Detection of Adversarial Examples in Deep Neural Networks with Natural Scene Statistics

Anouar Kherchouche[⋆§] Sid Ahmed Fezza[⋆] Wassim Hamidouche[§] and Olivier Déforges[§]
[⋆]*National Institute of Telecommunications and ICT*
Oran, Algeria
[§]*Univ. Rennes, INSA Rennes, CNRS, IETR - UMR 6164*
Rennes, France
{akherchouche,sfezza}@inttic.dz, {wassim.hamidouche,olivier.deforges}@insa-rennes.fr

***Abstract*—Recent studies have demonstrated that the deep neural networks (DNNs) are vulnerable to carefully-crafted perturbations added to a legitimate input image. Such perturbed images are called *adversarial examples (AEs)* and can cause DNNs to misclassify. Consequently, it is of paramount importance to develop detection methods of AEs, thus allowing to reject them. In this paper, we propose to characterize the AEs through the use of natural scene statistics (NSS). We demonstrate that these statistical properties are altered by the presence of adversarial perturbations. Based on this finding, we propose three different methods that exploit these scene statistics to determine if an input is adversarial or not. The proposed detection methods have been evaluated against four prominent adversarial attacks and on three standards datasets. The experimental results have shown that the proposed methods achieve a high detection accuracy while providing a low false positive rate.**

***Index Terms*—Adversarial examples (AEs), deep neural networks (DNNs), detection, natural scene statistics.**

I. Introduction

Deep neural networks (DNNs) models have led to impressive performance in various domains, especially in image classification task where they achieved near human performance [1]. However, despite the remarkable progress of DNNs, it has been found that they are vulnerable to adversarial attacks. Szegedy *et al.* [2] demonstrated that adding a small imperceptible perturbation to a correctly classified image can cause a DNN classifier to make incorrect predictions with high confidence. While a human observer cannot distinguish between the original image and the perturbed/attacked one. Such perturbed inputs that can fool the DNNs are called *adversarial examples (AEs)*.

This vulnerability brings up questions about the relevance of using the DNN models in sensitive applications such as autonomous cars, biometric, video surveillance, healthcare etc, where AEs can lead to fatal consequences.

Consequently, many defense mechanisms have been proposed attempting to correctly classify AEs and thereby increasing model's robustness. These defenses can be grouped under three different approaches: (1) augmenting the training data with AEs, *e.g.*, *adversarial training* [3], (2) modifying the training procedure to reduce the amplitude of network gradients exploited by adversaries to generate AEs, *e.g.*, *defensive distillation* [4], and (3) trying to remove the adversarial perturbation from the input samples [5]–[8].

However, most of these defense solutions are not effective enough at classifying AEs correctly, especially against new/unknown attacks or when the attacker knows the details of defense mechanism [9]–[11]. Therefore, many recent works have focused on detecting AEs instead. The detection of AEs may be useful to warn users or to take security measures in order to avoid tragedies. Furthermore, for online machine learning service providers, the detection can be exploited to identify malicious clients and reject their inputs [12]. Finally, combining a detection method with defenses that attempt to remove the adversarial noise may prove beneficial.

Several methods have been proposed to detect the AEs [10]–[20]. Some of them are based on the statistical properties of input or network parameters, others train a separate detector to classify images as clean or adversarial, and finally other methods exploit the prediction inconsistency. However, as shown in [11], the existing AEs detection methods are effective against some specific attacks, but fail to detect new or more powerful ones. In addition, most of the detection methods reported high detection accuracy, but have also obtained high false positive rate, meaning that they reject a significant amount of clean images, which can be considered a failure of these detection approaches.

In this paper, we propose a novel approach for detecting AEs based on natural scene statistics (NSS) using three different ways. The three proposed methods are based on the assumption that the presence of adversarial perturbations alters some statistical properties of natural images. Thus, quantifying these statistical outliers, *i.e.,* deviations from the regularity, using scene statistics enables the building of a binary classifier capable of classifying a given image as legitimate or adversarial. The experimental results on three widely used datasets, namely MNIST, CIFAR-10 and ImageNet, showed that the proposed detection methods achieves high detection accuracy, while providing a low false positive rate.

