NCIS: Neural Contextual Iterative Smoothing for Purifying Adversarial Perturbations

Sungmin Cha[1], Naeun Ko[3], Heewoong Choi[2], Youngjoon Yoo[3, 4], and Taesup Moon [2]
[1] New York University [2] ASRI / INMC / Seoul National University [3] NAVER Cloud [4] NAVER AI Lab
sungmin.cha@nyu.edu, naeun.ko@navercorp.com, chw0501@snu.ac.kr,
youngjoon.yoo@navercorp.com, tsmoon@snu.ac.kr

Abstract

We propose a novel and effective purification-based adversarial defense method against pre-processor blind white- and black-box attacks, without requiring any adversarial training or retraining of the classification model. Based on the observation of the adversarial noise, we propose a simple iterative Gaussian Smoothing (GS) that smoothes out adversarial noise and achieves substantially high robust accuracy. To further improve the method, we propose Neural Contextual Iterative Smoothing (NCIS), which trains a blind-spot network (BSN) in a self-supervised manner to reconstruct the discriminative features of the smoothed original image. From the extensive experiments on the large-scale ImageNet, we show that our method achieves both competitive standard accuracy and state-of-the-art robust accuracy against most strong purifier-blind white- and black-box attacks. Also, we propose a new evaluation benchmark based on commercial image classification APIs, including AWS, Azure, Clarifai, and Google, and demonstrate that users can use our method to increase the adversarial robustness of APIs.

1. Introduction

Despite the great success of deep learning-based image classification, it is well known that the classification models are vulnerable to the *adversarial attacks*, since the initiative reporting from [18]. Among various attempts to defend against such attacks, adversarial training (AT) [29] is regarded as one of the most robust defense methods. However, AT also possesses several limitations [4] as follows: (a) the computational cost is very expensive [38] particularly for the large-scale datasets like ImageNet [13], (b) it is difficult to guarantee the overall generalization capability as the model needs to be re-trained for every different task and adversary [29], and (c) significantly standard accuracy deteriorations has been reported even for a small perturbation budget for the attack [29] cases.

Regarding the above limitations, the input transformation methods [20, 27, 48], which attempt to remove the adversarial noise in the images before feeding them to classifiers, can be thought of as alternatives for AT since they do not require the re-training of the classifier. While such methods are widely regarded as broken by the strong [20] and sophisticated adaptive attacks, e.g., [2], they have been reconsidered recently due to their practical value, often under the term of *purification* of adversarial attacks [32, 41, 50].

In this paper, we also focus on the adversarial purification setting. In other words, we assume the adversary may (or may not) have full access to the classifier subject to attack, but has *no* access to the purification model. Such setting may seem relatively weak compared to other stronger attack scenarios, in which the adversary has additional access to the purification model or its output [3, 42]. However, we argue that such a pre-processor blind attack is what we may encounter the most in practice. For a more concrete argument, consider the situation in which a provider of large-scale image classification API would want to make the classifier "robust" to the potential adversarial attacks [36], while maintaining the standard accuracy as high as possible. In such a case, allowing the strong adaptive attack would be unrealistic since it basically means the entire API service system has been breached by the adversary to access the classifier and purifier, concerning the adversarial robustness of the classifier becomes a secondary issue. A more realistic scenario would be the one where the adversary aims to attack the classifier for the API (with/without the knowledge of the classifier) [28], and API providers devise additional guard, i.e., the purifier for defending the pre-trained classifier.

To this end, we devise a novel smoothing-based purification scheme that can significantly enhance the robustness of the classifier while maintaining the standard accuracy in the above-mentioned setting. Our contributions are threefold. First, we make a novel observation on the distribution of the adversarial noise that it is more or less zero mean and symmetric at the patch level. Second, from the above observation, we show that a very simple Gaussian Smoothing

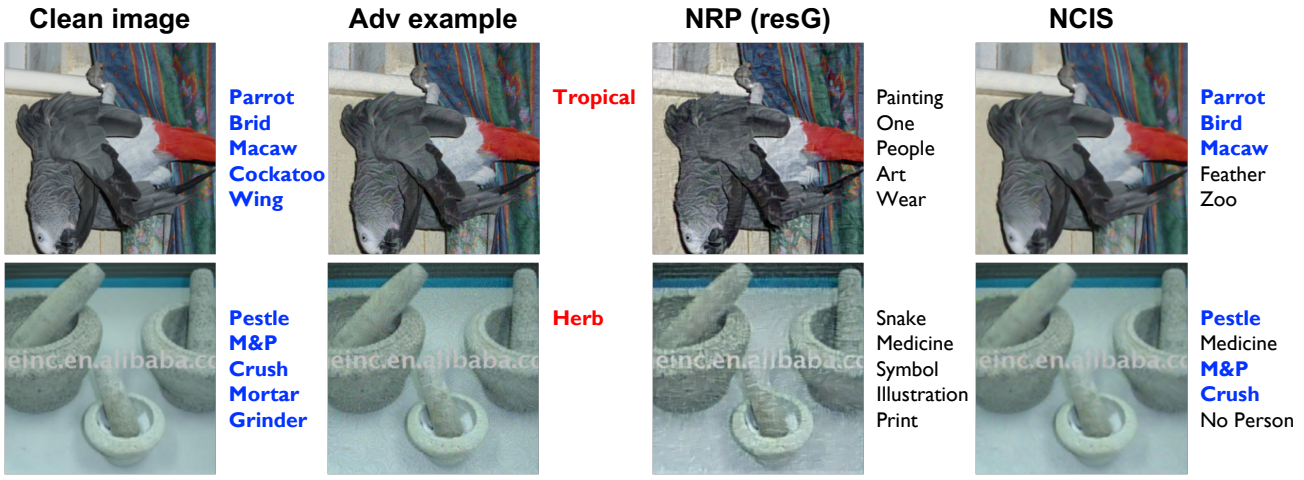


Figure 1. Visualization examples for defending the commercial vision API (Clarifai). The API predicts the correct top-5 predictions for the original clean images (first column), while it gets completely fooled by the adversarial examples (second column). The right two columns show the prediction results when two purifiers, NRP (resG) [32] and our NCIS, are applied to the adversarial examples, and we clearly observe the superior performance of NCIS.

(GS), which essentially employs a non-negative, symmetric convolution kernel, can achieve surprisingly high robust accuracy when it is iteratively applied as a purifier. Third, in order to compensate for the loss of the standard accuracy of the iterative GS, we employ a novel and efficient Blind-Spot Network (BSN), extended from a recent self-supervised learning-based denoiser [5], to reconstruct the discriminative features in the original image that are smoothed out by the iterative GS.

