length. We crop the LL band of each RGB layer to $\frac{N}{2} \times \frac{N}{2}$ (Line 2), and then perform the Bicubic interpolation to resize the extracted low-frequency component back to $N \times N$ (Line 4).

We visually show the effectiveness of the wavelet extension by the saliency map [14]. The saliency map is obtained by first calculating gradients of outputs of the penultimate layer with respect to the input images, and then by normalizing

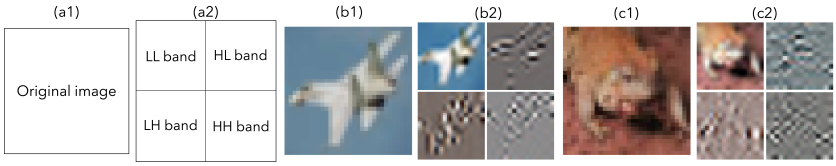

Fig. 2. Examples of 2D-DWT decomposition. 2D-DWT operations: (a1) and (a2); two images as examples: (b1) and (b2), (c1) and (c2).

ALGORITHM 1: `WED`: Wavelet Extension and Wavelet Denoising.

Input: Image x, size $N \times N \times 3$
Output: processed tensor $x_{Extended}$, size $N \times N \times 6$

1: LL, LH, HL, HH = wavelet-decomposition(x)
2: The size of LL is cropped to $N/2 \times N/2 \times 3$
3: The value of LL is re-scaled to [0, 255]
4: x_{Ext} = Bicubic-interpolation(LL)
5: **if** Inference phase **then**
6: $x_{Extended}$ = concat(wavelet-denoising(x), x_{Ext})
7: **else**
8: $x_{Extended}$ = concat(x, x_{Ext})
9: **end if**

the absolute values of those gradients. A brighter pixel in the saliency map means that the corresponding pixel in the original image has more influence on the model's output. Fig. 3 shows that although the saliency maps of a clean image (e) and its AE (f) are very different, the saliency maps of their extended components are nearly identical (see (g) and (h)). This indicates that the wavelet extension can effectively remove the effects of adversarial perturbations.

During training phase, we concatenate the extended component with the original input as the actual input tensor of size $N \times N \times 6$ (Line 8). We adjust the model structure to accept this input size only by changing the dimension of weights in the first layer. The reason that we include the original image is that they still keep details of the images, which can assist the classification and maintain accuracy. However, the existence of those original images provides adversaries with opportunities to manipulate the model behaviors. We introduce another mechanism to eliminate this threat during inference phase.

3.3 Wavelet Denoising

During inference phase, an extra non-differentiable transformation is applied to the original input (Fig. 1(b)). Due to the different pre-processing steps between training phase and inference phase, the AEs generated using the gradients of the trained weights will not have the optimized result as desired by the adversary.

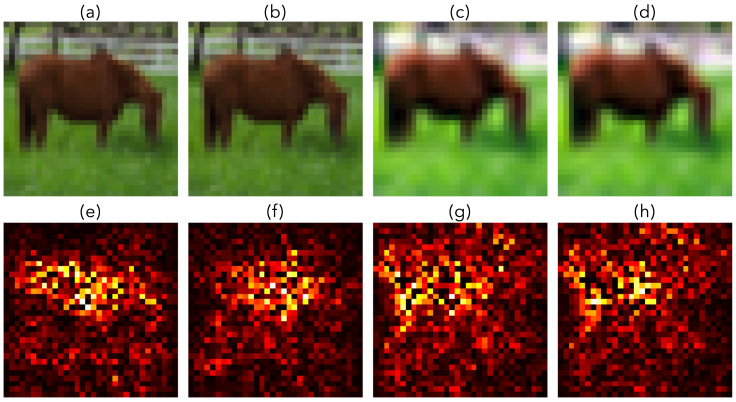

Fig. 3. Similar saliency maps between wavelet extension results ((g) and (h)). (a) and (e): a clean image and its saliency map; (b) and (f): corresponding AE image and its saliency map; (c) and (g): wavelet extension of image (a) and its saliency map; (d) and (h): wavelet extension of AE image (b) and its saliency map.