The rest of this paper is organized as follows: Section II reviews some attack techniques and detection methods that have been proposed in the literature. Section III describes the proposed approach. The experimental results are presented in Section IV. Finally, Section V concludes the paper.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

II. Related Work

In this section, first, the AEs are introduced, then different attack models are presented. Finally, some detection methods are described.

A. Adversarial Examples

Given an image space $\xi = [0,1]^{H\times W\times C}$, a target classification model $f(\cdot)$ and a legitimate input image $x \in \xi$. An adversarial example is a perturbed image $x' \in \xi$ such that $f(x') \neq f(x)$ and $d(x, x') \leq \epsilon$, where $\epsilon \geq 0$. d is a distance metric to quantify the similarity between the perturbed and clean unperturbed inputs [21]. In the literature, three metrics are commonly used for generating AEs, and all three are L_p norms, including L_0 distance, the Euclidean distance (L_2) and the *Chebyshev* distance (L_∞ norm) [9].

In addition, the adversary attacks can be divided into two categories: (1) *white-box attacks* that have a full access to both the defense strategy and the target model's architecture and parameters, (2) *black-box attacks* that have no access to the model's architecture and parameters. The attacker only knows the output of the model (label or confidence score) for a given input.

B. Adversarial Attacks

In the following, we describe four prominent attacks that we considered in the evaluation of our detector, for a complete description of the state-of-the-art attacks, the reader is refereed to the following review paper [22].

1) Fast Gradient Sign Method: Goodfellow *et al.* [3] introduced a fast attack method called Fast Gradient Sign Method (FGSM). The FGSM performs only one step gradient update along the direction of the sign of gradient at each pixel as follows

$$x' = x + \epsilon\, sign(\nabla_x J_\theta(x, y)), \tag{1}$$

where θ is the set of model's parameters and $\nabla J(\cdot)$ computes the gradient of the loss function J around the current value of θ w.r.t. x. The sign$(\cdot)$ denotes the sign function and ϵ is a small scalar value that controls the perturbation magnitude.

2) Projected Gradient Descent: The Projected Gradient Descent (PGD) has been introduced by Madry *et al.* in [23]. The authors formulated the generation of an adversarial example as a constrained optimisation problem. Specifically, they introduced the following saddle point optimization problem

$$\begin{aligned} &\min_\theta \rho(\theta), \\ \text{with}\ \ &\rho(\theta) = \mathbf{E}_{(x,y)\sim D}[\max_{\delta\in S} J_\theta(x+\delta, y)], \end{aligned} \tag{2}$$

where $\mathbf{E}$ is a risk function and δ is the magnitude of the perturbation.

This classic saddle point problem is a composition of an inner maximization problem and an outer minimization problem. The inner maximization is the same as attacking a neural network by finding an adversarial example. On the other hand, the outer minimization aims to minimize the adversarial loss.

3) DeepFool: Moosavi-Dezfooli *et al.* [24] proposed the DeepFool attack that searches for the minimal perturbation that can change classification labels. This is done using an iterative procedure to get a linear approximation of the decision boundary of the classifier.

4) Carlini & Wagner: Carlini and Wagner [9] introduced three attacks under three different distance metrics: L_0, L_2 and L_∞. The C&W attack aims to minimize a trade-off between the perturbation intensity $||\delta||_p$ and the objective function $g(x')$, with $x' = x + \delta$ and $g(x') \leq 0$ if and only if $f(x') = c$ and $f(x) \neq c$

$$\begin{aligned} &\min_\delta ||\delta||_p + \lambda\, g(x'), \\ &\text{such that}\ \ x' \in [0,1]^n, \end{aligned} \tag{3}$$

where c is the target class and $\lambda > 0$ is a constant calculated empirically through binary search.

C. Detecting Adversarial Examples

Different detection methods have been proposed in the literature in order to distinguish the clean images from adversarial ones. The state-of-the-art methods for detecting AEs can be divided into three categories [12]: (1) using hand-crafted statistical features, (2) training a separate detector using adversarial samples, and finally (3) those exploiting prediction inconsistency.