In order to validate our method, which is dubbed as Neural Contextual Iterative Smoothing (NCIS), we carry out an extensive evaluation of our method on the large-scale ImageNet [13] dataset for the various purifier-blind attack settings. More concretely, we follow the standard experimental protocol proposed in [15] and compare NCIS with other strong recent baselines for the following variations: four different classifier models, (pre-processor blind or full) white-box/black-box attacks, L_2/L_∞ targeted and untargeted attacks, and varying perturbation budget ϵ and attack iterations. Furthermore, to the best of our knowledge, we propose a new evaluation benchmark for the four commercial vision APIs, *i.e.*, Amazon AWS, Microsoft Azure, Google, and Clarifai, for the first time and evaluate the purifier performance for the strong transfer-based black-box attacks [28] (See Figure 1). Consequently, we demonstrate that our NCIS achieves very robust performance across all the tested attack settings and significantly outperforms the state-of-the-art purifier [32], with 14% fast inference time and $\times 14.7$ lower GPU memory requirement. Moreover, for pre-processor blind white-box L_∞ PGD attack, we show our NCIS even outperforms the strongest adversarial training (AT)-based defense FD [45], without any classifier re-training.

2. Related Work

Blind-spot networks (BSN) for image denoising Recent blind image denoising, training neural network-based denoiser only using noisy images, has made notable improvements. One of the main research directions of denoising is to use the blind-spot network (BSN). Several types of BSN are proposed for their own proposed method [5, 8, 9, 24, 26, 44]. Among them, FBI-Net [5] achieves the best denoising performance with a small-sized network add-on.

Adversarial attack White-box attacks generate adversarial examples based on the input gradient, having network information of target models. FGSM [18] generates adversarial examples with an optimization-based method by a single-step update. In follow-up studies, multi-step methods were proposed which strengthen the level of attacks by taking multiple gradient steps [12, 25, 29] in an iterative manner. Also, other attack approaches including new loss functions [7, 30], momentum-based iterative attack [16], and creating diverse input patterns [46] have been introduced. Black-box attack deals with the case an attacker cannot access the model gradient, and hence more difficult than white-box attack scenario. Transfer-based attacks [28] generate adversarial examples against a substitute model, and it is known that attacks are more successful when the substitute model and the target model architectures are similar [15]. Query-based attacks [1, 10, 22, 43] estimate the gradient through queries but require many queries for adversarial examples generation.

Adversarial defense Adversarial training (AT) methods [18, 29] train the classifier with adversarial examples by min-max optimization, and shows stable robustness against various adversarial attacks. However, they require excessive

training cost and have trade-off [15, 45] between the standard and robust accuracy. Certified defense methods [11, 36] aim to guarantee the model to be robust against adversarial perturbations with theoretical lower bounds on the robust accuracy. Input transformation processes the input images to achieve robustness against adversarial attacks [17, 20, 27, 48]. These methods are simple and cheap to apply but are easily broken by strong adversarial attacks [15]. Recently, purification methods that shift the adversarial examples back to the clean data representation have been actively proposed [32, 37, 39, 40, 50]. However, most of the methods conducted experiments only on small datasets (*e.g.*, MNIST and CIFAR-10/100). Among the methods, [32] is the only one that showed experimental results on ImageNet. Among these methods, Denoised Smoothing (DS) [36] proposed an approach that applies a pre-trained Gaussian denoiser for adversarial robustness. However, there exist several differences compared to our method, which will be discussed in the Supplementary Materials (S.M).

3. Motivation

3.1. Preliminary and Notations

Adversarial attack In a general image classification task, we denote the original input image by $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^D$, in which D is the number of pixels, and the ground-truth label of $\boldsymbol{x}$ by $\boldsymbol{y} \in \mathcal{Y} \subset \mathbb{R}^L$, in which L is the number of classes. A classification model parameterized by $\boldsymbol{\phi}$ is denoted by $g_{\boldsymbol{\phi}} : \mathcal{X} \to \mathcal{Y}$, and we assume $g_{\boldsymbol{\phi}}$ is pre-trained with external training data. Now, an adversary attempts to attack $g_{\boldsymbol{\phi}}$ by generating an adversarial example $\boldsymbol{x}'$ for $\boldsymbol{x}$ cause the misclassification of $\boldsymbol{x}$. The attack is called *untargeted* if $g_{\boldsymbol{\phi}}(\boldsymbol{x}') \neq \boldsymbol{y}$ and *targeted* if $g_{\boldsymbol{\phi}}(\boldsymbol{x}') = \boldsymbol{y}^*$ where $\boldsymbol{y}^*$ is a target class different from $\boldsymbol{y}$. The adversarial example $\boldsymbol{x}'$ is typically generated by solving an optimization problem, *e.g.*,

$$\boldsymbol{x}' = \underset{\boldsymbol{z}:\|\boldsymbol{z}-\boldsymbol{x}\|_p \leq \epsilon}{\operatorname{argmax}} \mathcal{L}(g_{\boldsymbol{\phi}}(\boldsymbol{z}), \boldsymbol{y}; \boldsymbol{\phi}), \tag{1}$$

for the untargeted attack, where $\mathcal{L}$ denotes a classification loss function, $\|\boldsymbol{z} - \boldsymbol{x}\|_p \leq \epsilon$ denotes the L_p-norm constraint of the optimization, and ϵ is the perturbation budget. Directly solving of (1) is intractable and various approximation methods have been proposed [7, 16, 18, 25, 29, 46]. We refer to *white-box* attack when the model architecture and weight of $g_{\boldsymbol{\phi}}$ are completely known, and *black-box* attack *vice versa*.

Purifier Purification is a pre-processing step for an input image before it is fed to a classification model. Let the goal of the purifier $T_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{X}$ is to reconstruct the purified image which allows $g_{\boldsymbol{\phi}}$ to predict an original prediction of $\boldsymbol{x}$ for a given input. Then, given the adversarial example $\boldsymbol{x}'$, the optimal purifier $T_{\boldsymbol{\theta}^*}$ will be

$$g_{\boldsymbol{\phi}}(T_{\boldsymbol{\theta}^*}(\boldsymbol{x}')) = \hat{\boldsymbol{y}}, \tag{2}$$

where $\hat{\boldsymbol{y}} = g_{\boldsymbol{\phi}}(\boldsymbol{x})$. However, note that the purifier randomly takes the adversarial example $\boldsymbol{x}'$ or the original image $\boldsymbol{x}$ as an input image during the general inference situation, hence it is difficult to be trained in a naive supervised manner.

