Refining the Additive Random Noise Detection Method for Adversarial Examples

Chaoqi LIU
liuchaoqi730@gmail.com

Abstract

Deep neural networks have become state-of-the-art tools for many machine learning tasks. However, it is known that adversarial examples can fool deep neural networks easily and cause serious consequences. The existence of threats caused by adversarial attacks motivate the development of defense mechanisms. Among the proposed defenses, the additive random noise detection method stands out for being both simple and fast. However, due to the stochasticity of the additive noise samples, the results from this detection method are often noisy and inconsistent. Therefore, we proposed a new *Ensemble Additive Random Noise Detection Method* which aims to reduce the noise in detection by using an ensemble of samples. The *Ensemble Method* shows a significant improvement in the detection rate of adversarial examples while inheriting the simplicity and speed of the additive random noise detection method.

1 Introduction

Deep neural networks have become the state-of-the-art in machine learning for image recognition [28], natural language processing [25], speech recognition [13] and reinforcement learning [3]. These models can be applied in fields such as self-driving cars [9], facial recognition and robotics. However, it is widely known that deep neural networks can be easily fooled by adversarial examples — input examples that an attacker has intentionally designed to cause machine learning models to make mistakes. Adversarial examples force the model to output the incorrect answer which may lead to dangerous consequences. In 2014, Szegedy et al. first noticed that neural networks can be fooled by imperceptible changes in the input [21]. Further research found that due to the vulnerability of neural networks, adversarial examples can be easily produced [8].

Malicious attacks can happen in almost all machine learning tasks, including image classification [1], sound recognition [6], natural language processing [2]. Since the discovery of adversarial examples, they have become a universal problem in machine learning, and there is currently no perfect solution. Adversarial examples are specifically designed by attackers and are outside the manifold of the training images. Therefore, improving the accuracy of the model is not enough to resist adversarial attacks.

In the domain of image recognition, models are designed to process images and provide descriptions or recognize the objects in the input images. Currently, deep neural networks already have a recognition accuracy that exceeds that of humans [11]. However, adversarial examples presented in the form of images (adversarial images) with subtle changes can interfere with the prediction of image recognition models, causing the models to output an incorrect answer with high confidence. Such problems limit the applications of neural networks. For example, in the case of self-driving cars, adversarial examples may cause serious accidents [17, 20].

The existence of threats caused by adversarial attacks motivate the development of defense mechanisms which can be simply divided into two categories. One approach aims to detect adversarial inputs and reject them. These detection methods [14, 15] include additive random noise detection

method [26] and defense GANs [19]. An alternative approach aims to improve the robustness of the model itself to defend against adversarial attacks. These defense methods include recovering semantic meaning through augmentation [4, 10] and adversarial training [12].

Defense methods require re-training to improve the robustness of models and usually have low accuracy. Because the underlying cause of adversarial examples and the performance of deep neural networks are not yet well-understood theoretically, it is difficult to directly improve robustness while maintaining accuracy. Hence the relatively easier detection methods are good alternative approaches. We do not need to completely understand these unsolved problems to design a detection mechanism based on the observed properties of adversarial examples. For example, studies show that natural images are relatively robust to random noise [21]. The additive random noise detection method exploits this fact to determine whether an input example is an adversarial example by measuring the degree of robustness of input examples to additive random noise. This detection method is fast, simple, and model-agnostic, which means it can be applied to any existing machine learning model. In addition, the additive random noise detection method does not require expensive re-training of the classification models.

However, there are some problems with the use of the additive random noise detection method. The detection process only uses a single sample of additive random noise to produce a rough detection result. In other words, the detection process is noisy and inconsistent. Therefore, we propose the *Ensemble Additive Random Noise Detection Method* which is a refinement of the original method that uses an ensemble of additive noise samples to reduce the noise in the detection result. In our experiments, we demonstrate a substantial improvement in the detection rate of adversarial examples using this new *Ensemble Method*.