For instance, Grosse *et al.* [17] applied statistical hypothesis testing to detect AEs. Under the assumption that the distribution of AEs statistically diverges from the training distribution, they used maximum mean discrepancy and energy distance as statistical distance measures to distinguish adversarial distributions from legitimate ones. In addition, the authors introduced an extra class in the model, in which the mode is trained to classify all AEs. However, this method requires a sufficiently large set of samples including both legitimate and adversarial, making it unusable to identify individual adversarial example. In addition, it has been shown in [11], that this method fails against black-box attacks. In order to identify adversarial subspaces, Feinman *et al.* [18] proposed to use Bayesian neural network uncertainty, available in dropout neural networks, and kernel density estimation in the feature space of the last hidden layer. These uncertainty and density estimate features are used as inputs to a logistic regression model. Similarly, Ma *et al.* [19] proposed to use local intrinsic dimensionality (LID) for characterizing the dimensional properties of adversarial regions. The authors empirically showed that LID of AEs is significantly higher than that of normal samples, and this difference is more pronounced in last layers of DNNs. Thus, they used the LID as features to train a detector to distinguish AEs. SafetyNet [20] quantized activations in late-stage ReLU, at some set of thresholds, to generate discrete codes. Next, a radial basis function (RBF)-support vector machine (SVM)-based classifier is used on these binary codes (activation patterns) to find AEs.

In a different way, other detection methods proposed to perform a preprocessing, usually denoiser, at each input. For

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

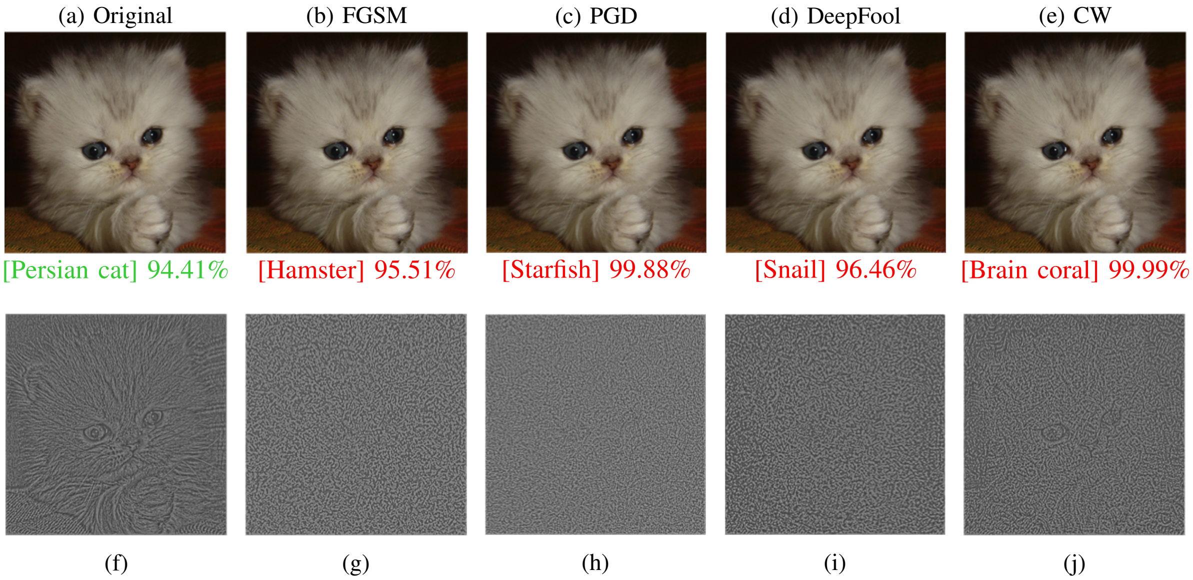

Fig. 1: Illustration of the relationship between natural scene statistics and adversarial perturbations. (a) the original image and (b)-(e) different attacked versions of it. The predicted class label and its corresponding probability are provided for each image. (f)-(j) the MSCN coefficients of the images shown in the top row.

instance, MagNet [10] consists of two components *detector* and *reformer*, where the detector learns a function that measures the distance between the input sample and the manifold. If this distance is greater than a threshold, then the detector rejects this input. Liao *et al.* [7] pointed out that these pixel guided denoiser methods, such as Magnet, are effective on small images but may not transfer well to large images. To fix this limitation, the authors proposed a high-level representation guided denoiser (HGD) for large images, which consists of using a loss function as the difference between top level outputs of the target model induced by original and AEs.