We implement the wavelet denoising method by combining two approaches, VisuShrink and BayesShrink [3]. Normally wavelet denoising relies on the basic assumption that the noise tends to be represented by small values in the frequency domain. These small values can be removed by setting coefficients below a given threshold to zero (hard threshold) or by shrinking different coefficients to zero by a soft threshold. First, we use the VisuShrink to set a threshold to remove additive noise. For an image X with N pixels, this threshold is given by $\sigma\sqrt{2\log N}$, where the σ is normally smaller than the standard deviation of noise. Second, we adopted the method from [12] and use the BayesShrink based on a soft threshold. We model the threshold for each wavelet coefficient as a Generalized Gaussian Distribution (GGD). The optimal threshold can be further approximated by $\frac{\sigma^2}{\sigma_x}$ where σ_x and β are parameters of the GGD for each wavelet sub-band (Eq. 5). Normally, an approximation of T_h, as shown on the right side of Eq. 5, is used to adapts to the amount of noise in the given image. We adopted the parameter settings from [12].

$$T_h^*(\sigma_x, \beta) = \underset{T_h}{\operatorname{argmin}} E(\widehat{X} - X)^2 \approx \frac{\sigma^2}{\sigma_x} \tag{5}$$

4 Evaluations

4.1 Experimental Settings and Implementations

Dataset and models. We consider an image classification task on CIFAR-10 with 50,000 images for training and 10,000 images for testing. Each image $(32 \times 32 \times 3)$ belongs to one of ten classes. All pixel values are normalized to the

range $[0, 1]$. The target model is ResNet-29 [6]. It consists of 29 layers containing three bottleneck residual blocks with channel sizes 64, 128, 256, respectively. We use Keras package with Tensorflow backend to implement the model. The training is done by using Adam optimization with its hyper-parameters $\beta_1 = 0.9, \beta_2 = 0.999$. The model reaches the Top-1 accuracy of 92.27% over the testing set after about 150 epochs. Experiments are done on a platform with Intel(R) Core(TM) i7-8700K CPU @ 2.40GHz and NVIDIA GeForce GTX 1080 Ti GPU.

Attack and defense implementations. We test our defense by 7 common AE generation methods: FGSM [15], I-FGSM [7], MI-FGSM [4], L-BFGS [5], JSMA [10], DeepFool [9], CW [2]. All attacks are implemented with the help of the CleverHans library (v3.0.1). We consider two attack scenarios. (1) Black-box scenario: the adversary does not have access to the model parameters, defense mechanism, etc. (2) White-box scenario: the adversary has detailed knowledge of the target model including the trained parameters and defense mechanism. Since the size of input after wavelet extension becomes $N \times N \times 6$, we train another ResNet-29 model by only changing the dimension of weights of the first layer to accept the extended input. The Top-1 testing accuracy reaches 91.96%, which means that `WED` does not influence the performance of the target model.

Under different attacks, we compare `WED` with two wavelet transform-based defense solutions: Wavelet Approximation (`WA`) [13] and Pixel Deflection (`PD`) [12]. `WA` uses level-1 wavelet approximation on the input image to get low-resolution images. `PD` randomly replaces pixels by other pixels randomly selected within a small window and use Bayeshrink wavelet denoising to reduce adversarial noise. For each experiment, we consider the targeted attack, where a new label different from the correct one is selected as the adversary's target. We randomly select 100 samples which are correctly predicted by the original model from the test set, generate the corresponding AEs and measure the top-1 accuracy.

4.2 Black-box Scenario

In the black-box scenario, the adversary does not know the model parameter. He trains another shadow model with the same network architecture (ResNet-29) to generate AEs. Table 1 shows the top-1 prediction accuracy on the generated AEs for the models without any protection (baseline), with `WA`, `PD` and `WED`, respectively. To clearly show the magnitude of AEs distortion compared with the original images, we calculate the average normalized L_{inf} and L_2 distance.