Blind-spot network (BSN) The BSN is a special form of neural network that tries to reconstruct a pixel in the middle based on its surrounding context pixels. Namely, for a given input $\boldsymbol{x}$, the output of the BSN for pixel i is denoted by

$$f_{\boldsymbol{\theta},i}(\boldsymbol{x}) = f(\boldsymbol{\theta}, \boldsymbol{C}^{-i}_{k\times k}), \tag{3}$$

in which $\boldsymbol{C}^{-i}_{k\times k}$ is the $k \times k$ patch of $\boldsymbol{x}$ surrounding i, that does *not* include the pixel i. The BSN is typically used as a denoiser to estimate the clean $\boldsymbol{x}$ based on the noisy $\boldsymbol{z}$, of which the model parameter $\boldsymbol{\theta}$ is trained by minimizing $\|\boldsymbol{z} - f_{\boldsymbol{\theta}}(\boldsymbol{z})\|_2^2$, *i.e.*, in a self-supervised manner. In this paper, we utilize BSN as a smoothing function that is trained with the original *clean* $\boldsymbol{x}$ and show that it can work as an effective purifier that can both remove the adversarial noise and maintain the original discriminative feature of $\boldsymbol{x}$. For a particular BSN architecture, we extend the recent state-of-the-art network called FBI-Net in [5].

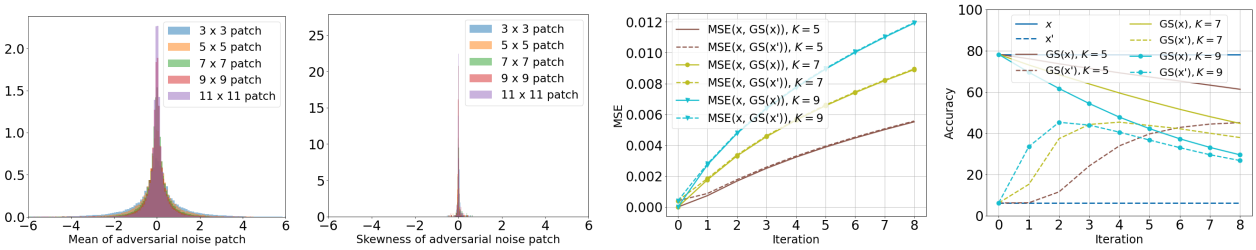
3.2. Analysis on Adversarial Noise

Suppose we have n clean images $\{\boldsymbol{x}_i\}_{i=1}^n$ and generated adversarial examples $\{\boldsymbol{x}'_i\}_{i=1}^n$ for each clean image by the process (1). Assuming that the adversarial noise is additive, we denote the i-th adversarial *noise* image as

$$\boldsymbol{N}'_i = \boldsymbol{x}'_i - \boldsymbol{x}_i, \;\; i = 1, \ldots, n, \tag{4}$$

and analyze the empirical distribution of $\boldsymbol{N}'_i$. More concretely, we randomly selected 1,000 images from the ImageNet training set and generated adversarial examples with the most generally used, untargeted L_∞ PGD white-box attack [29]. We collected 5,000 adversarial noise images in total by generating 5 attacked images for each $\boldsymbol{x}_i$ using the perturbation budgets $\epsilon = \{1/255, 2/255, 4/255, 8/255, 16/255\}$, respectively. Then, we randomly cropped 100 $K \times K$ patches from each noise image (thus, obtained 5×10^5 patches) and computed the empirical mean and skewness [19] of those patches.

Figure 2(a) shows the empirical mean of the adversarial noise in the patches (for varying patch sizes), and Figure 2(b) shows the empirical skewness. From the figures, we clearly observe that the adversarial noise in a patch is more or less zero-mean and has a symmetric distribution. This somewhat surprising regularity of adversarial noise, although generated from the complex iterative optimization process, motivates us to use a very simple Gaussian Smoothing (GS) based purifier in the next section. Note that additional experimental results for other types of attack are proposed in S.M.



(a) Dist. of mean. (b) Skewness. (c) MSE. (d) Accuracy.

Figure 2. Empirical analysis for adversarial examples generated from untargeted L_∞ PGD (α = 1.6 / 255, where α is a step size) attack with 10 attack iterations, and iterative Gaussian smoothing ($\sigma = (K-1)/6$ where K is the size of a kernel).

3.3. Iterative Gaussian Smoothing (GS)

Gaussian Smoothing (GS) is a widely used low-pass filter for image processing, which smoothes an image with a Gaussian kernel. The mechanism can be represented by the convolution operation between a Gaussian kernel $\mathbf{k}$ of size $K \times K$ and an input image. Namely, when an adversarial example $\boldsymbol{x}'$ is given as the input to the Gaussian smoothing function $\mathbf{G}(\cdot)$, we denote the GS process as

$$\mathbf{G}(\boldsymbol{x}') = \boldsymbol{x}' \otimes \mathbf{k}, \tag{5}$$

in which $\otimes$ is the convolution operation. Since $\mathbf{G}(\cdot)$ is a linear operation, and from (4), we have

$$\begin{aligned} \mathbf{G}(\boldsymbol{x}') &= \mathbf{G}(\boldsymbol{x} + \boldsymbol{N}') = \mathbf{G}(\boldsymbol{x}) + \mathbf{G}(\boldsymbol{N}') \\ &= \mathbf{G}(\boldsymbol{x}) + \boldsymbol{N}' \otimes \mathbf{k} \approx \mathbf{G}(\boldsymbol{x}). \end{aligned} \tag{6}$$

The last approximation follows from $\boldsymbol{N}' \otimes \mathbf{k} \approx 0$, which is from the observation in Figures 2(a) and 2(b) that the pixels in $K \times K$ patches in $\boldsymbol{N}'$ have symmetric, zero mean distribution and $\mathbf{k}$ is a non-negative, symmetric kernel. From (6), therefore, we can deduce that GS can mostly wash out the adversarial noise in $\boldsymbol{x}'$ and will make the smoothed version of $\boldsymbol{x}'$ become almost identical to that of $\boldsymbol{x}$.

Furthermore, by denoting $\mathbf{G}^i = \mathbf{G} \circ \mathbf{G}^{i-1}$ as applying GS i times iteratively, we can apply the similar logic as in equation (6) and deduce $\mathbf{G}^i(\boldsymbol{x}') \approx \mathbf{G}^i(\boldsymbol{x})$ as well, but with smaller approximation error. The reason is that the patches in $\mathbf{N}_i \otimes \mathbf{k}$ will still have zero mean, symmetric distributions, and the adversarial noise will keep getting washed out as we iteratively apply the Gaussian kernel $\mathbf{k}$. On the other hand, as GS continues, we can also expect that $\mathbf{G}^i(\boldsymbol{x}')$ and $\mathbf{G}^i(\boldsymbol{x})$ will get farther from $\boldsymbol{x}$, since GS will also wash out detailed and discriminative features in $\boldsymbol{x}$. Thus, as the iteration continues, we can conjecture that iterative GS may encounter a trade-off of increasing the robust accuracy, by removing adversarial noise, while hurting the standard accuracy, by also removing the discriminative features in $\boldsymbol{x}$.