Xu *et al.* [12] proposed a detection approach called *feature squeezing* (FS). In this method, a DNN model's prediction on the original input with that on squeezed inputs are compared. If the difference between the predictions exceeds a threshold level, the input is identified to be adversarial. As feature squeezing methods, the authors reduced the color bit depth of each pixel and used spatial smoothing.

All of these described approaches showed some limitations [11], for instance they are effective against some specific attacks and lack generalization ability against different types of attacks. Also, they can achieve high accuracy but at the cost of increasing the false positive rate, thus rejecting considerable legitimate inputs, which is not efficient.

III. Proposed Approach

In this work, we propose three detection approaches capable of discriminating whether an input of DNN is an AE or not, and the samples detected as AEs are rejected. To distinguish between normal and attacked samples, our detection methods use natural scene statistics (NSS) [25], [26]. We assume that clean images possess certain regular statistical properties that are altered by any adversarial perturbation. Thus, by characterizing these deviations from the regularity of natural statistics using NSS, it is possible to determine whether the input is benign or malicious.

In order to extract scene statistics from input samples, we use the efficient spatial NSS model proposed in [27], referred to as mean subtracted contrast normalized (MSCN) coefficients. The MSCN coefficients of a given image I are defined by

$$\widehat{I}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + c} \tag{4}$$

where i and j are the pixel coordinates, and c is a tiny constant added only to avoid the division-by-zero. μ and σ are respectively local mean and variance maps defined by

$$\mu(i,j) = \sum_k \sum_l w(k,l) I(i+k, j+l) \tag{5}$$

$$\sigma(i,j) = \sqrt{\sum_k \sum_l w(k,l) \left[I(i+k,j+l) - \mu(i,j)\right]^2} \tag{6}$$

where $\mathbf{w} = \{w(k,l) | k = -3, ..., 3; l = -3, ..., 3\}$ is a 2D circularly-symmetric Gaussian weighting function.

To clearly demonstrate that MSCN coefficients are affected by adversarial perturbations, Figure 1 illustrates the MSCN coefficients of the original (clean) image and the different attacked versions of it. For the sake of space, in this illustration, we have considered only four adversarial attacks used in this work, namely Fast Gradient Sign Method (FGSM) [3], Projected Gradient Descent (PGD) [23], DeepFool [24] and Carlini & Wagner (CW) [9]. However, the result remains the same for the other attacks.

In Figure 1, first, according to the obtained class labels, it is clear that all the attacks have succeeded in fooling the DNN

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

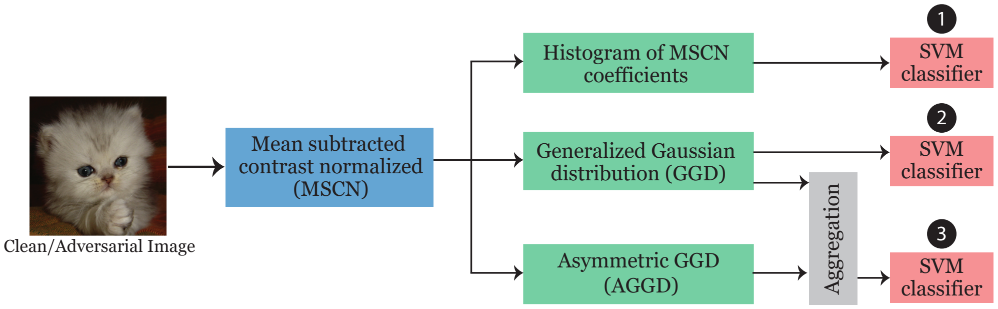
Clean/Adversarial Image
Mean subtracted contrast normalized (MSCN)
Histogram of MSCN coefficients
Generalized Gaussian distribution (GGD)
Asymmetric GGD (AGGD)
Aggregation
1
SVM classifier
2
SVM classifier
3
SVM classifier

Fig. 2: Flowchart of the proposed detection methods.