We observe that our defense is effective towards all kinds of attacks, and outperforms both `WA` and `PD` in most cases. Particularly, for attacks with larger L_{inf} distortion (e.g., I-FGSM, and MI-FGSM), `PD` is less effective. In contrast, our defense still shows strong resistance. Moreover, `WED` has better accuracy on clean images than `WA` or `PD`. Such high prediction accuracy on both clean images and AEs is due to the concatenation of two scales of information, details from the wavelet denoised part and structures from the wavelet extended part.

Table 1. The Top-1 accuracy on **baseline**, `WA`, `PD`, `WED` in the black-box scenario.

Attack	L_{inf}	L_2	**Baseline**	`WA`	`PD`	`WED`
Clean	0.0	0.0	1.0	0.89	0.92	**0.97**
FGSM	0.005	0.28	0.39	0.84	0.84	**0.94**
I-FGSM	0.005	0.21	0.21	0.85	0.86	**0.93**
MI-FGSM	0.005	0.25	0.29	0.84	0.86	**0.92**
JSMA	0.832	4.12	0.0	0.60	0.49	**0.68**
DeepFool	0.015	0.12	0.0	0.87	0.95	**0.95**
LBFGS	0.018	0.15	0.0	0.86	0.92	**0.97**
CW	0.011	0.09	0.0	0.86	0.95	**0.97**

4.3 White-box Scenario

In this scenario, the adversary knows the exact values of the model parameters. Then he can directly generate AEs based on the target model. In `WED`, the model input are the tensor ($N \times N \times 6$) extended by wavelet transformations, while the adversary can only provide AEs that have three channels ($N \times N \times 3$). Due to the non-differentiability of the transformations, it is hard for the adversary to adjust the original input with small perturbations to change the output tensors to the malicious ones. Thus in this scenario, we assume the adversary will directly use half of the calculated adversarial tensors as the input.

Table 2. Top-1 accuracy on **WavExt** and `WED` in the white-box scenario.

Attack	L_{inf}	L_2	**WavExt**(without denoising)	`WED`
FGSM	0.005	0.28	0.48	**0.62**
I-FGSM	0.005	0.21	0.24	**0.48**
MI-FGSM	0.005	0.24	0.28	**0.50**
JSMA	0.898	4.95	0.19	**0.68**
DeepFool	0.015	0.12	0.85	**0.95**
LBFGS	0.017	0.15	0.77	**0.97**
CW	0.012	0.09	0.87	**0.97**

Existing approaches like `WA` or `PD` cannot fit into this scenario as the model input and pre-processed input have different dimensions. So we only compare the effectiveness of `WED` and the one with only Wavelet Extension (Section 3). The results are shown in Table 2. The comparison results indicate that the robustness is increased with the wavelet denoising at the inference phase. We observe that our defense still shows strong resistance against some attacks even the adversary knows the parameters, especially for DeepFool, LBFGS, and CW.

5 Conclusion

In this paper, we propose a novel approach to defend DNN models against adversarial examples. First, we apply a wavelet transform to extend the input

images with their structures and basic elements. Second, we utilize wavelet denoising to further reduce the impact of the adversarial perturbations. These two non-differentiable operations can increase the difficulty of generating adversarial perturbations while maintaining the model performance. Our approach provides better robustness and effectiveness over other wavelet-based solutions in defeating different popular adversarial attacks under different scenarios.

References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. Proceedings of the 35th International Conference on Machine Learning, ICML (2018)
2. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE (2017)
3. Chang, S.G., Yu, B., Vetterli, M.: Adaptive wavelet thresholding for image denoising and compression. IEEE transactions on image processing **9**(9), 1532–1546 (2000)
4. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9185–9193 (2018)
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
6. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision. pp. 630–645. Springer (2016)
7. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
8. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
9. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
10. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS&P). pp. 372–387. IEEE (2016)
11. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP). pp. 582–597. IEEE (2016)
12. Prakash, A., Moran, N., Garber, S., DiLillo, A., Storer, J.: Deflecting adversarial attacks with pixel deflection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8571–8580 (2018)
13. Shaham, U., Garritano, J., Yamada, Y., Weinberger, E., Cloninger, A., Cheng, X., Stanton, K., Kluger, Y.: Defending against adversarial images using basis functions transformations. arXiv preprint arXiv:1803.10840 (2018)
14. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
15. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)