2 Background

2.1 Attack overview

Adversarial attacks can be simply categorised into black-box, grey-box or white-box settings. In a black-box setting, the attacker can only send information to the system and receive a prediction about the class. While in a white-box setting, the attacker is usually assumed to be an insider, so everything about the network is known, including model parameters, the dataset, and even the defense mechanism. So far, there is no standard method to deal with white-box attacks. In this paper, we focus on the grey-box setting where the attacker has full knowledge of the classification model but does not have access to the defense mechanism.

2.2 Attack algorithms

Adversarial examples exist outside the manifold of the training images. Hence, their behavior can be manipulated at will. These regions can be effectively discovered by a gradient-guided search, also referred to as a gradient-based attack.

2.2.1 Carlini-Wagner (CW)

In 2016, Carlini and Wagner [5] considered the adversarial attack as an optimization problem. The attack framework can be expressed as

$$\begin{aligned} &\text{minimize } D(x, x+\delta) + c \cdot f(x+\delta) \\ &\text{such that } x+\delta \in [0,1]^n \end{aligned} \tag{1}$$

where x is the input image, δ is the perturbation, D is the distance between the input image and the adversarial image, c is a chosen constant, and f is the objective function.

For an L_2 attack, a new variable w is introduced for optimization, set to

$$\delta_i = \frac{1}{2}(\tanh(w_i) + 1) - x_i. \tag{2}$$

The attacker aims to find the w that can solve

$$\text{minimize } \|\frac{1}{2}(\tanh(w) + 1) - x\|_2^2 + c \cdot f(\frac{1}{2}\tanh(w) + 1), \tag{3}$$

with objective function f defined as

$$f(x') = \max(\max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa), \tag{4}$$

where t is the class that x' will be classified as, κ is the confidence that we expect the classification model to have on the adversarial example, and $Z(x')$ is the logit, i.e. the output vector of probabilities for an input x'.

2.2.2 Projected Gradient Descent (PGD)

In 2017, a L_∞ based attack algorithm — Projected Gradient Descent attack — was introduced [16], which is also known as the iterative Fast Gradient Sign Method (iterative-FGSM). FGSM uses the gradients of the loss referencing the input example to create an adversarial example that maximises the loss.

FGSM is defined as

$$\mathbf{x}' = \mathbf{x} + \epsilon \operatorname{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\theta, \mathbf{x}, y)), \tag{5}$$

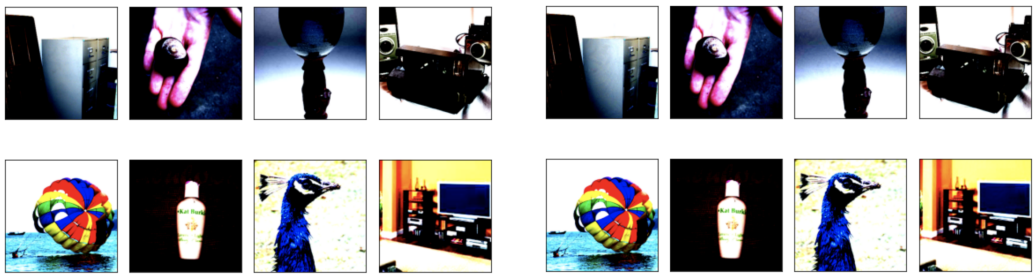
where $\mathbf{x}'$ is the adversarial image generated, $\mathbf{x}$ is the original input image, y is the original input label, $\mathcal{L}$ is measured as cross-entropy loss, ϵ is the multiplier to ensure the perturbations are small and θ is the model parameters.

The PGD attack is the iterative version of FGSM, defined as

$$\mathbf{x}^{t+1} = \prod_{\mathbf{x}\pm\epsilon} \left(\mathbf{x}^t + \alpha \operatorname{sign}\left(\nabla_{\mathbf{x}} \mathcal{L}(\theta, \mathbf{x}, y)\right)\right), \tag{6}$$

where $\prod_{\mathbf{x}\pm\epsilon}$ indicates that the image will be projected back to the space which is bounded by $\mathbf{x} \pm \epsilon$ and α is the learning rate. PGD can be simply interpreted as a multi-step version of the FGSM attack.

Using a small enough ϵ, the changes to images are nearly imperceptible, and the adversarial images cannot be recognised by a human observer (see Fig.1).



