RAIN: Robust and Accurate Classification Networks with Randomization and Enhancement

Jiawei Du\textsuperscript{1,2\star}, Hanshu Yan\textsuperscript{1\star}, Joey Tianyi Zhou\textsuperscript{2}, Rick Siow Mong Goh\textsuperscript{2}, and Jiashi Feng\textsuperscript{1}

\textsuperscript{1} Dept. ECE, National University of Singapore, Singapore
\textsuperscript{2} Institute of High performance Computing, A*STAR, Singapore

Abstract. Along with the extensive applications of CNN models for classification, there has been a growing requirement for their robustness against adversarial examples. In recent years, many adversarial defense methods have been introduced, but most of them have to sacrifice classification accuracy on clean samples to achieve better robustness of CNNs. In this paper, we propose a novel framework to improve robustness and meanwhile retain accuracy of given classification CNN models, termed as RAIN, which consists of two conjugate modules: structured randomization (SRd) and detail generation (DG). Specifically, the SRd module randomly downsamples and shifts the input, which can destroy the structure of adversarial perturbations so as to improve the model robustness. However, such operations also incur accuracy drop inevitably. Through our empirical study, the resultant image of the SRd module suffers loss of high-frequency details that are crucial for model accuracy. To remedy the accuracy drop, RAIN couples a deep super-resolution model as the DG module for recovering rich details in the resultant image. We evaluate RAIN on STL10 and the ImageNet datasets, and experiment results well demonstrate its great robustness against adversarial examples as well as comparable classification accuracy to non-robustified counterparts on clean samples. Our framework is simple, effective and substantially extends the application of adversarial defense techniques to realistic scenarios where clean and adversarial samples are mixed.

Keywords: Deep Neural Networks, Adversarial Robustness

1 Introduction

In the past decades, CNN based classification models have been successfully applied to a variety of important systems such as finance [4], security [30] and driving assistants [27]. In these real-world applications, system safety is often deemed to enjoy higher superiority over the performance. However, CNN models are revealed to be highly vulnerable to adversarial examples [31, 3] — even adding a few visually imperceptible perturbations could easily fool CNNs to make fatal predictions. With the ever growing applications of CNNs, their safety issue becomes more significant and needs more attention.

\textsuperscript{\star} Equal contribution.
Fig. 1: The pipeline of our proposed RAIN. The input image $x_i$ first goes through structured randomization (SRd) module to be randomly shifted and randomly downsampled. This module enhances the robustness but leads to accuracy drop. Then, the downsampled image $x_i^{↓s}$ is sent to detail generation (DG) module to recover details for remedying accuracy. Lastly, the resultant image $x_i^{↓s↑}$ is fed into the given CNN.

To enhance adversarial robustness of CNNs, lots of adversarial defense approaches have been developed which can be roughly divided into three categories: input transformation, adversarial training, and randomization. The input transformation [14,36,17] methods transform input images to cause obfuscated gradients or project adversarial examples onto clean data manifold. Such methods are not universal, and can be evaded by adaptive attacks [2]. The second category, adversarial training methods such as [24,35], achieve outstanding robustness by training CNNs from scratch with both clean images and augmented adversarial examples. However, the data augmentation also consumes extensive computing resource compared to regular training. The randomization methods [34,28] achieve a balance between robustness and implementation cost by adding randomness to either input or DNN model architectures to mitigate adversarial effects. Due to randomness, the path of generating gradients varies from that of predicting adversarial examples with a high probability, thus the randomization modules mitigate the adversarial effects. Unfortunately, all the above mentioned defense methods have to pay the price of accuracy drop for enhanced robustness [29,37].

In this work, we propose a Robust and Accurate classification Network (RAIN) targeting at better robustness and meanwhile good accuracy. Our framework contains two modules: a structured randomization (SRd) module and a detail generation (DG) module. The SRd module contains random pooling and random shifting, which downsamples and shifts input images both in random procedures to destroy the adversarial perturbations. However, the added randomness also deducts the accuracy on clean images as in other defense methods [34,28]. We compare the images processed by the SRd module with the original ones, and find the processed images are more smooth and tend to lack rich high-frequency details. We further conduct empirical study to show removal of such
high-frequency details worsens the accuracy greatly. In view of such findings, we remedy the accuracy drop by recovering the high-frequency details of processed images. This is achieved by a detail generation (DG) module that implements a deep super-resolution model. The pipeline of our proposed RAIN is shown in Figure 1.