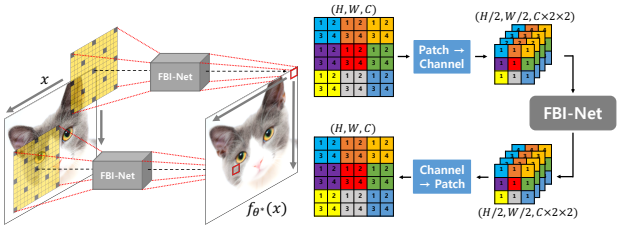
Figure 2(c) and 2(d) experimentally validate the above conjecture for the iterative GS. In the figures, for an ImageNet pre-trained ResNet-152 classifier, we tested with 1,000 clean images, $\boldsymbol{x}$, randomly subsampled from the ImageNet training dataset, and their adversarial examples, $\boldsymbol{x}'$, generated by L_∞ PGD ($\epsilon = 16/255$) attack. Figure 2(c) shows the MSE of $\|\boldsymbol{x} - \mathbf{G}^i(\boldsymbol{x})\|_2^2$ and $\|\boldsymbol{x} - \mathbf{G}^i(\boldsymbol{x}')\|_2^2$ as i increases, for different patch size K. From the figure, we can clearly observe that the two MSEs become almost identical, but increase (*i.e.*, both $\mathbf{G}^i(\boldsymbol{x})$ and $\mathbf{G}^i(\boldsymbol{x}')$ get farther from $\boldsymbol{x}$), as i increases, corroborating our above conjecture. Figure 2(d) reports both standard and robust accuracy of ResNet-152 when the iterative GS is used as a purifier for the same L_∞ PGD attack on the whole ImageNet validation set. We observe that the very simple iterative GS is surprisingly effective in purifying the adversarial noise as the robust accuracy of 44.92% is achieved when $K = 5$ and $i = 7$. However, we also note that the standard accuracy decreases as i gets larger due to the removed discriminative features. Motivated by this result, we propose our NCIS, which utilizes the iterative GS to maintain high robust accuracy but also employs BSN that can reconstruct the discriminative features washed out by GS to also achieve high standard accuracy.

4. Neural Contextual Iterative Smoothing

Motivated by the strong performance of iterative GS in the previous section, we propose a learnable neural network-based smoothing function as an iterative smoothing-based purifier. More concretely, we first present the improved performance of using self-supervised trained FBI-Net [5] for iterative smoothing, devise a more efficient version of FBI-Net dubbed as **FBI-E**, and present our **NCIS**, combining FBI-E with GS for a more stable and superior purifier.

Iterative smoothing with FBI-Net FBI-Net is a fully convolutional network that utilizes a special class of masked convolution filters as shown in Figure 3(a) such that the BSN condition (3) can be satisfied. Now, for given n clean images $\{\boldsymbol{x}_i\}_{i=1}^n$, we can train the network parameters, $\boldsymbol{\theta}$, of the FBI-Net by minimizing

$$\sum_{i=1}^{n} \|\boldsymbol{x}_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\|_2^2, \tag{7}$$



(a) Recons. process of FBI (b) Example of FBI-E for $m = 2$.

Figure 3. Illustration of FBI and FBI-E. (a): The black points denote masked pixels in a receptive field.

namely, in a self-supervised manner. Denoting the learned parameter by $\boldsymbol{\theta}^*$, we expect that $f_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ can be an effective smoothing-based purifier. The reasoning is, since the adversarial noise in $\boldsymbol{x}'$ is very small as shown in Figure 2(a), and it is generated independent of $f_{\boldsymbol{\theta}^*}$, the trained FBI-Net would have the similar outputs for both $\boldsymbol{x}$ and $\boldsymbol{x}'$, and they will be a smoothed reconstruction of $\boldsymbol{x}$. Consequently, we expect

$$\begin{aligned} &\|f_{\boldsymbol{\theta}^*}(\boldsymbol{x}') - \boldsymbol{x}\|_2^2 \approx \|f_{\boldsymbol{\theta}^*}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \\ &< \|\mathbf{G}(\boldsymbol{x}') - \boldsymbol{x}\|_2^2 \approx \|\mathbf{G}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2. \end{aligned} \tag{8}$$

Note the reasoning for (8) is possible since we are using BSN; when an ordinary fully convolutional network-based denoiser, *e.g.*, DnCNN [52], is used for $f_{\boldsymbol{\theta}}$ in (7), then it will end up learning an identity map, hence, $\|f_{\boldsymbol{\theta}^*}(\boldsymbol{x}') - \boldsymbol{x}'\|_2^2 \approx 0$. Thus, the adversarial example will be preserved.

Figure 5(a), which is for the same setting as Figure 2(c), experimentally verifies the above intuition for the iterative smoothing with the FBI-Net. Namely, denoting $f^i_{\boldsymbol{\theta}^*}(\boldsymbol{x}) = f_{\boldsymbol{\theta}^*} \circ f^{i-1}_{\boldsymbol{\theta}^*}(\boldsymbol{x})$, the figure shows $\|\boldsymbol{x} - f^i_{\boldsymbol{\theta}^*}(\boldsymbol{x})\|_2^2$ and $\|\boldsymbol{x} - f^i_{\boldsymbol{\theta}^*}(\boldsymbol{x}')\|_2^2$ as the iteration number i increases (the green solid and dashed line, respectively). Compared to the best iterative GS with $K = 5$ (brown solid and dashed lines), we observe that FBI-Net-based smoothing does a much better job than GS in reconstructing the original image $\boldsymbol{x}$ for both input cases ($\boldsymbol{x}$ and $\boldsymbol{x}'$), until $i \leq 4$. Moreover, in Figure 5(b), we show the Purification Success Rate (PSR) as,

$$\text{PSR}(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{g_\phi(\boldsymbol{x}) = g_\phi(\boldsymbol{z})\}.$$

in which $\boldsymbol{z}$ is the output of a purifier that takes either $\boldsymbol{x}$ or $\boldsymbol{x}'$ as input. The PSR is a metric for how much the PSR can recover the original classification result for $\boldsymbol{x}$. From the figure, we again observe that the iterative smoothing with FBI-Net does a much better job than iterative GS, in line with Figure 5(a), in preserving the original classification for $\boldsymbol{x}$ (green solid), until iteration $i \leq 6$. We also observe that PSR for $\boldsymbol{x}'$ (green dashed) increases as well until $i \leq 6$.

FBI-E: Extending FBI-Net While promising, we also observe that iterative smoothing with FBI-Net shows a few limitations. First, PSR for the given $\boldsymbol{x}'$ is still lower than iterative GS. Second, MSE and PSR diverge significantly, causing unstable results after a certain iteration. Finally, FBI-Net requires a large memory cost for the reconstruction and has a slow inference time to be used as a purifier.