(a) Original (b) Adversarial

Figure 1: An illustration of the original and adversarial versions of images from ImageNet. These adversarial images are generated via Projected Gradient Descent attack with epsilon 0.03, learning rate 2/255 and 40-step iteration.

2.3 Defense mechanisms and their limitations

Detection methods [14, 15], including additive random noise [26] and defense GANs [19], are designed to detect adversarial inputs and reject them. However, the detection methods do not directly improve the robustness of classification models. Defense GANs have an additional significant shortcoming. GANs are notoriously hard to train and require re-training for each new model, and therefore, transferability of defense GANs to general models would be difficult.

Defense methods, such as recovering semantic meaning via augmentation [4, 10] and adversarial training [12], aim to improve the robustness of the model itself to defend against adversarial attacks. Recovering semantic meaning via augmentations [4, 10] uses rotations and transformations to improve the robustness of the model. However, one has to re-train the classification model, which is time-consuming. Adversarial training only uses single robust features to improve the robustness of the models, and it also requires the model to be re-trained. Additionally, adversarial training significantly reduced the accuracy of the model [22, 27].

3 The Additive Random Noise Detection Method

Detection methods are test-time methods. The model rejects adversarial images and outputs the prediction for natural images. The additive random noise method is a standard detection method. Simple and fast are the prominent advantages of this method. Additionally, this detection method is model-agnostic, which means one can apply this method on any model without re-training. The additive random noise detection method can be considered as a method for preliminary screening, which means one can combine this method with other adaptive methods to obtain a better result.

Natural images are robust to random noise [21]. It is also known that features of natural images extracted by convolutional neural networks are robust to random corruption in the input [21, 24, 10]. Hence, in most cases, we expect the input natural images to be more robust to random noise than adversarial images. The additive random noise should not lead to changes in the predicted label of a given input. The additive random noise detection method follows this intuition.

The random noise is generated as $\epsilon \sim N(0, \sigma^2 I)$ (σ is the variance or noise radius, I is the identity matrix). We then compute $\Delta = \|h(\mathbf{x}) - h(\mathbf{x}+\epsilon)\|_1$ ($h(\mathbf{x})$ is the output vector of probabilities by the classification model). The input $\mathbf{x}$ will be considered as adversarial images if Δ is sufficiently large. After setting the threshold Δ^* (a hyperparameter), if the input image has $\Delta > \Delta^*$, then the input is considered to be adversarial.

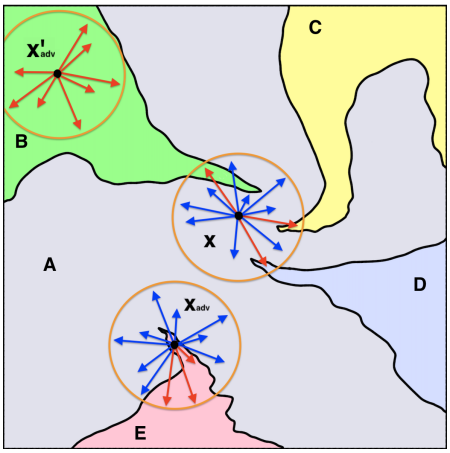


Figure 2: A 2D illustration of the high dimensional decision boundary around the natural images $\mathbf{x}$. $\mathcal{A}$ is the true class of $\mathbf{x}$. $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{E}$ are the incorrect classes. The orange circle shows the region where the input sample can be pushed to by additive random noise. Red arrows and blue arrows are examples of noise that lead to incorrect and correct detection respectively.

Fig.2 illustrates the principle behind the additive random noise detection method. Adversarial attacks will generate adversarial images $\mathbf{x}_{\text{adv}}$ that are close to the decision boundary. The adversarial images $\mathbf{x}_{\text{adv}}$ can be randomly perturbed to the true class $\mathcal{A}$ because they are not deep inside the incorrect class $\mathcal{E}$. The additive random noise that can perturb $\mathbf{x}_{\text{adv}}$ to class $\mathcal{A}$ is shown by blue arrows in Fig.2. In contrast, natural images $\mathbf{x}$ tend to be far from the decision boundary, so adding random noise does not generally change their classification. Therefore, we can distinguish adversarial images and reject them.