We evaluate the RAIN framework on STL10 and ImageNet datasets. Specifically, it achieves robustness of 68.6% under FGSM attack, and concurrently retains accuracy of 93.3% on ImageNet, outperforming the existing randomization-based baselines, and also maintaining the highest accuracy among all defense baselines.

To summarize, we make the following contributions in this work:

1) We propose a simple and practical framework, termed as RAIN, to help CNNs achieve enhanced robustness and meanwhile maintain high accuracy. The RAIN could be dropped in any given CNNs for enhancing their performance.

2) We introduce two simple yet effective structured randomization based defense methods. Besides serving as components of the RAIN framework, they are also of independent interest and can be integrated with other defense methods to improve their robustness further.

3) We reveal the origin of accuracy deduction through our proposed randomization and develop a corresponding solution to remedy the accuracy deduction. The experiments verify the compensation of our solution to the accuracy.

2 Preliminaries on Adversarial Robustness

In this section, we specify the notations, goals and capabilities of the adversary in our defense scenarios. We provide an overview of adversarial attack, and evaluation metrics of adversarial robustness. Suppose we are given a dataset \( D = \{ (x_i, y_i) \}_{i=1}^{N} \) with samples from \( c \) categories, where \( x_i \in \mathbb{R}^{h \times w \times 3} \) is an RGB color image and \( y_i \in \mathcal{Y} = \{ 1, 2, \ldots, c \} \) is its label. We train a CNN classification model, denoted as \( C(x) : \mathbb{R}^{h \times w \times 3} \to \mathcal{Y} \). Let \( L(C(x), y) \) be the loss function for evaluating the model prediction \( C(x) \) w.r.t. the label \( y \), which is typically a cross-entropy one.

2.1 Attack Models

Given an input image \( x \) from category \( y \), adversarial attack is to craft an adversarial example \( x^{\text{adv}} = x + \delta \) that causes prediction error of the classification model, i.e., \( C(x^{\text{adv}}) \neq y \). Here \( \delta \) is the additive and imperceptible adversarial perturbation generated by certain adversarial attack methods \( A_{\epsilon}(x) \). We take Fast Gradient Sign Method (FGSM) [16] for illustration, which is an effective one-step adversarial attack method. The adversarial examples are crafted as

\[
x^{\text{adv}} = A_{\epsilon}(x) = x + \alpha \cdot \text{sign}(\nabla_x L(C(x), y)),
\]  

(1)
where $\alpha$ is the step size that controls the magnitude of the added noise. FGSM is a representative one among gradient-based attack methods [12,6,25] which are based on the information of back-propagated gradients on the inputs to craft the perturbations.

For a fair comparison of different defense methods, a perturbation budget $\epsilon$ is specified so that any adversarial example must satisfy $\|x^{\text{adv}} - x\|_p \leq \epsilon$. In this paper, we only consider $l_\infty$-norm with the perturbation budget $\epsilon$ to define the adversary’s capability.

We first consider a white-box ensemble-pattern attack [34] in robustness evaluation. The adversary is aware of the given CNN model $C(x)$ and the defense module, as well as the internal gradients of them. The adversary can craft adversarial examples with the back-propagated gradients except for designing any adaptive attack modification. Such an attack scenario is more difficult to defend and thus, many previous defense methods only reported results under simpler vanilla attack scenarios [26,28]. Another attack scenario we consider is a score-based black-box attack, which only allows the adversary to access the logits output, and the accessing times will be restricted to a given maximum number.

### 2.2 Evaluation Metrics of Adversarial Robustness

A widely used metric to evaluate robustness is the prediction accuracy over the adversarial examples generated by certain attack methods [11]. Here, we only consider samples which have been classified correctly by the given CNN model before getting attacked. Formally, given a CNN model $C(x)$, we randomly collect a robustness test set, $\mathcal{T} = \{(x_j, y_j)\}_{j=1}^M$, where each element $(x_j, y_j) \in \mathcal{D}$ and $C(x_j) = y_j$. For a certain attack method $\mathcal{A}_\epsilon(x)$ with the perturbation budget $\epsilon$, the robustness is evaluated as

$$R(C, \mathcal{A}_\epsilon) = \frac{1}{M} \sum_{j=1}^M \mathbb{1}[C(\mathcal{A}_\epsilon(x_j)) = y_j].$$

The above equation calculates the accuracy of a given CNN model $C(x)$ on the adversarial examples crafted by a given adversarial attack method $\mathcal{A}_\epsilon(x)$ with perturbation budget $\epsilon$. Note, the given CNN model $C(x)$ could also be a defending model for robustness evaluation.