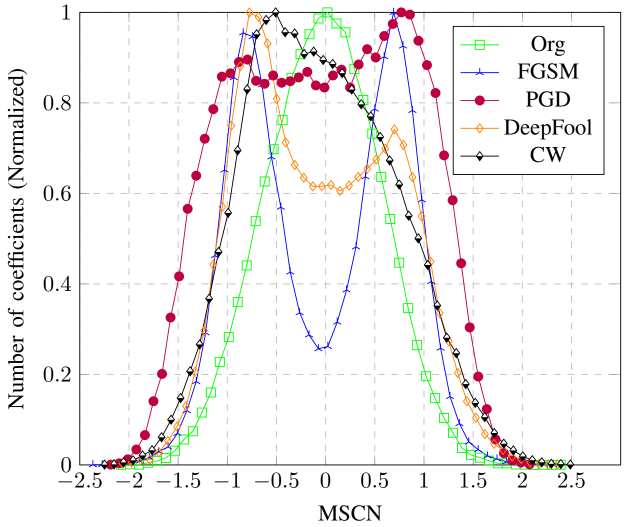

Fig. 3: Histogram of MSCN coefficients of the original image and the different attacked versions of it.

model with high confidence. While the attacked images, *i.e.,* AEs, are visually very close to the original image. Also, we can see that the MSCN coefficients of the original image differ significantly from those of AEs.

In addition, in order to show how the MSCN coefficients vary with the presence of adversarial perturbations, Figure 3 plots the histogram of MSCN coefficients of images shown in Figure 1 (top row). The original image exhibits a Gaussian-like MSCN distribution, while the same does not hold for the AEs which produce distributions with notable differences. It is evident that the MSCN coefficient distributions are affected by adversarial attacks. Thus, capturing these changes will allow the detection AEs.

Since the MSCN coefficients of clean samples can be easily differentiated from those of the AEs, we built a separate binary classifier using MSCN coefficients as input features. This is achieved using three different ways, as described below:

1) Histogram of MSCN coefficients: As illustrated in Figure 2, given an input image, in the first step, we extract the statistical features, *i.e.,* MSCN coefficients. Then, we sample the histogram of these coefficients between –2 to 2 with an interval of $\frac{4}{n-1}$ [28]. The range of $[-2, 2]$ is used, because, for the most images, the values of MSCN coefficients outside this range are so rare as to be negligible [27]. Finally, based on the obtained n-dimensional vector, the classifier will predict whether the input is an AE or not. n has been fixed to 81 and we have chosen the support vector machine (SVM) with Sigmoid kernel as classifier. Because Sigmoid kernel is suitable for binary classification problems.

2) Generalized Gaussian Distribution (GGD): As mentioned previously, in contrast to the clean image, the MSCN coefficient distributions of AEs are non-Gaussian distributions. Consequently, as a second approach, we propose to model these statistical distributions using the generalized Gaussian distribution (GGD). The GGD function is defined as follow:

$$f(x;\alpha,\sigma^2) = \frac{\alpha}{2\beta\Gamma(1/\alpha)} e^{-\left(\frac{|x|}{\beta}\right)^{\alpha}} \tag{7}$$

where

$$\beta = \sigma\sqrt{\frac{\Gamma\left(\frac{1}{\alpha}\right)}{\Gamma\left(\frac{3}{\alpha}\right)}} \tag{8}$$

and $\Gamma(\cdot)$ is the gamma function : $\Gamma(a) = \int\limits_0^{\infty} t^{a-1}e^{-t}dt\, a > 0.$

α and σ^2 are the shape-parameter and variance of the distribution, respectively. Due to the symmetry caused by the MSCN coefficients, the parameters of the distribution $\left(\alpha, \sigma^2\right)$ are estimated using the moment-matching method proposed in [29]. Therefore, instead of using 81 features as an input to our SVM classifier, in this second approach, each sample is represented with these two parameters.