4 *Ensemble Additive Random Noise Detection Method*

The additive random noise pushes the input example one step in a random direction. As a result, the detection process of the additive random noise detection method is noisy and inconsistent due to the incorrect detections caused by the random noise in rare cases (see Fig.2). Therefore, theoretically, by taking more samples of random noise and executing more tests, the detection process can be made more consistent and less noisy, leading us to propose the *Ensemble Method*.

The theoretical detection strategy can be expressed as follows. We generate n samples of additive random noise via

$$\epsilon_i \sim N(0, \sigma_i^2 I), \quad i \in \{1, 2, \ldots, n\} \tag{7}$$

with different or same noise radius σ_i, and set n thresholds corresponding to additive random noise Δ_i^*. Then, we compute the difference in the output vector of probabilities of the model $h(\mathbf{x})$ corresponding to the additive random noise sample,

$$\Delta_i = \|h(\mathbf{x}) - h(\mathbf{x} + \epsilon_i)\|_1. \tag{8}$$

For each test, we obtain a result

$$d_i = \begin{cases} 1 & \text{if } \Delta_i > \Delta_i^* \\ 0 & \text{if } \Delta_i \leq \Delta_i^* \end{cases}. \tag{9}$$

Taking the average of these results,

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i. \tag{10}$$

According to the majority rule, if $\bar{d} > 0.5$, we will consider the input example as an adversarial example. Otherwise, the input example will be considered as a natural example.

5 Experiments

We test the performance of the *Ensemble Additive Random Noise Detection Method* against Projected Gradient Descent [16] and Carlini-Wagner [5] attacks (grey-box setting) and release our code publicly.[1] We conduct our experiments on ImageNet [7] and use the pre-trained convolutional neural network — ResNet101 [11] in PyTorch. We sample 100 images that ResNet101 has successfully classified from ImageNet for further experiments. As expected, the PGD and CW attacks are extremely effective against ResNet101 trained on ImageNet. The parameters of attack algorithms we used in the experiment are shown in Appendix A, and the performance of the attack algorithms on ResNet101 is shown in Appendix B.

The *Ensemble Method* is very fast (Appendix C). To execute one test, the time taken is 0.036 seconds on average, which is almost negligible, and for the multi-noise-radius tests, we can compute them in parallel. The speed of this method makes it a suitable choice for preliminary screening.

5.1 Detection results for the *Ensemble Method*

We first implement the additive random noise detection method and then apply the *Ensemble Method*. The detection results indicate that there is an improvement in the detection rate after deploying the *Ensemble Method*. The true positive rates (TPR[2]) are shown in Table.1. In our experiments, we fix the false positive rate (FPR[3]) at 0.1. Comparing the original additive random noise method with our *Ensemble Method*, we observe a significant improvement in the average TPR and conclude that, for the same noise radius, increasing the number of tests to 2 or 3 results in improved detection rates.

PGD			CW		
noise radius	FPR	TPR ($\pm$0.01)	noise radius	FPR	TPR ($\pm$0.01)
0.1	0.1	0.84	0.01	0.1	0.88
0.2	0.1	0.94	0.02	0.1	0.88
0.3	0.1	0.91	0.03	0.1	0.86
0.1, 0.1	0.1	0.88	0.01, 0.01	0.1	0.90
0.1, 0.2	0.1	0.96	0.01, 0.02	0.1	0.90
0.1, 0.3	0.1	0.95	0.01, 0.03	0.1	0.91
0.2, 0.2	0.1	0.97	0.02, 0.02	0.1	0.90
0.2, 0.3	0.1	0.95	0.02, 0.03	0.1	0.91
0.3, 0.3	0.1	0.94	0.03, 0.03	0.1	0.90
0.1, 0.1, 0.1	0.1	0.88	0.01, 0.01, 0.01	0.1	0.90
0.2, 0.2, 0.2	0.1	0.97	0.02, 0.02, 0.02	0.1	0.91
0.3, 0.3, 0.3	0.1	0.95	0.03, 0.03, 0.03	0.1	0.90

Table 1: The table shows the average detection results (average TPR) of additive random noise detection method and the *Ensemble Method*.