To overcome these limitations and improve efficiency, we introduce two tensor operations for FBI-Net, *patch→channel* and *channel→patch*, to expand the reconstruction process from *context→pixel* to *context→patch*. Hence, the FBI-E mapping, $F_{\boldsymbol{\theta}}(\boldsymbol{x})$, can be denoted as

$$F_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathcal{O}_{C\to P}(f_{\boldsymbol{\theta}}(\mathcal{O}_{P\to C}(\boldsymbol{x}))), \tag{9}$$

in which $\mathcal{O}_{P\to C}(\cdot)$ and $\mathcal{O}_{C\to P}(\cdot)$ denotes *patch→channel* and *channel→patch* operation, respectively. Figure 3(b) illustrates the two operations being applied as a pre- and post-operation for FBI-Net; $\mathcal{O}_{P\to C}(\cdot)$ transfers pixels of each $m \times m$ patch in an input image (color-coded) to channel-wise pixels and $\mathcal{O}_{C\to P}(\cdot)$ exactly does the reverse operation. By these operations, FBI-E is also a BSN, but the reconstruction is now done in a patch level based on the context around the patch, and we expect (8) would also hold for $F_{\boldsymbol{\theta}}$.

The advantage of the two proposed operations is quite huge since it reduces the spatial resolution of both the input image and all feature maps of FBI-Net by $m \times m$ times. As shown in Figure 5(a) and 5(b), we experimentally checked that FBI-E with $m = 2$ achieves superior MSE and PSR for both input cases (orange solid and dashed lines) than GS and FBI-Net. Furthermore, the inference time and memory improvement of FBI-E over FBI-Net is given in the ablation study given in the later section.

Neural contextual iterative smoothing (NCIS) After applying the proposed two operations, the level of difficulty for the reconstruction increases since FBI-E has to reconstruct the entire patch of size $m \times m$, not just a single pixel. To address this problem, we propose to combine FBI-Net with GS and propose a new smoothing function as

$$\mathcal{F}_{\boldsymbol{\theta},\text{NCIS}}(\boldsymbol{x}) = F_{\boldsymbol{\theta}}(\boldsymbol{x}) + \mathbf{G}(\boldsymbol{x}). \tag{10}$$

The network parameter $\boldsymbol{\theta}^*$ is obtained by minimizing (7), which results in $F_{\boldsymbol{\theta}^*}(\boldsymbol{x})$, learning to reconstruct the residual $\boldsymbol{x} - \mathbf{G}(\boldsymbol{x})$. The intuition is that, when an adversarial example $\boldsymbol{x}'$ is given as input, we expect $\mathbf{G}(\boldsymbol{x}')$ to wash out the adversarial noise such that $\mathbf{G}(\boldsymbol{x}') \approx \mathbf{G}(\boldsymbol{x})$, and then we let $F_{\boldsymbol{\theta}^*}(\boldsymbol{x}')$ to reconstruct the discriminative features of the original $\boldsymbol{x}$ that is also smoothed out by $\mathbf{G}(\boldsymbol{x})$. Hence, we expect $F_{\boldsymbol{\theta}^*}(\boldsymbol{x}') \approx F_{\boldsymbol{\theta}^*}(\boldsymbol{x}) \approx \boldsymbol{x} - \mathbf{G}(\boldsymbol{x})$. Finally, our proposed Neural Contextual Iterative Smoothing (NCIS) is obtained by iteratively applying $\mathcal{F}_{\boldsymbol{\theta}^*,\text{NCIS}}(\boldsymbol{x})$ denoted as $\mathcal{F}^i_{\boldsymbol{\theta}^*,\text{NCIS}}(\boldsymbol{x}) = \mathcal{F}_{\boldsymbol{\theta}^*,\text{NCIS}} \circ \mathcal{F}^{i-1}_{\boldsymbol{\theta}^*,\text{NCIS}}(\boldsymbol{x})$. The overall procedure of our NCIS is depicted in Figure 4.

In Figure 5(a) and 5(b), we verify the promising results of our NCIS. First, NCIS ($m = 2$ for FBI-E and $K = 11$ for GS) achieves the lowest MSE for both input cases until the $i = 7$ compared to other methods, which shows that

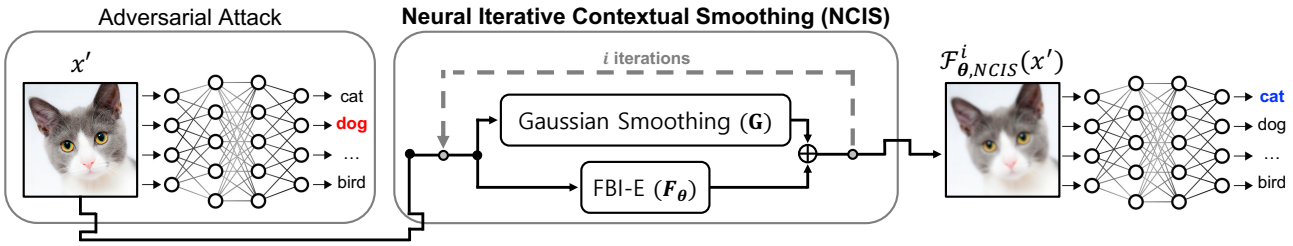
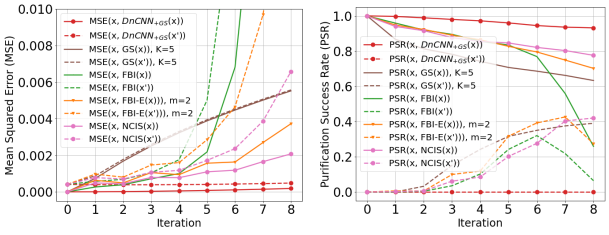


Figure 4. Overall procedure of NCIS for adversarial purification.



(a) MSE for original $\boldsymbol{x}$ (b) Purification Success Rate

Figure 5. Empirical verification using randomly selected 1,000 images from ImageNet training dataset and adversarial examples of generated from untargeted L_∞ PGD ($\epsilon = 16/255$, $\alpha = 1.6/255$) attack with ten attack iterations.

NCIS can successfully estimate the original image $\boldsymbol{x}$ for both inputs ($\boldsymbol{x}$ and $\boldsymbol{x}'$). Second, consequently, NCIS with seven iterations attains the highest PSR for $\boldsymbol{x}$ maintaining the competitive PSNR for $\boldsymbol{x}'$ compared to FBI-E. The results show that iterative GS and NCIS, which are trained based solely on self-supervised learning using only original images (*i.e.*, without any adversarial training), have a potential to become a powerful purifier for adversarial defense. Finally, we add the DnCNN + GS ($K = 11$), using DnCNN in place of $F_{\boldsymbol{\theta}}(\boldsymbol{x})$ in (4), and we observe that the general CNN-based denoiser model cannot be used as the smoothing function for the iterative smoothing since it almost perfectly reconstructs the input image, even the adversarial noise in the input.

5. Experimental Results

5.1. Experimental settings

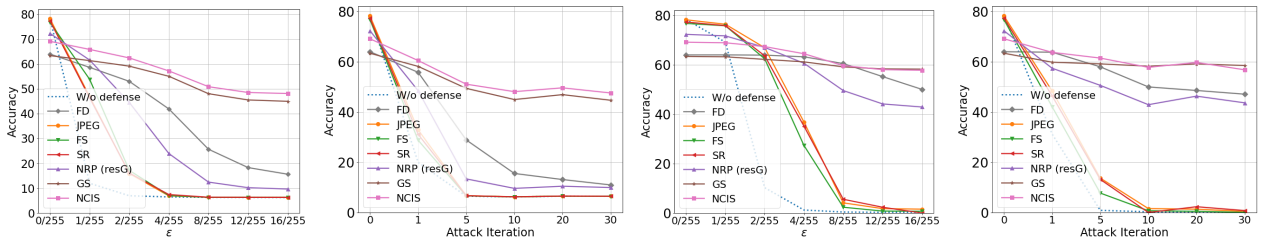
In this section, we validate our NCIS against various types of attacks on the ImageNet validation dataset. We designed our experimental setting following [6, 15] for rigorous verification of our proposed method. Also, we used attacks and defenses implemented on public packages [14, 15, 23, 33].

Adversarial attack For using *white-box* attack methods, we mainly consider *pre-processor blind* white-box attack, which has access only to classification model weights. We selected four ImageNet pre-trained models, ResNet-152 [21], WideResNet-101 [51], ResNeXT-101 [47] and RegNet-32G [34], and then we generated adversarial examples of the ImageNet validation dataset using four *untargeted* gradient-based iterative attacks, PGD [29], CW [7], MIFGSM [16], DIFGSM [46] and AutoAttack [12]. For *black-box* attack, we experimented with *transfer-based* attacks which still achieve a strong and efficient attack success rate compared to other types of black-box attacks. For transfer-based attack, we generated adversarial examples by attacking a substitute model with L_∞ PGD attack, and the substitute models are listed in the S.M.

Adversarial defense We selected four *input transformation-based* defense methods abbreviated as FS (depth= 4) [49], JPEG (quality= 90) [17], TVM ($p = 0.3, \lambda = 0.5$, max iteration= 10) [20, 35] and SR ($\sigma = 0.04$) [31]. Moreover, we implemented the current state-of-the-art *purification-based* method, NRP and NRP(resG) [32], from their official code. NRP (resG) is the lightweight version of NRP. Lastly, we added FD [45], one of the current state-of-the-art *adversarial training* method, by implementing the code and weight for ResNet-152 proposed in [15]. Additionally, we had considered adding recently proposed purification methods such as SOAP [40] and ADP [50], but neither of them publicized their code nor experimented on ImageNet. A more detailed comparison of settings with other purifier-based methods is proposed in the S.M. For Gaussian smoothing (GS) used in all experiments including GS in NCIS, we set $\sigma = (K-1)/6$ by following the fact that the length for 99 percentile of Gaussian distribution is 6σ. If there are no additional notations, we set K for GS to 5 and only used NCIS consisting of FBI-E ($m = 2$) and GS ($K = 11$) for all experiments. We conducted experiments using NCIS trained by three different seeds and report the average result. The detailed description of the experimental settings, the architecture of FBI-Net, and hyperparameters are in the S.M.

5.2. Experimental results for pre-processor blind white-box attacks

Experiments with various white-box attacks Table 1 shows the experimental results of each defense method against five attacks for four different ImageNet pre-trained classification models. First, as for standard accuracy, JPEG achieved the best results. However, as already presented in [15, 20], traditional input transformation-based methods are easily broken by white-box attacks. Second, GS achieves superior robust accuracy in most cases and even outperforms NRP and NRP (resG). One possible reason NRP and NRP (resG) show deteriorated performance compared to its original paper is that the proposed loss function for training NRP and NRP (resG) is not well generalized to purify adversarial



(a) Untargeted (various ϵ) (b) Untargeted (various iters) (c) Targeted (various ϵ) (d) Targeted (various iters)

Figure 6. Experimental results against L_∞ white-box PGD attacks with ResNet-152. For the experiments on various ϵ, we set $\alpha = 1.6/255$, where α is a step size, with ten attack iterations. In the case of experiments for attack iterations, we set $\epsilon = 16/255$, and $\alpha = 1.6/255$ if the iteration is lower than 10, and used $\alpha = 1/255$ otherwise. Note that FD denotes the model trained by adversarial training proposed in [45].

Table 1. Experimental results of *untargeted* white-box adversarial attack. For L_∞ attacks, we set $\epsilon = 16/255$, $\alpha = 1.6/255$ and attack iterations = 10. For L_2 PGD attack, we used $\epsilon = 5$ and $\alpha = 0.1$. For L_2 CW attack, other than setting attack iterations as 10, we applied default hyperparameters proposed in [23]. **Boldface** denotes our proposed methods, and red and blue denotes the highest and second highest results respectively. For ResNet-152, WideResNet-101 and ResNeXT-101, we set $i = 7$ for GS and NCIS.

Model / Defense		Standard Accuracy	CW (L_2)	MIFGSM (L_∞)	DIFGSM (L_∞)	PGD (L_∞)	PGD (L_2)
ResNet-152	W/o defense	78.25	9.37	6.34	0.43	6.20	10.66
	JPEG	78.13	26.78	6.36	0.61	6.25	29.01
	FS	76.66	46.37	6.35	0.46	6.22	41.08
	TVM	69.84	59.37	9.32	5.02	17.41	59.15
	SR	77.24	40.87	6.36	0.05	6.22	31.06
	NRP (resG)	72.24	58.05	16.39	2.40	9.60	55.84
	NRP	74.04	58.16	12.35	2.59	10.71	55.58
	GS	**63.32**	**60.36**	**24.28**	**22.28**	**44.92**	**60.65**
	NCIS	**68.93**	**64.32**	**39.05**	**33.29**	**48.06**	**64.28**
WideResNet-101	W/o defense	78.91	9.42	6.96	0.31	6.60	11.97
	JPEG	78.22	41.55	6.95	0.10	6.91	33.19
	FS	77.03	48.82	6.95	0.10	6.91	46.41
	TVM	69.17	69.18	12.15	7.56	20.33	59.84
	SR	77.85	40.24	6.94	0.10	6.93	33.14
	NRP (resG)	72.45	58.96	19.66	4.34	12.18	58.17
	NRP	74.58	58.60	14.97	5.53	13.22	57.69
	GS	**60.33**	**57.88**	**28.15**	**25.30**	**45.04**	**57.96**
	NCIS	**68.54**	**63.96**	**34.18**	**33.98**	**49.26**	**64.03**
ResNeXT-101	W/o defense	79.21	9.43	8.59	0.61	7.81	13.84
	JPEG	78.28	43.23	8.46	0.65	7.96	36.77
	FS	77.54	47.88	8.61	0.62	7.94	46.14
	TVM	70.66	59.47	12.52	6.68	20.67	61.05
	SR	78.08	40.45	8.59	0.65	8.0	34.31
	NRP (resG)	73.65	59.04	20.57	4.14	13.00	59.53
	NRP	75.28	58.30	15.88	5.23	13.58	58.15
	GS	**63.78**	**60.77**	**28.01**	**23.50**	**46.43**	**61.18**
	NCIS	**70.08**	**65.12**	**36.47**	**35.53**	**51.46**	**65.57**

examples generated from various types of attack. On the other hand, because our GS is based on the findings of the characteristic of adversarial noise, it can be applied to purify varied types of adversarial examples. Furthermore, NCIS consistently surpasses the robust accuracy of GS and other baselines for all the architectures, including RegNet-32G reported in the S.M. Additionally, since we observed that there is little difference in performance between NRP and NRP (resG) and NRP is computationally expensive, we only use NRP (resG) for the remaining experiments.

In the S.M, we present additional results as follows: First, we report experimental results against AutoAttack [12], which is one of the most powerful attack methods, and the results are consistent with our previous findings. Second, in order to compare our method with Denoised Smoothing (DS) [36], we implemented both their official code and the pre-trained weights of a denoiser. We experimentally confirm that not only DS requires a significant amount of time for defense, but its robust accuracy against PGD attack is also lower than that of our method. Third, we compare the inference time, GPU memory usage, and number of parameters of each method. As a result, we confirm that NRP has slower inference time than NCIS due to its larger number of parameters and the increased GPU memory usage required for purification.

Experiments with PGD attack with various settings To evaluate the proposed method more rigorously, we experimented with both *targeted* and *untargeted* L_∞ PGD attack with various ϵ and the number of attack iterations.

Figure 6(a) and 6(b) show the experimental results for *untargeted* L_∞ PGD attack in ResNet-152 [21]. As ϵ and the number of attack iterations increase, the robust accuracy of baseline defense methods goes significantly down. Besides, FD has lower standard accuracy (63.96) compared to NCIS. Moreover, input transformation-based methods and NRP are crumbling like a wreck when the attack setting is strong. However, NCIS and GS achieve the highest robust accuracy, showing up to four times higher robust accuracy of FD. Besides, both the standard and robust accuracy of NCIS are higher than those of GS in all settings.

Also, Figure 6(c) and 6(d) show experiments on *targeted* L_∞ PGD attack. We observe that the tendency of experimental results has changed from the *untargeted* results. First, FD and NRP (resG) show better performance than the *untargeted* attack case. We presume that both FD and NRP (resG) have a generalization problem and fail to defend against all types of attacks stably. Second, the robust accuracy of FD and NRP degrade as the number of iterations and ϵ increase, resulting in lower robust accuracy than our methods against strong

Table 2. Experimental results for vision APIs. **Boldface**, red and blue each denotes the proposed, highest and second highest result. The numbers in the Table each denotes the standard and robust accuracy (in parenthesis) for each case. Vaniila case does not use any of defenses.

	Prediction Accuracy				Top-1 (Top-5) Accuracy			
	AWS	Azure	Clarifai	Google	AWS	Azure	Clarifai	Google
Vanilla	1.00/0.00	1.00/0.00	1.00/0.00	1.00/0.00	1.00(1.00)/0.00(0.00)	1.00(1.00)/0.00(0.00)	1.00(1.00)/0.00(0.00)	1.00(1.00)/0.00(0.00)
JPEG	0.78/0.12	0.82/0.08	0.78/0.58	0.75/0.10	0.75(0.91)/0.04(0.14)	0.72(0.94)/0.02(0.12)	0.79(0.95)/0.45(0.80)	0.60(0.84)/0.04(0.13)
FS	0.60/0.21	0.60/0.05	0.76/0.56	0.55/0.11	0.44(0.72)/0.08(0.23)	0.42(0.67)/0.06(0.10)	0.69(0.94)/0.45(0.77)	0.43(0.67)/0.07(0.14)
SR	0.78/0.19	0.81/0.12	0.85/0.62	0.69/0.13	0.72(0.89)/0.06(0.25)	0.75(0.95)/0.10(0.17)	0.87(0.99)/0.52(0.80)	0.46(0.77)/0.07(0.18)
NRP	0.45/0.19	0.47/0.16	0.45/0.19	0.39/0.14	0.30(0.57)/0.09(0.18)	0.34(0.54)/0.12(0.18)	0.30(0.57)/0.09(0.18)	0.26(0.47)/0.09(0.16)
GS	**0.68/0.26**	**0.63/0.23**	**0.79/0.65**	**0.55/0.21**	**0.45(0.80)/0.11(0.25)**	**0.53(0.78)/0.20(0.30)**	**0.72(0.99)/0.55(0.86)**	**0.42(0.69)/0.11(0.23)**
NCIS	**0.74/0.28**	**0.80/0.21**	**0.86/0.62**	**0.72/0.27**	**0.60(0.88)/0.13(0.31)**	**0.78(0.95)/0.13(0.29)**	**0.76(0.98)/0.55(0.80)**	**0.54(0.83)/0.20(0.37)**

attacks. On top of this, NCIS and GS achieve superior robust accuracy against strong attacks. See S.M for the results on other classifiers (*e.g.*, WideResNet-101, ResNeXT-101, and RegNet-32G), showing similar tendencies.

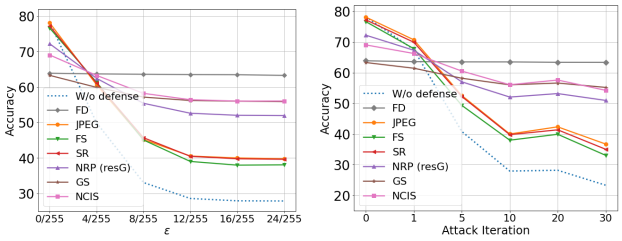
Although our paper focuses on pre-processor blind white-box attack for a classification model, we also conducted experiments on ***full*** and ***purifier-aware*** **white-box attacks** where attackers can access or are aware of the purifier. To counter both strong attacks, we propose a simple variant of NCIS, adding Gaussian noise in the procedure of iterative smoothing. Also, note that we conducted ablation studies to confirm the role of each proposed component. The details and experimental results are proposed in the S.M.

5.3. Experimental results for black-box attacks

Transfer-based black-box attack Figure 7 shows ResNet-152 results attacked by adversarial examples generated by attacking WideResNet-101 [51]. The notable discovery is that FD shows the almost constant but most robust result for all cases, similar to experimental results already proposed in [15]. Also, different from white-box attack cases, NRP (resG) achieves competitive performance compared to other input transformation-based methods, supporting the result of their paper. However, note that our GS and NCIS not only surpass the performance of NRP (resG) but also NCIS achieves competitive performance compared to FD when considering both standard and robust accuracy. Additionally, we conducted experiments with the state-of-the-art score-based black-box attack (Square [1]) and the experimental results on other classifiers, reported in the S.M.

Experiment with APIs [28] showed existing APIs for multi-label classification can be fooled by ensemble transfer-based black-box attacks. Based on this finding, we propose a new experiment for evaluating purification methods using APIs of Azure, AWS, Clarifai, and Google. From this experiment, we evaluate how well each purifier can restore the top five labels predicted from a clean image when given strong adversarial examples. To evaluate each purifier, we used three evaluation metrics: *Prediction Accuracy*, *Top-1 Accuracy*, and *Top-5 Accuracy*. The additional details of metrics and used hyperparameters are noted in the S.M.

Table 2 shows experimental results on the generated test dataset. First, we observe that traditional input transformation-based methods generally show high standard accuracy with competitive robust accuracy compared to NRP (resG). We believe that this result shows another example of insufficient generalization of NRP (resG) for purifying various types of adversarial examples. Second, NCIS and GS show the most uniformly competitive performance for various APIs. Among them, NCIS achieves better purification performance than GS, considering the average of the standard and robust accuracy.



(a) Experiments for various ϵ (b) Experiments for various iters

Figure 7. Experiments on transfer-based black-box attack with L_∞ PGD for ResNet-152.

6. Concluding Remarks

We proposed Neural Contextual Iterative Smoothing (NCIS) for adversarial purification against adversarial attack to a classifier. First, we proposed the novel observation that adversarial noise has almost zero mean and a symmetric distribution. Second, based on the above finding, we proposed the learnable neural network-based smoothing function, named as NCIS, for adversarial purification. From the extensive experiments, we observed NCIS robustly purifies adversarial examples generated from various types of white- and black-box attack without requiring re-training of the classification model.

7. Acknowledgment

This work was done while S. Cha did a research internship at NAVER AI Lab. All authors thank NSML team for the GPU support. The work was also supported in part by NRF grant [2021R1A2C2007884, 2021M3E5D2A01024795], IITP grant [No.2021-0-01343, No.2021-0-02068, No.2022-0-00113, No.2022-0-00959] funded by the Korean government, and SNU-Naver Hyperscale AI Center.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In European Conference on Computer Vision, pages 484–501. Springer, 2020. 2, 8

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International Conference on Machine Learning, pages 274–283. PMLR, 2018. 1

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International conference on machine learning, pages 274–283. PMLR, 2018. 1

[4] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. arXiv preprint arXiv:2102.01356, 2021. 1

[5] Jaeseok Byun, Sungmin Cha, and Taesup Moon. Fbi-denoiser: Fast blind image denoiser for poisson-gaussian noise. arXiv preprint arXiv:2105.10967, 2021. 2, 3, 4

[6] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705, 2019. 6

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pages 39–57. IEEE, 2017. 2, 3, 6

[8] Sungmin Cha and Taesup Moon. Neural adaptive image denoiser. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2981–2985, 2018. 2

[9] Sungmin Cha and Taesup Moon. Fully convolutional pixel adaptive image denoiser. In IEEE International Conference on Computer Vision (ICCV), pages 4160–4169, 2019. 2

[10] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM workshop on artificial intelligence and security, pages 15–26, 2017. 2

[11] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In International Conference on Machine Learning, pages 1310–1320. PMLR, 2019. 3

[12] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In International conference on machine learning, pages 2206–2216. PMLR, 2020. 2, 6, 7

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia 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. Ieee, 2009. 1, 2

[14] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. arXiv preprint arXiv:1902.07623, 2019. 6

[15] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 321–331, 2020. 2, 3, 6, 8

[16] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9185–9193, 2018. 2, 3, 6

[17] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. arXiv preprint arXiv:1608.00853, 2016. 3, 6

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014. 1, 2, 3

[19] Richard A Groeneveld and Glen Meeden. Measuring skewness and kurtosis. Journal of the Royal Statistical Society: Series D (The Statistician), 33(4):391–399, 1984. 3

[20] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117, 2017. 1, 3, 6

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 6, 7

[22] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In International Conference on Machine Learning, pages 2137–2146. PMLR, 2018. 2

[23] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. arXiv preprint arXiv:2010.01950, 2020. 6, 7

[24] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2129–2137, 2019. 2

[25] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016. 2, 3

[26] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In Advances in Neural Information Processing Systems (NIPS), pages 6968–6978, 2019. 2

[27] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1778–1787, 2018. 1, 3

[28] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770, 2016. 1, 2, 8

[29] 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. 1, 2, 3, 6

[30] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2574–2582, 2016. 2

[31] Aamir Mustafa, Salman H Khan, Munawar Hayat, Jianbing Shen, and Ling Shao. Image super-resolution as a defense against adversarial attacks. IEEE Transactions on Image Processing, 29:1711–1724, 2019. 6

[32] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 262–271, 2020. 1, 2, 3, 6

[33] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. CoRR, 1807.01069, 2018. 6

[34] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10428–10436, 2020. 6

[35] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena, 60(1-4):259–268, 1992. 6

[36] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. Advances in Neural Information Processing Systems, 33, 2020. 1, 3, 7

[37] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605, 2018. 3

[38] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! arXiv preprint arXiv:1904.12843, 2019. 1

[39] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. arXiv preprint arXiv:1707.05474, 2017. 3

[40] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervised learning. In International Conference on Learning Representations, 2020. 3, 6

[41] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervision. arXiv preprint arXiv:2101.09387, 2021. 1

[42] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. arXiv preprint arXiv:2002.08347, 2020. 1

[43] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In International Conference on Machine Learning, pages 5025–5034. PMLR, 2018. 2

[44] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In European Conference on Computer Vision (ECCV), pages 352–368, 2020. 2

[45] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 501–509, 2019. 2, 3, 6, 7

[46] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In Proceedings of the IEEE/CVF Conference

on Computer Vision and Pattern Recognition, pages 2730–2739, 2019. 2, 3, 6

[47] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1492–1500, 2017. 6

[48] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155, 2017. 1, 3

[49] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155, 2017. 6

[50] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. arXiv preprint arXiv:2106.06041, 2021. 1, 3, 6

[51] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016. 6, 8

[52] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE transactions on image processing, 26(7):3142–3155, 2017. 5