3) AGGD & GGD: Finally, as a third approach, inspired by [27], we also consider the relationships between adjacent coefficients that provide information about the structure of the image. This structure is regular for a clean image, while it is altered for the case of the AE. Thus, in order to capture that, we use the pairwise products of neighboring MSCN coefficients along four directions (1) horizontal H, (2) vertical V, (3) main-diagonal $D1$ and (4) secondary-diagonal $D2$,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

which are defined as follow:

$$H(i,j) = \hat{I}(i,j)\hat{I}(i,j+1) \tag{9}$$

$$V(i,j) = \hat{I}(i,j)\hat{I}(i+1,j) \tag{10}$$

$$D1(i,j) = \hat{I}(i,j)\hat{I}(i+1,j+1) \tag{11}$$

$$D2(i,j) = \hat{I}(i,j)\hat{I}(i+1,j-1) \tag{12}$$

where i and j are the pixel coordinates.

It is clear that these pairwise products lead to an asymmetric distribution, so instead of using GGD, we apply the asymmetric generalized Gaussian distribution (AGGD), which is defined as follow:

$$f(x;\nu,\sigma_l^2,\sigma_r^2) = \begin{cases} \frac{\nu}{(\beta_l+\beta_r)\Gamma\left(\frac{1}{\nu}\right)} e^{\left(-\left(\frac{-x}{\beta_l}\right)^{\nu}\right)} & x < 0 \\ \frac{\nu}{(\beta_l+\beta_r)\Gamma\left(\frac{1}{\nu}\right)} e^{\left(-\left(\frac{x}{\beta_r}\right)^{\nu}\right)} & x \geq 0 \end{cases} \tag{13}$$

where

$$\beta_{side} = \sigma_{side}\sqrt{\frac{\Gamma\left(\frac{1}{\nu}\right)}{\Gamma\left(\frac{3}{\nu}\right)}} \tag{14}$$

where $side$ can be either r or l, ν represents the shape-parameter and σ_{side}^2 express the left or the right variance parameters. To estimate these parameters $(\nu, {\sigma_l}^2, {\sigma_r}^2)$, we use the method described in [30]. Another parameter that is not mentioned in the previous formula is the mean, which is defined as follow:

$$\eta = (\beta_r - \beta_l)\frac{\Gamma\left(\frac{2}{\nu}\right)}{\Gamma\left(\frac{1}{\nu}\right)} \tag{15}$$

after fitting the AGGD parameters, we get 4 features $(\eta, \nu, \sigma_l^2, \sigma_r^2)$ for each of the four pairwise products. By concatenating the GGD parameters with those of AGGD, we finally obtain 18 features per image denoted by:

$$\begin{aligned} f = [&\alpha, \sigma^2, \eta_H, \nu_H, {\sigma_{l_H}}^2, {\sigma_{r_H}}^2, \eta_V, \nu_V, {\sigma_{l_V}}^2, {\sigma_{r_V}}^2, \\ &\eta_{D1}, \nu_{D1}, {\sigma_{l_{D1}}}^2, {\sigma_{r_{D1}}}^2, \eta_{D2}, \nu_{D2}, {\sigma_{l_{D2}}}^2, {\sigma_{r_{D2}}}^2] \end{aligned}$$

IV. Experimental Results

We evaluated the three proposed AEs detection methods on three standard datasets: MNIST [31], CIFAR-10 [32] and ImageNet [33]. We built our own DNN classifiers for MNIST and CIFAR-10, for which we obtained accuracies of 99.39% and 89.87%, respectively. For ImageNet dataset, we used the pre-trained model inception-v3 [34].

The proposed method based on histogram of MSCN coefficients is referred as *our method 1*, while those based on GGD and GGD & AGGD are designated by *our method 2* and *our method 3*, respectively. In order to evaluate the efficiency of these proposed detection methods, we tested it against four different attacks, including Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), DeepFool and Carlini & Wagner (CW). Except for DeepFool and CW, for which we used the implementations of their authors [9], [24], we implemented the rest of these attacks by using the open-source software CleverHans library [35] based on TensorFlow.

In the training stage of the three proposed methods, we used a mixture of clean and attacked samples. For MNIST, we have chosen 1000 clean samples and applied on them the PGD attack, since it has been shown that the adversarial training with PGD attack tends to generalize well across a wide range of attacks [23]. The perturbed samples of the training set were generated using only the PGD attack, with a perturbation magnitude ϵ ranging between 0.03 and 0.7. It is important to note that the PGD attack has not been considered in our test phase.