[1] https://github.com/Ivan-Lcq/RefiningTheAdditiveRandomNoiseDetectionMethodForAdversarialExamples

[2] *True positive rate* measures the proportion of positives that are correctly identified.

[3] *False positive rate* measures the proportion of positives that are incorrectly identified.

5.2 Effect of the number of tests on performance

We see that TPR increases significantly when 2 or 3 tests are performed (see Fig.3). However, when the number of tests is increased beyond 3, the TPR no longer improves and instead begins to fluctuate. Hence, in our case, it is optimal to perform 2 or 3 tests. In general, the optimal number of tests should be found by experimentation.

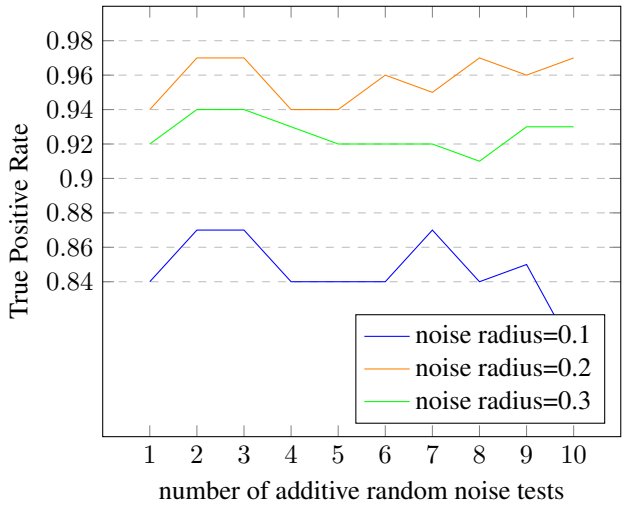


Figure 3: The relationship between the number of additive random noise tests and TPR of different noise radius is shown in the diagram. The blue line, orange line, and green lines represent the relationship between TPR and the number of additive random noise tests with noise radii of 0.1, 0.2, and 0.3, respectively. When the number of tests increases, the TPR fluctuates greatly.

6 Discussion

Adversarial attacks happen in many domains [1, 2, 6] and currently have no clear solution. In this work, we refine the existing additive random noise detection method [26] by proposing the *Ensemble Additive Random Noise Detection Method*. Our experimental results show that using the *Ensemble Method* results in an improvement in the detection rate of adversarial examples.

The *Ensemble Method* is a superior version of the additive random noise detection method [26] and can be considered a simple and easy-to-use tool for preliminary screening. Other prevalent defense mechanisms [4, 10, 12, 19] usually require re-training for each classification model, while the *Ensemble Method* does not require re-training and is model-agnostic. The *Ensemble Method* can also be combined with other defense methods that improve the robustness of models. To be specific, we can run the *Ensemble Method* to first screen out and reject some of the adversarial images before passing the filtered images to a robust model with additional defense methods applied. In the future, we will combine the *Ensemble Method* with other defense methods, e.g. adversarial training, to investigate whether detection methods and defense methods are likely to complement each other and together obtain better performance.

The *Ensemble Method* is simple and easy to use. There are only a few hyperparameters: the noise radius σ, the threshold Δ^*, and the number of tests n. Additionally, the *Ensemble Method* appears to be fairly robust to changes in hyperparameters. The most critical value is the number of tests n. Based on our experiments, only 2 or 3 tests are enough to obtain better detection results. However, the incorrect choice of n may cause the detection rate to be suboptimal. As our experiments show, when the number of tests increases beyond 3, the TPR fluctuates significantly. We believe that this may be the consequence of the majority rule aggregation method and plan to carry out further work to better understand this phenomenon. We also plan to carry out experiments on a wider variety of classification models and image datasets.

Despite our positive results, the *Ensemble Method* still retains some of the limitations of the additive random noise detection method. The additive random noise detection method has been successfully applied to defend against black-box and grey-box attacks [10, 23, 18]. However, this method alone is not sufficient to defend against white-box attacks. Since a white-box attacker has a comprehensive understanding of the defense mechanism, the attacker can optimize the adversarial loss via Monte Carlo sampling of noise vectors, resulting in the adversarial images $\mathbf{x}'_{\text{adv}}$ generated deep inside the incorrect class $\mathcal{B}$ (see Fig.2). Research has proven that this method can successfully bypass the additive random noise detection method [26]. There is currently no perfect solution to defend against white-box attacks, so there is still a great need for future research to develop and study novel defense mechanisms.

7 Conclusion

Adversarial attacks happen in many machine-learning domains and have serious consequences, such as accidents in self-driving cars [17, 20]. We propose a new *Ensemble Additive Random Noise Detection Method* to refine the existing additive random noise detection method and show that the *Ensemble Method* achieves a substantial improvement in detection rate. In contrast to other defense mechanisms, the *Ensemble Method* is fast, model-agnostic, and easy to use, and we believe it should be the new standard for performing additive random noise-based detection of adversarial examples.

Acknowledgments

I would like to express my sincere appreciation to Wenting Zhao for the consistent mentoring and support. I also wish to extend my deepest gratitude to Peter Y. Lu for the enriching discussion on adversarial attack algorithms and defense mechanisms, and the anonymous reviewers for their valuable feedback.

References

[1] Yigit Alparslan, Ken Alparslan, Jeremy Keim-Shenk, Shweta Khade, and Rachel Greenstadt. Adversarial attacks on convolutional neural networks in facial recognition domain, 2021.

[2] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[3] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A brief survey of deep reinforcement learning. *CoRR*, abs/1708.05866, 2017.

[4] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018.

[5] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644, 2016.

[6] Nicholas Carlini and David A. Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. *CoRR*, abs/1801.01944, 2018.

[7] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[9] Sorin Mihai Grigorescu, Bogdan Trasnea, Tiberiu T. Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *CoRR*, abs/1910.07738, 2019.

[10] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. *CoRR*, abs/1711.00117, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[12] Joong-Won Hwang, Youngwan Lee, Sungchan Oh, and Yuseok Bae. Adversarial training with stochastic weight average, 2020.

[13] Yogesh Kumar and Navdeep Singh. A comprehensive view of automatic speech recognition system - a systematic literature review. pages 168–173, 04 2019.

[14] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. *CoRR*, abs/1612.07767, 2016.

[15] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Michael E. Houle, Grant Schoenebeck, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *CoRR*, abs/1801.02613, 2018.

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[17] Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, and Yuval Weisglass. Fooling a real car with adversarial traffic signs. *CoRR*, abs/1907.00374, 2019.

[18] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. *CoRR*, abs/1902.04818, 2019.

[19] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *CoRR*, abs/1805.06605, 2018.

[20] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. DARTS: deceiving autonomous cars with toxic signs. *CoRR*, abs/1802.06430, 2018.

[21] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[22] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy, 2019.

[23] Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *CoRR*, abs/1805.11770, 2018.

[24] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. *CoRR*, abs/1711.01991, 2017.

[25] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *CoRR*, abs/1708.02709, 2017.

[26] Tao Yu, Shengyuan Hu, Chuan Guo, Wei-Lun Chao, and Kilian Q. Weinberger. A new defense against adversarial images: Turning a weakness into a strength. *CoRR*, abs/1910.07629, 2019.

[27] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S. Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack, 2019.

[28] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *CoRR*, abs/1807.05511, 2018.

Supplementary Material: Refining the Additive Random Noise Detection Method for Adversarial Examples

A The parameters of attack algorithms

The parameters of attack algorithms. These parameters remain the same in the experiments.

Table 2: Parameters of attack algorithms

Parameters	PGD attack	CW attack
iteration	40	1000
learning rate	2/255	0.01
epsilon	0.03	—
constant c	—	0.1
targeted	false	false

B Attack results

Clean test accuracy of ResNet101 on 100 images that has been successfully classified by ResNet101. Robust accuracy under PGD attack and CW attack respectively.

Table 3: Performance of ResNet101

	test accuracy
clean	1
PGD attack	0
CW attack	0.36

C Detection time

Table 4: Time taken to generate random noise

Attack algorithm	test number	average time taken (second)
PGD attack	1	0.035
	2	0.070
	3	0.105
CW attack	1	0.037
	2	0.077
	3	0.105