### 3 Randomization for Robustness

We propose a structured randomization module that consists of two operations, random pooling and random shifting, to defend against adversarial examples. In this section, we provide details of the two operations and reveal the mechanism behind their effectiveness. Lastly, we report robustness evaluation results of the two operations.
3.1 Structured Randomization

We first introduce random pooling and random shifting operations in detail. Both of them can damage the crafting of adversarial perturbations and enhance robustness.

**Random Shifting.** CNN is known to be almost shift-invariant due to its pooling and convolution layers, which means a small input shifts seldom affect the correct prediction of CNN. The shift-invariance of CNN inspires us to add randomness to CNN through shifting inputs slightly and randomly.

We design a random shifting operation over the input images before feeding them to the given CNN models. The input image is shifted differently for each inference as follows. First two shift values $\Delta h, \Delta w$ are randomly sampled from a uniform distribution, where $\Delta h \sim \mathcal{U}(-hp, hp), \Delta w \sim \mathcal{U}(-wp, wp)$. Here, the magnitude of $\Delta h, \Delta w$ is the number of shifted pixels, with sign indicating shifting direction; $h, w$ are the width and height of the input images, and $p$ is a predefined proportion and hence $p < 0.5$. Usually a very small $p$ is enough and we use $p = 0.05$ in our experiments. Then, the random shifting operation shifts the input image by $\Delta h$ vertically and by $\Delta w$ horizontally.

Figure 2 demonstrates our random shifting operation. Consequently, randomness is added to CNN, which mitigates the adversarial effect of adversarial examples.

**Random Pooling.** In addition to random shifting, we also introduce a random pooling operation to improve the robustness further. For any input image, the random pooling operation divides the image into non-overlap $2 \times 2$ patches completely. Next, it randomly picks one pixel from each patch with the uniform probability. The resultant images, denoted by $x^\downarrow$, are downsampled with factor $= 0.5$ into the size of $\frac{1}{2}h \times \frac{1}{2}w \times 3$. For the CNN classifier trained on the original dataset $\mathcal{D}$, we upsample $x^\downarrow$ into the original size $h \times w \times 3$ for prediction. In specific, the bicubic upsampling was applied here firstly. (We replace bicubic upsampling with the detail generation module (DG) in section 5.) We use superscripts $\downarrow$ and $\uparrow$ to represent the downsampling and upsampling operations respectively, and the upsampled image is denoted as $x^\uparrow$.

Implementation details of the random shifting and random pooling are summarized in Algorithm 1.
Algorithm 1 Structured randomization

Input: Input image $x$, CNN model $C(x)$, random shifting maximum proportion $p$

1: $x = \text{RANDOM\_POOLING}(x)$
2: $x = \text{RANDOM\_SHIFT}(x, p)$
3: $x = \text{Upsample}(x, \text{factor} = 2)$
4: $y_{\text{pred}} = \text{softmax}(C(x))$

Output: Prediction $y_{\text{pred}}$

5: \textbf{function} \text{RANDOM\_POOLING}(x)
6: \hspace{1em} \textbf{for} $i$ in $(0, \lceil \frac{1}{2}h \rceil - 1)$ and $j$ in $(0, \lceil \frac{1}{2}w \rceil - 1)$ \textbf{do}
7: \hspace{2em} Randomly initialize $\Delta i \sim \mathcal{U}([0, 1]), \Delta j \sim \mathcal{U}([0, 1])$
8: \hspace{2em} $x^\downarrow(i, j) = x(\min(2i + \Delta i, h - 1), \min(2j + \Delta j, w - 1))$
\hspace{1em} \textbf{return} $x^\downarrow$

9: \textbf{function} \text{RANDOM\_SHIFT}(x, p)
10: \hspace{1em} \textbf{for} $i_s$ in $(0, h - 1)$ and $j_s$ in $(0, w - 1)$ \textbf{do}
11: \hspace{2em} Assert $p < 0.5$
12: \hspace{2em} Randomly initialize $\Delta h \sim \mathcal{U}(-hp, hp), \Delta w \sim \mathcal{U}(-wp, wp)$
13: \hspace{2em} $i \equiv (i^s + \Delta h) \mod h$
14: \hspace{2em} $j \equiv (j^s + \Delta w) \mod w$
15: \hspace{2em} $x^s(i^s, j^s) = x(i, j)$
\hspace{1em} \textbf{return} $x^s$

3.2 Randomization Brings Robustness

As we conjecture, that randomization improves robustness is due to the caused misalignment between the fixed activation path of adversarial perturbations and the random inference path, which is analysed as below.

Given a clean image $x_i$, when an adversary computes the gradient for crafting the adversarial perturbation, the gradient is generated by $\delta_{i1} = \nabla_x C(x)|_{x=x^1_i}$, where $x^1_i$ is the random shifted version of $x_i$ w.r.t. $\Delta h_{i1}$ and $\Delta w_{i1}$. The inference path for generating gradients $\delta_{i1}$ is denoted by $(\Delta h_{i1}, \Delta w_{i1})$. Then, for the prediction of the crafted adversarial example $x_i + \delta_{i1}$, the classifier CNN follows the inference path $(\Delta h_{i2}, \Delta w_{i2})$. Thus, the most adversarial perturbation corresponding to the inference path $(\Delta h_{i2}, \Delta w_{i2})$ changes to $\delta_{i2} = \nabla_x C(x)|_{x=x^2_i}$, where $x^2_i$ represents the random shifted version of $x_i$ with $\Delta h_{i2}$ and $\Delta w_{i2}$. The probability that $\delta_{i1}$ is the same with the most adversarial perturbation $\delta_{i2}$ is as low as $\Pr\{\Delta h_{i1} = \Delta h_{i2}, \Delta w_{i1} = \Delta w_{i2}\} = 1/(2ph \cdot 2pw)$. Thus, randomization brings the robustness against adversarial attack.

Similarly in the random pooling operation, the probability that pooling positions selected for generating gradients are the same with those in the adversarial example prediction path is also as low as $\Pr\{\delta_{i1} = \delta_{i2}\} = (1/4)h^2w^2$. The combination of two randomization operations makes the given CNN very robust against gradient-based adversarial attack. We conduct experiments to examine the robustness of this structured randomization module, and also attempt to find the best order of implementing the three steps including random shifting, random pooling and upsampling.
Table 1: Robustness evaluation experiments of three orders of steps. “P”, “B” and “S” stand for random pooling, bicubic upsampling and random shifting respectively. We evaluate the robustness on STL10 and ImageNet Datasets against the FGSM and PGD attacks. All three orders are more robust than the vanilla CNN model. Among them, “P S B” order achieves strongest robustness.

|                      | STL10          |          |          |          | ImageNet       |          |          |          |
|----------------------|----------------|----------|----------|----------|----------------|----------|----------|----------|
|                      | Clean Images   | FGSM-8/255| PGD-16/255|          | Clean Images   | FGSM-8/255| PGD-16/255|          |
| Original model       | 1.000          | 0.090    | 0.000    |          | 1.000          | 0.197    | 0.000    |          |
| P B S                | 0.810          | 0.710    | 0.227    |          | 0.642          | 0.566    | 0.121    |          |
| P S B                | 0.824          | 0.720    | 0.287    |          | 0.644          | 0.573    | 0.252    |          |
| S P B                | 0.806          | 0.712    | 0.270    |          | 0.639          | 0.563    | 0.244    |          |

3.3 Robustness Evaluation Experiments

Setup. The two randomization operations and the upsampling have three possible orders as listed in Table 1. Note, the upsampling has to be placed behind the random pooling. Here we conduct robustness verification experiments on the three orders. For reference, we also conduct robustness verification experiments on vanilla CNN model. The robustness is evaluated under FGSM attack with $\epsilon = 8/255$, and PGD attack with $\epsilon = 16/255$, stepsize = 1/255, iteration= 40.

We use STL10 [8] and ImageNet [9] datasets here for evaluation. The evaluation metrics of robustness are formulated in Section 2.2. We evaluate robustness by testing the prediction accuracy on a predefined robustness set $T$. The test set $T$ contains 5,000 samples from testing set which are predicted correctly by the given CNN model. The experiments in the following sections also follow the same settings if without any specification. We aim to verify the robustness and find the best order for the three steps, random shifting, random pooling and upsampling.

We use both well-trained Resnet [18] in STL10 and ImageNet datasets for robustness verification experiments. The Resnet on STL10 dataset contains 11 convolutional layers and 1 fully-connected layer. The Resnet50 on ImageNet dataset consists of 5 stages each with a convolution and identity block. The detailed architectures of the used Resnets are given in the appendix.

Results. The experiment results are listed in Table 1, from which we can find that all the combinations achieve more than 50% robustness under the FGSM attack. This verifies the effectiveness of randomization in enhancing model robustness. Furthermore, we find the order “P S B” achieves the best robustness performance for more than 60% under the FGSM attack and 20% under the PGD attack. As a reference, the vanilla CNN model only achieves 19.7% and
0.0% robustness performance under the FGSM and PGD attacks. This is a significant improvement on robustness of our proposed structured randomization module compared to the vanilla CNN model.

4 Analyzing Accuracy Drop from Robustness

Even with the proposed structured randomization that enhances robustness against adversarial examples, the accuracy of defense models on clean images is only around 80% on STL10 and around 64% on ImageNet, much worse than 100% accuracy of the vanilla CNN model on clean images. There is a significant drop in accuracy due to the pursuit of model robustness. In this section, we try to dig the root of such drop and then mitigate it.

To find the reasons of the accuracy drop, we compare the original images with the images processed after the structured randomization module, as shown in Figure 3. The left image has been downsampled, random shifted and upsampled in the structured randomization module. Compared to the right original image, it is obvious that the processed images are more smooth and lack of details. The details of an image are usually the high-frequency components of an image, and the frequency spectrum of Figure 3 verifies that point. Therefore, we hypothesize that the accuracy drop may come from loss of high-frequency components. We then conduct experiments to study the contribution of high-frequency components to the accuracy of a well-trained CNN model, to verify our hypothesis.

Setup. The test datasets and the corresponding trained CNN models we evaluate here are the same as Section 3.3 indicated.

We conduct the experiments in an ablative manner to examine the impact of losing high-frequency details on accuracy. We continually decrease the threshold for removing the high-frequency components from the image, and keep other
factors the same, to find the change of accuracy. The following are the steps in details. We first transfer the input image into the frequency domain by FFT: 

\[ z = \mathcal{F}(x) \]

where \( z \) is the image in frequency domain with complex values, of the same size \( h \times w \times 3 \) as \( x \). Then, we remove the high-frequency components from \( z \) beyond the given threshold \( r \), the resultant spectrum is:

\[
z'(\mu, \nu) = \begin{cases} 
0, & \text{if } \frac{d((\mu, \nu), (c_\mu, c_\nu))}{\frac{1}{2}\sqrt{h^2 + w^2}} \geq r; \\
z(\mu, \nu), & \text{otherwise}.
\end{cases}
\]  

(3)

Here \( d(\cdot, \cdot) \) is the Euclidean distance, \( (\mu, \nu) \) is the indexing in frequency domain and \( (c_\mu, c_\nu) \) is the indexing of the centroid, which represents the element with 0 frequency. \( (c_\mu, c_\nu) \) will be the same for the images with same size in frequency domain. Then, we calculate the energy of the resultant frequency spectrum. The energy \( E(\cdot) \) over a spectrum \( z \) is computed by:

\[
E(z) = \int \int |z(\mu, \nu)|^2 d\mu d\nu.
\]  

(4)

Noting that removing the high-frequency components will lead to the loss of corresponding spectral densities, thus the energy over the frequency spectrum also decreases, i.e. \( E(z') < E(z) \). Next, to avoid the influence of energy loss, we uniformly scale up \( z' \) by \( \sqrt{E(z)/E(z')} \) so that it holds the same energy as the original spectrum’s. Lastly, we could compare the impact on the accuracy of removing different proportion of high-frequency components fairly.

**Results.** As Figure 4 shows, removing high-frequency components does reduce the accuracy significantly. Along with decreasing the threshold \( r \), the accuracy drops very faster in both datasets even after holding the energy to be constant. These results verify that the damaged details after the structured randomization modules is an important factor leading to the accuracy drop.

As the destroyed details is an essential reason causing the drop of accuracy, we are motivated to develop an approach to recover the details and hence obtain a robust and accurate defense framework.
5  RAIN: Robust and Accurate Defense Network

Based on the experiments in the previous sections, we can see that the structured randomization module (SRd) in our RAIN framework can enhance the robustness of CNN models substantially on both STL10 and ImageNet datasets, but at the price of deducted accuracy. Besides, the damaged details caused by the SRd module are proved to be the important factors leading to the drop in accuracy. To remedy such accuracy drop, we introduce a detail generation (DG) module, which is detailed in Section 5.1. We replace bicubic upsampling in Algorithm 1 with the DG module to propose the complete framework of RAIN in this section. Then we conduct experiments to show our proposed framework contributes to better robustness with less sacrificing accuracy, against both white-box and black-box attacks.

5.1 Architectures of RAIN

Detail Generation. In our RAIN framework, to remedy accuracy drop due to pursuit of better robustness with the structured randomization (SRd) module, we apply a detail generation (DG) module. This module is implemented with a Super Resolution (SR) model to generate the details of the images processed by the structured randomization module. SR models are able to upsample low-resolution images and to enhance details [10]. The deep-learning-based SR models, such as EDSR [23], achieve impressive performance in Super-Resolution tasks. Therefore, we replace the bicubic upsampling operation with a deep-learning-based SR model, viz., the EDSR.

In order to show the effectiveness of EDSR for generating details, a certain image $x_i$ is processed respectively by the SRd + bicubic pipeline and the SRd + EDSR pipeline. The resultant images are denoted by $x_i^{\text{bicubic}}$ and $x_i^{\text{EDSR}}$. In Figure 5a, we compare the spectrum maps of the two resultant images. Given a spectrum $z(\mu, \nu)$, the spectral density at certain frequency $q$ is calculated by

$$D(q) = \log \left( \sum_{\mu^2 + \nu^2 = q^2} \frac{z(\mu, \nu)^2}{\sqrt{\mu^2 + \nu^2}} \right).$$

We implement Fourier transform over $x_i^{\text{bicubic}}$ and $x_i^{\text{EDSR}}$. The spectral density $D(q)$ in terms of frequency $q$ is plotted in Figure 5a. We can see that the spectrum of $x_i^{\text{EDSR}}$ contains stronger high-freq components in comparison to $x_i^{\text{bicubic}}$. Thereby, EDSR enhances high-freq details in the resultant images.

Overall Pipeline. The whole pipeline of the RAIN is as follows: for a given well-trained classification model $C(\cdot)$, RAIN first processes the images through the random pooling and the random shifting operations. Then, the well-trained EDSR model upsamples the images back to the regular size and enriches the details. Afterwards, the resultant images are fed to the given CNN classifier.
Fig. 5: Analysis of the effects of RAIN in processing images. (a) Spectrum comparison on the images processed by EDSR and Bicubic. (b) Comparison on feature maps from the Res3a block for the vanilla adversarial images (middle) and the image processed by RAIN (right). The strong responses of some locations have been alleviated. The adversarial images, originally recognized wrongly, are classified correctly after RAIN.

$C(\cdot)$. Lastly, to better generate the detail that are useful to $C(\cdot)$, a few fine-tuning operations for the parameters of the SR model are performed to better remedy the drop of accuracy.

5.2 Robustness to White-box Attacks

We conduct experiments to compare our RAIN with other adversarial defense methods under white-box attacks.

Experiment Proposal We compare our proposed RAIN with other baselines under white-box adversarial attack. The experiment settings and evaluation metrics are the same as the section 3.3 indicates. The EDSR model contains 16 residual blocks, 64 filters and are trained in DIV2K dataset [1]. The robustness are evaluated under FGSM attack with $\epsilon = 8/255$, and PGD attack with $\epsilon = 16/255$, stepsize = 1/255, iteration= 40. We choose four recent baselines as the benchmark of our evaluation. Two of the baselines are randomization-based defense methods, Random-padding [34] and Pixel deflection[28]. The rest two are adversarial-training based defense methods, adversarial training [24] and feature denoising [35].

Experiment Results Table 2 shows the result of accuracy and robustness under white-box attacks. Our proposed RAIN outperforms the other four baselines both in accuracy (92.9%,93.3%) and robustness (74.5%,68.6%) under the FGSM attack in both datasets. Although the two adversarial-training-based baselines
Table 2: Robustness Evaluation of RAIN and baselines under white-box attacks on STL10 and ImageNet datasets. The white-box attacks are end-to-end FGSM attack with $\epsilon = 8/255$ and PGD attack with $\epsilon = 16/255$.

|          | STL10  | Accuracy | Robustness |
|----------|--------|----------|------------|
| Clean Images | FGSM-8/255 | PGD-16/255 |
| Pixel Deflection [28] | 0.883 | 0.286 | 0.065 |
| Random Padding Resizing [34] | 0.907 | 0.576 | 0.070 |
| Adversarial Training [24] | 0.705 | 0.592 | 0.649 |
| Feature Denoising [35] | 0.696 | 0.631 | **0.668** |
| RAIN      | **0.929** | **0.745** | **0.237** |

|          | ImageNet | Accuracy | Robustness |
|----------|----------|----------|------------|
| Clean Images | FGSM-8/255 | PGD-16/255 |
| Pixel Deflection [28] | 0.858 | 0.406 | 0.117 |
| Random Padding Resizing [34] | 0.928 | 0.644 | 0.154 |
| Adversarial Training [24] | 0.623 | 0.620 | 0.417 |
| Feature Denoising [35] | 0.653 | 0.648 | **0.455** |
| RAIN      | **0.933** | **0.686** | **0.273** |

achieves best robustness under the PGD attack, they have much worse accuracy. More importantly, both adversarial-training-based baselines are trained from scratch with adversarial examples crafted by the PGD attack with same $\epsilon = 16/255$, which makes them more robust against iterative attack methods. Figure 5 show the difference of feature map with and without RAIN for a given CNN model. We can see that the RAIN mitigate the adversarial effect of malicious examples.

5.3 Robustness to Black-box Attacks

We then compare our proposed RAIN with the baselines under black-box adversarial attacks to test its robustness.

Experiment Proposal The experiments follow the same benchmark as the white-box attack experiments in section 5.2 apart from the adversarial attack methods. We select three black-box adversarial attack method, ZOO [7] and NES [19] to evaluate our defense approach RAIN. The perturbation budget $\epsilon = 8/255$ for all experiments under black-box adversarial attack.

Experiment Results Table 3 shows the result of accuracy and robustness under black-box attacks. Similar to the experiment result in white-box attack. Our propose RAIN achieved both highest robustness(91.2%,88.5%) among the four baselines under FGSM attack in both datasets. The experiment results verify that our proposed RAIN are robust under different adversarial attack methods and different datasets. Last but not least, our proposed RAIN does little harm to the accuracy with better robustness.
Table 3: Robustness Evaluation of our proposed RAIN and baselines under black-box attacks on STL10 and ImageNet datasets. All the black-box attacks are with $\epsilon = 8/255$

|                | Clean Images | ZOO-8/255 | NES-8/255 |
|----------------|--------------|-----------|-----------|
| **STL10**      |              |           |           |
| Pixel Deflection [28] | 0.883       | 0.679     | 0.650     |
| Random Padding Resizing [34] | 0.907       | 0.854     | **0.873** |
| Adversarial Training [24] | 0.705       | 0.705     | 0.663     |
| Feature Denoising [35] | 0.696       | 0.621     | 0.594     |
| RAIN            | **0.929**    | **0.912** | 0.871     |

| **ImageNet**    |              |           |           |
| Pixel Deflection [28] | 0.858       | 0.846     | 0.841     |
| Random Padding Resizing [34] | 0.928       | 0.867     | 0.881     |
| Adversarial Training [24] | 0.623       | 0.620     | 0.611     |
| Feature Denoising [35] | 0.653       | 0.663     | 0.641     |
| RAIN            | **0.933**    | **0.885** | **0.882** |

Discussion and Future Work We can see that the DG module increases the spectral density in the high-frequency part, which is the procedure of generating details. Although the SR model in DG module shows good performance in generating details, the DG module is not limited by SR model. Other generative models such as GAN [22] and image enhancement [15] would also be suitable for the detail generating. We will keep on investigating different solution for remedying accuracy in the future.

6 Related Work

Adversarial Attack The investigation of adversarial examples was initiated by [31], which shows well-crafted visually imperceptible perturbations could cause prediction error of well-trained CNNs. Later, Fast Gradient Sign Method (FGSM) [16] was developed to compute such adversarial perturbations by conducting gradient ascent on the original input. Following FGSM, several powerful iterative adversarial attack methods, including DeepFool [25], PGD [24], MIFGSM [12] and C&W attack [6] were developed. All of them need to access the internal back-propagated gradients on the original images to generate attacks and thus are called the white-box attack methods. On the contrary, the gradients are not available in the black-box setting. To generate attacking perturbations, ZOO [7] applies symmetric difference quotient [21] to estimate the back-propagated gradient of each pixel according to the output change to the queries. Though achieving comparable attack effect as many white-box methods, it requires an excessive number of queries for gradients estimation. Recently, many researchers focus on improving the black-box attack efficiency [33,20,13]. For instance, a natural evolution strategy (NES) was proposed by [21,19] to estimate the back-propagated gradient on the original images.
Adversarial Defense Randomization was introduced by [28] at inference to obfuscate back-propagated gradients, by randomly sampling and replacing the pixels with their neighbors. Besides, random resizing and padding layers [34] are also able to interrupt the gradient computation and thus impair the attack methods. Such methods perform well against both white-box attack and black-box attack methods [11]. The randomization-based defense methods do not require training from scratch and can be applied to robustify any well-trained CNN models directly and quickly. This is one of their significant advantages over other defense methods. Adversarial training from scratch could improve more in robustness against adversarial examples [24]. In addition to adversarial training, a recent work [35] proposes a feature denoising filter to improve the robustness further, by non-local means [5] to denoise the perturbed features. Although the adversarial training methods offer stronger robustness, they require much longer training time and still suffer accuracy drop.

Image super-resolution (SR) issue is also explored to improve robustness of CNN models but the relevant studies are very few. A recent work [26] uses SR to upsample the adversarial examples into the natural image manifold and applies wavelet denoising on upsampled examples for defense. However, the authors only tested their method in black-box attack scenarios, where the SR model is unknown to the adversary. Our experiments show their method is fragile in the ensemble-pattern white-box attack scenario, where the adversary is aware of the defense module and the gradients. More importantly, their released codes show that they only upsampled the adversarial examples and did not resize the examples back to the original size. In contrast, the clean images were not upsampled by SR. Such leaked prior information makes the accuracy evaluation unfair. Different from that work, our RAIN processes adversarial examples the same as clean images, we apply SR to recover the details of the input images to remedy accuracy drop, and we evaluate our RAIN also in the white-box scenarios.

The performance decline on clean images is the cost of enhanced robustness. The trade-off between adversarial robustness and accuracy on clean images was reported in [29]. Consecutively, a theoretical explanation was given in [32], claiming that the trade-off exists due to the features learned by robust models and accurate models are fundamentally different, whereas they also did not propose a solution to improving the trade-off either.

7 Conclusions

In this paper, we enhance the robustness and at the same time maintain the accuracy of given CNNs by proposing a RAIN framework. The RAIN contains a structured randomization module and a detail generation module. The structured randomization module downsamples and shifts the input images randomly for improved robustness. We then investigate the root of performance drop after applying the randomization module through observation, experiments and analysis. We find the detail information is damaged in the randomization progress
which leads to accuracy drop. Inspired by such findings, we devise a deep Super Resolution model as the detail generation module to upsample and recover the details loss during the structured randomization module, thus to remedy the accuracy deduction. Last but not least, the evaluation experiments conducted on the STL10 and ImageNet datasets confirm the robustness improvement and maintained accuracy of our proposed framework.
References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 126–135 (2017)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
3. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. In: Joint European conference on machine learning and knowledge discovery in databases. pp. 387–402. Springer (2013)
4. Bottou, L., Bengio, Y., Le Cun, Y.: Global training of document processing systems using graph transformer networks. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 489–494. IEEE (1997)
5. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 60–65. IEEE (2005)
6. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE (2017)
7. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J.: 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. pp. 15–26. ACM (2017)
8. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223 (2011)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
10. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence 38(2), 295–307 (2015)
11. Dong, Y., Fu, Q.A., Yang, X., Pang, T., Su, H., Xiao, Z., Zhu, J.: Benchmarking adversarial robustness. arXiv preprint arXiv:1912.11852 (2019)
12. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9185–9193 (2018)
13. Du, J., Zhang, H., Zhou, J.T., Yang, Y., Feng, J.: Query-efficient meta attack to deep neural networks. arXiv preprint arXiv:1906.02398 (2019)
14. Dziugaite, G.K., Ghahramani, Z., Roy, D.M.: A study of the effect of jpg compression on adversarial images. arXiv preprint arXiv:1608.00853 (2016)
15. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. ACM Transactions on Graphics (TOG) 36(4), 1–12 (2017)
16. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
17. Guo, C., Rana, M., Cisse, M., Van Der Maaten, L.: Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117 (2017)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
19. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. arXiv preprint arXiv:1804.08598 (2018)
20. Ilyas, A., Engstrom, L., Madry, A.: Prior convictions: Black-box adversarial attacks with bandits and priors. arXiv preprint arXiv:1807.07978 (2018)
21. Lax, P.D., Terrell, M.S.: Calculus with applications. Springer (2014)
22. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
23. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 136–144 (2017)
24. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
25. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
26. Mustafa, A., Khan, S.H., Hayat, M., Shen, J., Shao, L.: Image super-resolution as a defense against adversarial attacks. IEEE Transactions on Image Processing 29, 1711–1724 (2019)
27. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against deep learning systems using adversarial examples. arXiv preprint arXiv:1602.02697 1(2), 3 (2016)
28. Prakash, A., Moran, N., Garber, S., DiLillo, A., Storer, J.: Deflecting adversarial attacks with pixel deflection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8571–8580 (2018)
29. Su, D., Zhang, H., Chen, H., Yi, J., Chen, P.Y., Gao, Y.: Is robustness the cost of accuracy? a comprehensive study on the robustness of 18 deep image classification models. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 631–648 (2018)
30. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6479–6488 (2018)
31. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
32. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. arXiv preprint arXiv:1805.12152 (2018)
33. Tu, C.C., Ting, P., Chen, P.Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.J., Cheng, S.M.: Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 742–749 (2019)
34. Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.: Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991 (2017)
35. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 501–509 (2019)
36. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
37. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573 (2019)