We used the 1000 clean images with its corresponding 1000 AEs to train the SVM detector. We performed the same process for CIFAR-10 dataset, while for ImageNet, we used the training data provided by the NIPS challenge 2017.

For the test stage, we selected 1000 different samples from MNIST and CIFAR-10 test datasets, and 1000 test images from ImageNet's NIPS challenge dataset. Therefore, for each attack and for each of the three datasets, we have 2000 samples, *i.e.,* 1000 clean images and their attacked version.

The performance of the proposed detection method have been evaluated in terms of detection accuracy and false positive rate (FP), and they have been compared to four state-of-the-art detection methods, namely BU+KD [18], LID [19], FS [12] and HGD [7]. Since BU+KD cannot be used on ImageNet, we substituted it with HGD method.

Table I reports the performance of our detection methods against the four considered attacks. For MNIST dataset, globally, all the methods provided high detection accuracy, except for BU+KD method against BIM attack, where this method achieved the lowest performance. It is clear that, for all attacks, our methods have achieved the best performance. In addition, in contrast to the other detection methods, the proposed methods achieved these good results without increasing the FP rate, for which the proposed methods obtained the lowest value, for instance the proposed method 3 obtained 1.9% FP rate.

The same conclusion can be drawn for CIFAR-10 dataset, nevertheless, the FS method obtained the worst results on this dataset, especially against FGSM and BIM attacks, and achieved high result for CW only. The proposed methods obtained the highest results with always the lowest FP rate.

Finally, for ImageNet dataset, the proposed methods performed better than the other detectors and achieved the lowest FP rate and the highest detection accuracies, except against CW attack, for which the FS method obtained the best result. However, FS method provided the lowest accuracies for the remaining attacks. While HGD and LID methods provided a stable and somewhat acceptable result. It is important to note that the FP rate obtained by the proposed methods are very low compared to the others detection methods.

According to the obtained results, it is clear that the proposed approaches outperform the state-of-the-art detectors for most attacks, while achieved the lowest FP rate values. These low FP values are mainly due to the fact that the clean images

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE I: Comparison of the three proposed methods with state-of-the-art detectors in terms of detection accuracy and false positive rate (FP).

Detector	**Dataset**	**FGSM**	**BIM**	**DeepFool**	**CW**	**FP**
LID [19]	MNIST	97%	96%	92%	91%	4.4%
BU+KD [18]		91%	82%	-	98%	-
FS [12]		**100%**	99%	-	**100%**	4.0%
Our method 1		**100%**	**100%**	**100%**	**100%**	**2%**
Our method 2		**100%**	**100%**	**100%**	**100%**	**2%**
Our method 3		**100%**	**100%**	**100%**	**100%**	**1.9%**
LID [19]	CIFAR-10	94%	94%	84%	88%	5.6%
BU+KD [18]		72%	**100%**	-	92%	-
FS [12]		27%	52%	80%	**100%**	4.9%
Our method 1		**100%**	**100%**	**88%**	**100%**	**2.3%**
Our method 2		**100%**	**100%**	**91%**	**100%**	**2.1%**
Our method 3		**100%**	**100%**	**97%**	**100%**	**2%**
LID [19]	ImageNet	82%	78%	83%	80%	14.5%
HGD [7]		97%	95%	83%	85%	9.7%
FS [12]		44%	59%	80%	**100%**	8.3%
Our method 1		**100%**	**100%**	**87.1%**	84%	**6.2%**
Our method 2		**100%**	**100%**	**90%**	85%	**3.9%**
Our method 3		**100%**	**100%**	**91%**	91%	**3.6%**

constantly yield a Gaussian-like distribution, making them easy to discriminate by the NSS-based classifier. In addition, in contrast to the FS method that is effective only against CW attack, our methods generalize well against different attack models, without requiring training on them. Also, the proposed methods provide good results against all the attacks and across the three databases, thus showing high efficiency, especially for method 3, which are based on AGGD & GGD.

V. Conclusion

This paper presented three detection methods of AEs. Based on the observation that the natural scene statistics are altered by the adversarial perturbations, we developed NSS-based methods to detect these adversarial perturbations. The AEs can be easily distinguished from those of normal samples using MSCN coefficients as NSS tool. These MSCN coefficients are used as features by a binary classifier capable of classifying a given image as legitimate or adversarial.

The proposed detectors have been evaluated against four attacks and on three standard datasets. The experimental results demonstrated that the proposed detection approaches can achieve high detection accuracy, while maintaining a low value of FP rate. Thus, increasing the robustness of DNNs against adversarial attacks.

Even though the proposed detection methods are providing satisfactory results, we seek to improve them against the highly challenging CW attack on the ImageNet dataset. In addition, we believe that combining our method with an efficient defense approach can substantially increase the robustness of DNN.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[3] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[4] N. Papernot, P. McDaniel, X. Wu, S. Jha, , and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," *CoRR*, vol. abs/1511.04508, 2015. [Online]. Available: http://arxiv.org/abs/1511.04508

[5] S. Lee and J. Lee, "Defensive denoising methods against adversarial attack," 2018.

[6] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Transactions on Dependable and Secure Computing*, 2018.

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

[8] Y. Bakhti, S. A. Fezza, W. Hamidouche, and O. Déforges, "Ddsa: a defense against adversarial attacks using deep denoising sparse autoencoder," *IEEE Access*, vol. 7, pp. 160 397–160 407, 2019.

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

[10] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 135–147.

[11] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 3–14.

[12] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[13] D. Hendrycks and K. Gimpel, "Early methods for detecting adversarial images," *arXiv preprint arXiv:1608.00530*, 2016.

[14] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5764–5772.

[15] S. Ma, Y. Liu, G. Tao, W. C. Lee, and X. Zhang, "Nic: Detecting adversarial samples with neural network invariant checking." in *NDSS*, 2019.

[16] A. N. Bhagoji, D. Cullina, and P. Mittal, "Dimensionality reduction as a defense against evasion attacks on machine learning classifiers," *arXiv preprint arXiv:1704.02654*, 2017.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

[17] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, “On the (statistical) detection of adversarial examples,” *arXiv preprint arXiv:1702.06280*, 2017.

[18] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, “Detecting adversarial samples from artifacts,” *arXiv preprint arXiv:1703.00410*, 2017.

[19] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, J. Bailey *et al.*, “Characterizing adversarial subspaces using local intrinsic dimensionality,” *arXiv preprint arXiv:1801.02613*, 2018.

[20] J. Lu, T. Issaranon, and D. Forsyth, “Safetynet: Detecting and rejecting adversarial examples robustly,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 446–454.

[21] S. Fezza, Y. Bakhti, W. Hamidouche, and O. Deforges, “Perceptual evaluation of adversarial attacks for cnn-based image classification groundtruth adversarial examples,” in *Proc. IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, 2019.

[22] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” *IEEE transactions on neural networks and learning systems*, 2019.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *arXiv preprint arXiv:1706.06083*, 2017.

[24] Moosavi-Dezfooli, S. M., A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[25] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu, “On advances in statistical modeling of natural images,” *Journal of mathematical imaging and vision*, vol. 18, no. 1, pp. 17–33, 2003.

[26] D. L. Ruderman, “The statistics of natural images,” *Network: computation in neural systems*, vol. 5, no. 4, pp. 517–548, 1994.

[27] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-reference image quality assessment in the spatial domain,” *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[28] K. Gu, X. Xu, J. Qiao, Q. Jiang, W. Lin, and D. Thalmann, “Learning a unified blind image quality metric via on-line and off-line big training instances,” *IEEE Transactions on Big Data*, 2019.

[29] K. Sharifi and A. Leon-Garcia, “Estimation of shape parameter for generalized gaussian distributions in subband decompositions of video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 52–56, 1995.

[30] N. Lasmar, Y. Stitou, and Y. Berthoumieu, “Multiscale skewed heavy tailed model for texture analysis,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 2281–2284.

[31] Y. LeCun, C. Cortes, and C. Burges, “Mnist handwritten digit database,” *http://yann. lecun. com/exdb/mnist*, 2009.

[32] A. Krizhevsky, V. Nair, and G. Hinton, “The cifar-10 dataset,” *http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, A. C. Berg *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[35] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, and P. McDaniel, “cleverhans v1. 0.0: an adversarial machine learning library,” *arXiv preprint arXiv:1610.00768*, vol. 10, 2016.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE