An Effective and Efficient Preprocessing-based Approach to Mitigate Advanced Adversarial Attacks

Han Qiu[1], Yi Zeng[2*], Qinkai Zheng[3], Tianwei Zhang[4], Meikang Qiu[5], Bhavani Thuraisingham[6]

[1]Telecom Paris, Institut Polytechnique de Paris, Palaiseau, France, 91120.
[2]University of California San Diego, CA, USA, 92122.
[3]Shanghai Jiao Tong University, Shanghai, China, 200240.
[4]Nanyang Technological University, Singapore, 639798.
[5]Texas A&M University, Texas, USA, 77843.
[6]The University of Texas at Dallas, Texas, USA, 75080.

Abstract

Deep Neural Networks are well-known to be vulnerable to *Adversarial Examples*. Recently, advanced gradient-based attacks were proposed (e.g., BPDA and EOT), which can significantly increase the difficulty and complexity of designing effective defenses. In this paper, we present a study towards the opportunity of mitigating those powerful attacks with only pre-processing operations. We make the following two contributions. First, we perform an in-depth analysis of those attacks and summarize three fundamental properties that a good defense solution should have. Second, we design a lightweight preprocessing function with these properties and the capability of preserving the model's usability and robustness against these threats. Extensive evaluations indicate that our solutions can effectively mitigate all existing standard and advanced attack techniques, and beat 11 state-of-the-art defense solutions published in top-tier conferences over the past 2 years.

Introduction

Szegedy et al. (Szegedy et al. 2013) proposed the concept of *Adversarial Examples* (AEs): with imperceptible modifications to the input, the Deep Learning (DL) model will be fooled to give wrong prediction results. Since then, a huge amount of research effort has been spent to enhance the powers of the attacks, or mitigate the new attacks. This leads to an arms race between adversarial attacks and defenses. Basically, the generation of AEs can be converted into an optimization problem: searching for the minimal perturbations that can cause the model to predict a wrong label. Standard attacks used the gradient-based approaches to identify the optimal perturbations (e.g., FGSM (Goodfellow, Shlens, and Szegedy 2014), I-FGSM (Kurakin, Goodfellow, and Bengio 2016), LBFGS (Szegedy et al. 2013), C&W (Carlini and Wagner 2017b)). To defeat those attacks, a lot of defenses were proposed to obfuscate the gradients such as making them shattered or stochastic (Guo et al. 2018; Prakash et al. 2018; Xie et al. 2018; Buckman et al. 2018).

Unfortunately, those gradient obfuscation-based defenses were further broken by advanced attacks (Athalye, Carlini, and Wagner 2018; Athalye et al. 2018). *Backward Pass Differentiable Approximation* (BPDA) was introduced to handle the shattered gradients by approximating the gradients of non-differentiable functions. *Expectation over Transformation* (EOT) was designed to deal with the stochastic gradient by calculating the expectation of gradients of random functions. These two attacks have successfully defeated the previous defenses (Athalye, Carlini, and Wagner 2018), and even new defenses published after their disclosure was still proven to be vulnerable to either BPDA, EOT, or their combination (Tramer et al. 2020).

The question we want to address is: *is it possible to continue the arms race by mitigating the aforementioned advanced attacks with more robust defense solutions?* This is a challenging task. First, these attacks assume very high adversarial capabilities (Tramer et al. 2020): the attacker knows every detail of the DL model and the potential defenses. This significantly increases the difficulty of defense designs and invalidates existing solutions that require to hide the model or defense mechanisms. Second, BPDA and EOT target the root causes of gradient obfuscation: the non-differentiable operation can always be approximated, and the random operation can be estimated by its expectation. It is indeed difficult for the defender to bypass these assumptions while still preserving model usability.

One possible defense strategy is adversarial training (Kurakin, Goodfellow, and Bengio 2016): we can keep generating adversarial examples from the training-in-progress model using the Projected Gradient Descent (PGD) attack technique, and augmenting them into the training set to improve the model's robustness. This strategy is shown to be effective against different types of adversarial attacks including BPDA and EOT. However, it can bring a significant cost to perform adversarial training with large-scale DNN models and datasets. So we are more interested in an efficient method, which can be directly applied to a given model without altering it. (Raff et al. 2019) proposed a preprocessing-based solution: they tested 25 existing preprocessing functions and placed them into 10 groups. For each inference, an ensemble of 5∼10 functions is randomly selected to transform the input before feeding it to the target model. This

*Yi Zeng and Han Qiu have the equal contribution.

strategy can mitigate a more sophisticated BPDA, where the adversary attempts to use a neural network to approximate the non-differentiable operations.

In this paper, we also focus on the preprocessing-based defense to enhance the model's robustness against all existing adversarial attacks. Different from (Raff et al. 2019), we aim to utilize a single lightweight transformation function to preprocess the input images. This is expected to significantly reduce the computation cost and logic complexity for model inference, which is critical when the task is deployed in resource-constrained edge and IoT devices. To achieve this goal, we make the following contributions.

First, we analyze the features and assumptions of different attacks and identify three properties for designing a qualified preprocessing function $g(\cdot)$. The first one is *usability-preserving*, which is to guarantee $g(\cdot)$ will not affect the model performance on clean samples. The next two properties are *non-differentiability* and *non-approximation*, to enhance the model robustness against both standard and advanced gradient-based attacks.

Second, we introduce a novel preprocessing function that can meet the above properties. Our function consists of two steps: (1) a DCT-based quantization is used to compress the input images, which can achieve *non-differentiability*; (2) a dropping-pixel strategy is further introduced to distort the image via random pixel dropping and displacement. This step can increase the difficulty and fidelity of *approximation*. Both steps are *usability-preserving*, thus their integration will cause a negligible impact on the model performance.

We conduct extensive experiments to show the effectiveness of our solutions. It can constrain the attack success rate under 7% even with 10000 rounds of BPDA+EOT attack (dozens of GPU hours for 100 samples), which significantly outperform 11 state-of-the-art gradient obfuscation defenses published recently in top-tier conferences. We release our code online[1] to better promote this research direction.

Backgrounds

In this section, we briefly review the arms race between adversarial attacks and defenses on DNN models.

Round 1: attack. L-BFGS (Szegedy et al. 2013) was initially adopted to solve the optimization problem of AE generation. Then, many gradient-based attacks were introduced: gradient descent evasion attack (Biggio et al. 2013), *Fast Gradient Sign Method* (FGSM) (Goodfellow, Shlens, and Szegedy 2014), I-FGSM (Kurakin, Goodfellow, and Bengio 2016), MI-FGSM (Dong et al. 2017), Deepfool (Moosavi-Dezfooli, Fawzi, and Frossard 2016).

Round 2: defense. Three categories of defenses against the above attacks were proposed. The first direction is adversarial training (Kurakin, Goodfellow, and Bengio 2016; Huang et al. 2015; Shaham, Yamada, and Negahban 2018), where AEs are used with normal examples together to train DNN models to recognize and correct malicious samples. The second direction is to train other models to assist the target one such as Magnet (Meng and Chen 2017) and Generative Adversarial Trainer (Lee, Han, and Lee 2017). The third direction is to design AE-aware network structures or loss functions, such as Deep Contractive Networks (Gu and Rigazio 2014), Input Gradient Regularization (Ross and Doshi-Velez 2018), and Defensive Distillation (Papernot et al. 2016).

Round 3: attack. A more powerful attack, C&W (Carlini and Wagner 2017b), was proposed by updating the objective function to minimize l_p distance between AEs and normal samples. C&W can effectively defeat Defensive Distillation (Carlini and Wagner 2017b) and other defenses with assisted models, e.g. Magnet (Carlini and Wagner 2017a).

Round 4: defense. Since then, gradient obfuscation was introduced to improve the defense. Five input transformations were tested to counter AEs in (Guo et al. 2018): image cropping and rescaling, Bit-depth Reduction (BdR), JPEG, Total Variance minimization (TV), and image quilting. Random functions were also proposed for defense such as Pixel Deflection (PD) (Prakash et al. 2018), Randomization layer (`Rand`) (Xie et al. 2018), and SHIELD (Das et al. 2018). Those solutions are effective against all prior attacks.

Round 5: attack. To defeat the gradient obfuscation techniques, two advanced gradient approximation attacks were designed. BPDA (Athalye, Carlini, and Wagner 2018) copes with non-differentiable operations by approximating the gradients. EOT (Athalye et al. 2017) deals with random operations by averaging the gradients of multiple sessions. (Tramer et al. 2020).

Round 6: defense. Although a large number of defense works were published after the discovery of BPDA and EOT, most of them did not consider or correctly evaluate these two attacks. One promising defense solution is adversarial training (Kurakin, Goodfellow, and Bengio 2016), which augments the training data with the adversarially-crafted examples. This is indeed an effective approach to defeat these advanced attacks. But it can incur a high cost as it needs to keep generating adversarial examples adaptively during training. It also suffers from the scalability issue, especially with large-scale datasets. An alternative solution is the ensemble of different preprocessing functions to increase the difficulty of AE generation (Raff et al. 2019). This is also in a lack of efficiency, as it requires the implementations of dozens of preprocessing methods to guarantee the model robustness. How to identify a simple yet effective solution against both standard and advanced gradient approximation attacks is still worth research effort, which we aim to explore in this paper.

Threat Model and Problem Definition

Threat Model

There are two main types of adversarial attacks (Carlini and Wagner 2017b): untargeted attacks try to mislead the DNN models to an arbitrary wrong label, while targeted attacks succeed only when the DNN model predicts the input as one specific label desired by the adversary. In this paper, we only evaluate the targeted attacks. The untargeted attacks can be mitigated in a similar way.

We consider a white-box scenario: the adversary has all

[1]Link removed for anonymity. The code and data are uploaded in supplementary materials.

details of the DNN model and full knowledge of the defense. The adversary is outside of the DNN system and is not able to compromise the inference process or the model parameters. In the context of computer vision, he can directly modify the pixel values of the input image within a certain range. We use l_2 norm to measure the scale of added perturbations. We allow AEs with a maximum l_2 distortion of 0.05 (as in prior work (Athalye, Carlini, and Wagner 2018).)

Defense Requirements

Based on the above threat model, we list a couple of requirements for a good defense solution.

First, no modifications on the target DNN model are allowed, such as training an alternative model with different structures (Papernot et al. 2016) or datasets (Yang et al. 2019); or retraining the target model with AEs (e.g. adversarial training (Tramèr et al. 2017)). We set this requirement since (1) training or retraining a model significantly increases the computation cost and may not be practical on large-scale datasets like ImageNet; (2) these defenses lack generality to cover different attacks since they "explicitly set out to be robust against one specific threat model" (Carlini et al. 2019).

Second, we consider adding a preprocessing function over the input samples before feeding them into the DNN model. Such function can either remove the impacts of adversarial perturbations on the inference or make it infeasible for the adversary to generate AEs adaptively even in the white-box scenario. This function should be independent of the datasets and DNN models of similar tasks.

Third, this solution should be lightweight with a negligible influence on the computation cost or performance of the inference task. Input preprocessing can introduce a trade-off between security and usability: the side effect of correcting AEs may also alter the prediction results of benign samples. Designing a defense preprocessing function should balance this trade-off with maximum impact on the AEs and minimal impact on the benign ones.

Methodology Insights

We aim to design a preprocessing function $g(\cdot)$, which transforms an input image $x \in \mathcal{X}$ to an output with the same dimension. Then given a DNN model $f(\cdot)$, the inference process becomes $y = f(g(x))$. This function $g(\cdot)$ needs to mitigate the adversarial attacks within the threat model and satisfy the defense requirements, as described in the previous section. We identify some properties and design philosophy of a good methodology in this section and give a specific algorithm in the next section.

First, this preprocessing function must preserve the usability of the target model, i.e., exerting minimal influence on the accuracy of clean samples. This gives the first property:

Property 1 *(Usability-preserving) $g(\cdot)$ cannot affect the prediction results of clean input: $f(g(x)) \approx f(x)$, $\forall x \in \mathcal{X}$*

Second, as most of the attacks generate adversarial examples by calculating the gradients of the model parameters. When a preprocessing function is introduced, this calculation becomes: $\bigtriangledown_x f(g(x)) = \bigtriangledown_x f(x) \bigtriangledown_x g(x)$. So a common approach is shattered gradient-based defense, where the preprocessing operation $g(\cdot)$ is designed to be non-differentiable. With this property, the adversary is not able to craft AEs based on the gradient of the model using standard methods (e.g., FGSM, C&W, Deepfool, etc.).

Property 2 *(Non-differentiability) $g(\cdot)$ is non-differentiable, i.e., it is hard to compute an analytical solution for $\bigtriangledown_x g(x)$.*

It is interesting to note that this property can defeat the advanced EOT attack (Athalye, Carlini, and Wagner 2018) as well. This attack was proposed to invalidate the defense solutions based on model input randomization, by statistically computing the gradients over the expected transformation of the input x. Formally, for a preprocessing function $g(\cdot)$ that randomly transforms x from a distribution of transformations T, EOT optimizes the expectation over the transformation with respect to the input by: $\bigtriangledown_x \mathbb{E}_{t\sim T} f(g(x)) = \mathbb{E}_{t\sim T} \bigtriangledown_x f(g(x))$. EOT can help to get a proper expectation with samples at each gradient descent step. However, if $g(\cdot)$ is non-differentiable, the adversary cannot calculate the gradient expectation to generate AEs either.

A function $g(\cdot)$ with the non-differentiability property can still be vulnerable to sophisticated attacks, e.g., BPDA (Athalye, Carlini, and Wagner 2018), where the adversary can approximate $g(\cdot)$ with a differentiable function $g'(x)$. For instance, in the experimentation of the initial BPDA attack (Athalye, Carlini, and Wagner 2018), the adversary used $g'(x) = x$ as an approximation to calculate the gradient of $g(x)$. He keeps $g(\cdot)$ on the forward pass and replaces it with x on the backward pass. The derivative of the $g(\cdot)$ will be approximated as the derivative of the identity function, which is 1. In (Raff et al. 2019), neural nets were further trained to approximate non-differentiable functions, which can defeat a wider range of shattered gradient-based defenses than the identity function. To mitigate such threats, the preprcessing function must meet the following property:

Property 3 *(Non-approximation) It is difficult to find a differentiable $g'(x)$ that can approximate the non-differentiable preprocessing function $g(x)$ when calculating its gradients, i.e., $\bigtriangledown_x g'(x) \approx \bigtriangledown_x g(x)$.*

A common strategy to reduce the possibility and fidelity of approximating a non-differentiable function is to add randomization in the operation. If the degree of randomization is large enough, then it will be difficult for the adversary to find a qualified deterministic differentiable function for replacement, even using neural networks. However, a high random transformation can also affect the model's usability (Property 1). So the key to the design of this function $g(\cdot)$ is to balance the trade-off between Properties 1 and 3 with a random non-differentiable operation. Past work (Raff et al. 2019) adopted an ensemble of dozens of weak preprocessing functions to defend against BPDA, making the entire inference system quite complex. In this paper, we aim to simplify this by designing one single function to achieve the same goal.

Summary

A preprocessing function $g(\cdot)$ that can meet the above three properties can effectively increase the DNN model's robustness against existing adversarial attacks. Specifically, for standard gradient-based attacks (FGSM, C&W, LBFGS, Deepfool), non-differentiability in Property 2 can prohibit the direct calculation of gradients, and the randomization employed in Property 3 can obfuscate the gradient values. A function with these two properties can provide higher robustness against these standard attacks.

For those advanced attacks, the gradient expectation attack (EOT) can be mitigated by Property 2. If a qualified function with Property 3 is identified, the adversary may have difficulty in discovering a replacement that can accurately approximate this function. Then gradient approximation attack (BPDA) becomes infeasible or at least requires a much higher cost. The combination of these two attacks cannot compromise the model's robustness either.

Our Proposed Solution

Our proposed function $g(\cdot)$ involves two critical steps to process the input images. The first step adopts a DCT-based defensive quantization. Based on (Liu et al. 2019), we further improve the quantization table to better adapt to the machine's visionary behavior. This can realize the non-differentiability property while preserving the model's usability. The second step is inspired by a dropping-pixel strategy (Xie et al. 2018; Guo et al. 2018). We propose a novel technique to distort images by dropping randomly selected pixels of input images and displacing each pixel away from the original coordinates. This can achieve highly randomized outputs while keeping a high model accuracy.

Step 1: DCT-based Quantization

The first step is described in Lines 4-9 in Algorithm 1. The input image is cut into grids of pixels with the size of the grid d. Pixels in each grid are transformed into the frequency space via Discrete Cosine Transform (DCT) (Ahmed, Natarajan, and Rao 1974) as shown. Here we use a 2D-DCT with a grid size of 8×8. A defensive quantization table Q is then used to quantize all the frequency coefficients. These DCT coefficients are further de-quantized and transformed back into the spatial space with an inverse DCT.

The critical factor in this step is the quantization table Q. (Guo et al. 2017) directly used the JPEG quantization table Q_{50} to remove the adversarial perturbations. This was proved ineffective. as the JPEG quantization table was designed to compress the image based on the sensitivity of the human visual system. Later on, more effective approaches were improved to defeat certain adversarial attacks with randomized quantization tables (Das et al. 2018) or a new quantization table (Liu et al. 2019). Each time when adversary generates AEs, the perturbation on pixel values must be large enough to influence the quantization results. In our solution, we introduce a novel and more effective way to generate the quantization table, as shown in Algorithm 2.

We generate our new quantization table Q in a statistical learning manner by summarizing the patterns of the AEs. Specifically, first, all the 8×8 blocks in the spatial space (I in Line 8) are collected from all the images' color channels for both the clean image set and the AE set ($\dot{I}$ in Line 9).

ALGORITHM 1: *Defense Preprocessing Function.*

Input: original image $I \in \mathbb{R}^{h \times w}$
Output: processed image $I' \in \mathbb{R}^{h \times w}$
Parameters: defensive quantization table Q, distortion limit $\delta \in [0, 1]$, size of grid d.

1 $x_0 = 0, y_0 = 0$;
2 $n_w = w/d, n_h = h/d$;
3 $\mathcal{G}_I = \{(x_m, y_n) | (m, n) \in \{(0, ..., n_w) \times (0, ..., n_h)\}\}$;
4 **for** (x_m, y_n) *in* $\mathcal{G}_I \setminus \{(x_0, y_0)\}$ **do**
5 $\quad dct = DCT(I(x_{m-1} : x_m, y_{n-1} : y_n))$;
6 $\quad dct_q = Quantization(dct, Q)$;
7 $\quad dct_d = Dequantization(dct_q, Q)$;
8 $\quad I_{quantized} = IDCT(dct_d)$;
9 **end**
10 **for** (x_m, y_n) *in* $\mathcal{G}_I \setminus \{(x_0, y_0)\}$ **do**
11 $\quad \delta_x \sim \mathcal{U}(-\delta, \delta)$;
12 $\quad \delta_y \sim \mathcal{U}(-\delta, \delta)$;
13 $\quad x_m = x_{m-1} + d \times (1 + \delta_x)$;
14 $\quad y_n = y_{n-1} + d \times (1 + \delta_y)$;
15 **end**
16 $\mathcal{G}_{I'} = \{(x'_m, y'_n) | x'_m = d \times m, y'_n = d \times n, (m, n) \in \{(0, ..., n_w) \times (0, ..., n_h)\}\}$;
17 **for** (x'_m, y'_n) *in* $\mathcal{G}_{I'} \setminus \{(x'_0, y'_0)\}$ **do**
18 $\quad I'(x'_{m-1} : x'_m, y'_{n-1} : y'_n) = \mathtt{Remap}(I_{quantized}(x_{m-1} : x_m, y_{n-1} : y_n))$;
19 **end**
20 $I' = \mathtt{reshape}(I')$ s.t. $I' \in \mathbb{R}^{h \times w}$;
21 **return** I';

The AE set is generated by one standard AE attack method (using different AE generation methods will lead to similar results). By conducting DCT on all the 8×8 small blocks, we compare the difference of DCT frequency values (Line 10) to statistically understand the coordinates of the particular frequency coefficients which has the largest changes.

Fig. 1 (a) shows the spatial space of an AE with our DCT-based quantization. We can observe that the DC coefficient (up-left corner) is always changed the most, while low frequencies are relatively changed more than high frequencies. The quantization table is then designed according to such statistics with a principle that the frequencies that are changed more often with larger values are sensitive to DNN models. We normalize all the values within $(0, 1)$ and remap each value to the range of $(20, 100)$ (Line 15). The final Q table is shown in Fig. 1 (b).

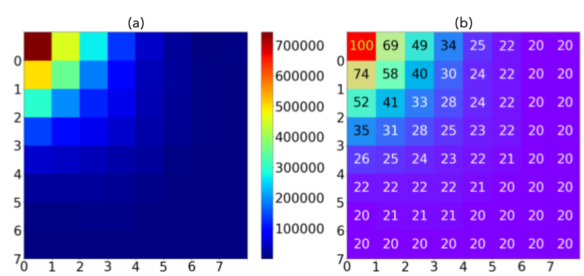

Figure 1: Frequency space statistical results of AEs (a) and the defensive quantization table (b).

ALGORITHM 2: DCT-based Quantization

Input: clean set $I^n \in \mathbb{R}^{n \times h \times w \times 3}$,
adversarial set $\dot{I}^n \in \mathbb{R}^{n \times h \times w \times 3}$,
Output: defensive quantization table Q

$Q_0 = O_{8\times 8}$;
for I_i *in* I^n **do**
  for $I_{i,channel}$ *in* I_i **do**
    $x_0 = 0, y_0 = 0$;
    $n_w = w/8, n_h = h/8$;
    $\mathcal{G}_I i = \{(x_m, y_n) | (m, n) \in \{(0, ..., n_w) \times (0, ..., n_h)\}\}$;
    for (x_m, y_n) *in* $\mathcal{G}_I i \backslash \{(x_0, y_0)\}$ **do**
      $dct_I = DCT(I_{i,channel}(x_{m-1} : x_m, y_{n-1} : y_n))$;
      $dct_{Adv} = DCT(\dot{I}_{i,channel}(x_{m-1} : x_m, y_{n-1} : y_n))$;
      $difmat = |dct_I - dct_{Adv}|$;
      $x_Q, y_Q = argmax(difmat)$;
      $Q_0(x_Q, y_Q) += 1$;
    end
  end
end
$Q = (Q_0/max(Q_0)) \times 80 + 20$;
return Q;

Step 2: Image Distortion

Lines 10-20 in Algorithm 1 illustrate the second step of our method. First, one of the four corners of the input image is randomly selected as a starting point, (e.g. the upper-left corner in Line 10). The original image is a randomly distorted grid by grid. For each grid, it will be either stretched or compressed based on a distortion level sampled from a uniform distribution $\mathcal{U}(-\delta, \delta)$ (Lines 11-12). Distorted grids are then remapped to construct a new image (Lines 13-14). This remapping process will drop certain pixels: the compressed grids will drop rows or columns of data; the stretched grids will cause the new image to exceed the original boundary, thus the pixels mapped outside of the original boundary will be dropped. For instance, in Fig. 2, the grid at the lower-right corner in stage 3 is dropped in stage 4. Then, the distorted image is reshaped to the size of the original image by cropping or padding (Line 18).

This step can drop a certain ratio of pixels and change a huge number of pixel coordinates. In our experiments, the distortion limit δ is set as 0.25. In the ImageNet dataset, each image will have around 20%-30% pixels randomly dropped and more than 90% pixel coordinates changed each time after such preprocessing operation. This can guarantee high randomness and improve the difficulty of approximation with differentiable functions, while the model can still give correct predictions.

Security Analysis

Our preprocessing function can satisfy the three requirements, with the following quantitative justification.

For usability-preserving, we measure the prediction accuracy of clean samples for $f(g(x))$. Table 1 compares our solution with prior methods. We can observe all the methods can maintain very high model accuracy (ACC). For Property 2, our solution introduces defensive quantization, which is non-differentiable.

For Property 3, we measure the uncertainty of the preprocessed output to reflect the difficulty of approximation. Specifically, given one image, we use $g(\cdot)$ to preprocess it for 100 times, and randomly select 2 outputs. We use l_2 norm and Structural Similarity (SSIM) score (Hore and Ziou 2010) to measure the variance between these two output images. Note that a larger l_2 norm or smaller SSIM score indicates a larger variance between the two images. When l_2 norm is 0 or SSIM is 1, the output images are identical and the preprocessing function is deterministic. For each preprocessing function, we repeat the above process with 1000 randomly selected input images from the ImageNet dataset. The average SSIM score and l_2 norm are listed in Table 1. Our method can outperform other defenses with a larger l_2 norm and smaller SSIM. This indicates that our preprocessing function can introduce the highest randomness to the output, as well as the highest difficulty for the adversary to approximate it with differentiable functions.

Defense	l_2	SSIM	ACC
Our method	**0.22**	**0.30**	0.95
Rand (Xie et al. 2018)	0.21	0.31	0.96
FD (Liu et al. 2019)	0.00	1.00	0.97
SHIELD (Das et al. 2018)	0.03	0.88	0.94
TV (Guo et al. 2018)	0.02	0.97	0.95
BdR (Xu, Evans, and Qi 2018)	0.00	1.00	0.92
PD (Prakash et al. 2018)	0.02	0.98	0.97

Table 1: Quantitive measurement of variance of output images introduced by various kinds of defenses.

Evaluation

Implementation

Configurations. We adopt Tensorflow as the DL framework for implementation. The learning rate of BPDA is 0.1 and the ensemble size[2] of EOT is 30. All experiments were conducted on a server equipped with 8 Intel I7-7700k CPUs and 4 NVIDIA GeForce GTX 1080 Ti GPUs.

Target Model and Dataset. Our designed method is general and can be applied to various DNN models for image preprocessing. We choose a pre-trained Inception V3 model (Szegedy et al. 2016) over the ImageNet dataset in this paper. This state-of-the-art model can reach 78.0% top-1 and 93.9% top-5 accuracy. We randomly select 100 images from the ImageNet Validation dataset for AE generation. These images can be predicted correctly by this Inception V3 model.

Metrics. We use the l_2 norm to measure the size of perturbations generated by each attack. We only accept AEs with a l_2 norm smaller than 0.05. We consider the targeted attacks that the target label is randomly generated to be different from the correct one (Athalye, Carlini, and Wagner 2018). As BPDA

[2]We tested different ensemble sizes for EOT ranging from 2 to 40. The ensemble size has little influence on ASR or ACC. With a larger ensemble size, it is possible to generate AEs with smaller l_2.

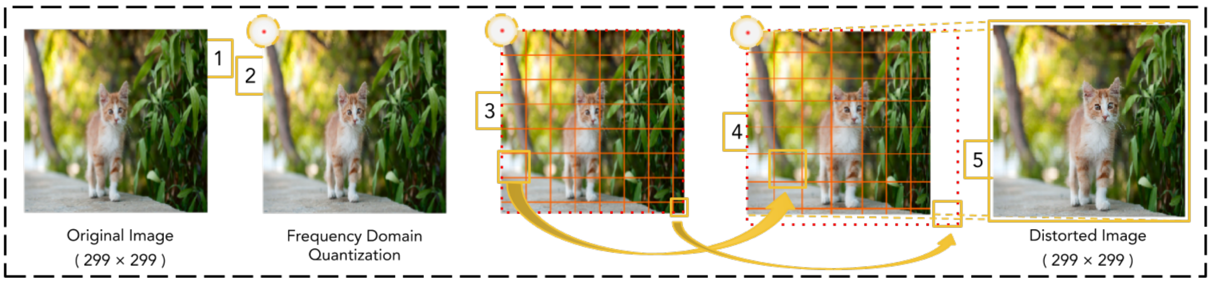

Figure 2: Processing stages in the image distortion step.

and EOT are iterative processes, we stop the attack when an AE is successfully generated (predicted as the target label with l_2 smaller than 0.05). For each round of the attack, we measure the prediction accuracy of the generated AEs (ACC) and the attack success rate (ASR) for the targeted attack.

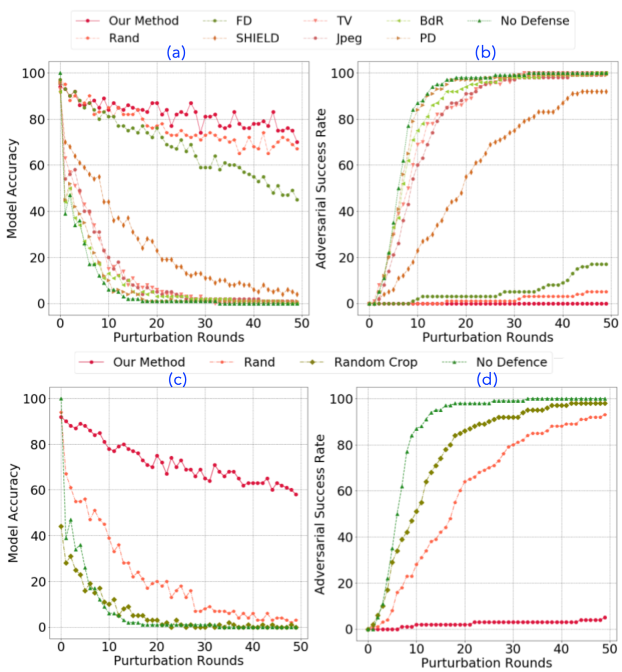

Figure 3: Defense results on BPDA: ACC (a) and ASR (b) and defense results on BPDA+EOT: ACC (c) and ASR (d).

Mitigating BPDA Attack

We first evaluate our method against BPDA. For comparison, we re-implemented 7 prior solutions including FD (Liu et al. 2019), Rand (Xie et al. 2018), SHIELD (Das et al. 2018), TV (Guo et al. 2018), JPEG (Guo et al. 2018), BdR (Xu, Evans, and Qi 2018), and PD (Prakash et al. 2018). We select these methods because they are all preprocessing-only defense which fits our defense requirements. We give a broader comparison with the defenses that need to alter the target model in Table 2 at the end of this section. Fig. 3 (a) and (b) give the ACC and ASR versus the perturbation rounds.

After 50 attack rounds, the ACC of all the previous solutions except FD drops below 5%, and the corresponding ASR reaches higher than 90%. FD can keep the ASR lower than 20% and the ACC around 40%, which is still not very effective in defending against BPDA. However, our method is particularly effective against the BPDA attack. We can maintain an acceptable ACC (around 70% for 50 attack rounds), and restrict the ASR to almost 0. RAND can also defeat BPDA with a slightly lower ACC than ours. However, it will be broken by the EOT attack, as we will show later. These results are consistent with the l_2 norm and SSIM metrics in Table 1: the randomization in those operations causes large variances for one image each time during inference which significantly increase the difficulty for attackers to generate AEs.

We continue the attack until the images with perturbations reach the l_2 bound (0.05). For our method, the adversary needs 231 rounds to reach this l_2 bound with ACC of 57% and ASR of 2%. Therefore, we conclude that our solutions can effectively mitigate the BPDA attack.

Mitigating BPDA+EOT Attack

Next, we consider a more powerful attack by combining BPDA and EOT (Tramer et al. 2020) which can defeat both shatter gradients and stochastic gradients based defenses. Here we only consider defense methods that can mitigate the BPDA attack. This gives us two baselines: Rand and Random Crop[3] (Guo et al. 2018). Fig. 3 (c) and (d) report ACC and ASR under BPDA+EOT attack.

We can observe both Rand and Random Crop fail to mitigate this strong attack: ACC drops to below 20% after 20 rounds, and ASR reaches 100% after 50 rounds. In contrast, our solution can still hold ACC of around 60% and ASR of less than 10% after 50 attack rounds. These results confirm our claims and the effectiveness of our method.

We continue the attacks until the images with adversarial perturbations reach the l_2 bound (0.05) and our method can maintain the ACC to 58% and keep the ASR to 7%.

Mitigating Adaptive BPDA Attack

In previous implementation of BPDA attack, we use a naive identity function ($g(x) \approx x$) to approximate the preprocessing function following (Athalye, Carlini, and Wagner 2018).

[3] Random Crop are not considered in the previous subsection due to its low model usability (30%-40% ACC drop).

Solutions	Requirement	Attack	#1	#2	#3	$l_\infty = 0.031$	$l_2 = 0.05$
Rand (Xie et al. 2018)	$\lozenge$	EOT	✓		✓	0%	-
PixelDefend (Song et al. 2017)	$\lozenge$, $\triangle$	BPDA	✓	✓		9%	-
Crop (Guo et al. 2018)	$\lozenge$, $\triangle$	BPDA+EOT		✓		-	0%
JPEG (Guo et al. 2018)	$\lozenge$, $\triangle$	BPDA	✓	✓		-	0%
TV (Guo et al. 2018)	$\lozenge$, $\triangle$	BPDA+EOT	✓	✓		-	0%
Quilting (Guo et al. 2018)	$\lozenge$, $\triangle$	BPDA+EOT	✓	✓		-	0%
SHIELD (Das et al. 2018)	$\lozenge$, $\triangle$	BPDA	✓	✓		-	0%
PD (Prakash et al. 2018)	$\lozenge$	BPDA	✓	✓		0%	-
Guided Denoiser (Liao et al. 2018)	$\lozenge$	BPDA	✓	✓		-	0%
ME-Net (Yang et al. 2019)	$\square$, $\lozenge$, $\triangle$	BPDA+EOT		✓	✓	13%	-
FD (Liu et al. 2019)	$\lozenge$	BPDA	✓	✓		-	10%
Our method	$\lozenge$	BPDA+EOT	✓	✓	✓	-	**58%**

Table 2: Comparisons with a broader defenses on bounded attacks. (For defense requirements, $\square$: target model modification; $\lozenge$: input preprocessing; and $\triangle$: adversarial training).

(a) (b) (c)

Figure 4: (a) Original image I_0. (b) Image produced by our method I_1, (c) Image produced by the approximated neural network I_2. $\|I_1 - I_2\|_2 = 0.22$, $\|I_1 - I_2\|_{SSIM} = 0.35$.

However, the adversary can improve the attacks by approximating the transformation with a neural network (Raff et al. 2019). Thus, we adopt this adaptive BPDA attack to evaluate our defense method. We use a 6-layer DenseNet autoencoder (same approximation attack method as (Raff et al. 2019)) to evaluate our method.

The result is that the attacker cannot find a proper approximation with such an attack. One example is shown in Fig. 4: the approximated image (c) has a large variance compared with the image preprocessed by our method (b) with l_2 norm as 0.22 and SSIM score as 0.35. Thus, such approximation cannot give a useful gradient to generate a successful AE.

We run the end-to-end attack with the trained neural network on 100 images randomly selected from ImageNet and the ASR is 0 under a maximum l_2 norm of 0.05. The average quantitative variance between the approximated image and the image processed by our method for the 100 images is as follows: l_2 norm is 0.16 and the SSIM score is 0.36.

Mitigating Standard Attacks

We also test our method against standard attacks (I-FGSM, LBFGS, and C&W). The results are shown in Table 3. Our solution has little influence on the ACC of benign samples. The ASR of those attacks can be kept as 0% and ACC can be maintained as around 90%. More details and results can be found in the supplementary material.

Attack	l_2	No Defense		Our method	
		ACC	ASR	ACC	ASR
No attack	0.0	100%	Nan	95%	Nan
I-FGSM	0.010	2%	95%	93%	0%
LBFGS	0.001	0%	100%	91%	0%
C&W	0.016	0%	100%	87%	0%

Table 3: Results of our defenses against standard attacks.

A Broader Comparison with More Defenses

We compare our solution with a broader set of defenses against bounded attacks. These methods also adopt preprocessing while some of them require model changes, e.g., model retraining (ME-Net) or adversarial training (Crop, JPEG, TV, Quilting, and ME-Net). These methods were proved to be broken partially or entirely by BPDA or BPDA+EOT in (Carlini et al. 2019).

We summarize the analytic results, experimental data as well as conclusions from literature in Table 2. The AE generation is either bounded by l_∞ (0.031) or l_2 (0.05). Even combined with adversarial training, most of them cannot provide enough robustness. We can observe that our method shows much better robustness against BPDA+EOT (ACC is as high as 58% under the l_2 bound). We also reveal the satisfactory of the three properties (#1 to #3 in Table 2) of those methods. All the defenses in Table 2 can satisfy only part of the properties. Note that ME-Net meets properties #2 and #3 but not #1, as it retrains the model with preprocessed clean samples. We conclude that our three properties are indeed an accurate indicator to reveal the difficulty of adversarial attacks.

Conclusion

We propose a novel and efficient preprocessing-based solution to mitigate advanced gradient-based adversarial attacks (BPDA, EOT, their combination, and adaptive attacks). Specifically, we first identify three properties to reveal possible defense opportunities. Following these properties, we design a preprocessing transformation function to enhance the robustness of the target model. We comprehensively evaluate our solution and compare it with 11 state-of-the-art prior defenses. Empirical results indicate that our solution has the best performance in mitigating all these advanced gradient-based adversarial attacks.

We expect that our solution can heat the arms race of adversarial attacks and defenses, and contribute to the defender's side. The proposed three properties can inspire people to come up with better defenses. Meanwhile, we expect to see more sophisticated attacks that can fully tackle our defenses in the near future. All these efforts can advance the study and understanding of AEs and DL model robustness.

References

Ahmed, N.; Natarajan, T.; and Rao, K. R. 1974. Discrete cosine transform. *IEEE transactions on Computers* 100(1): 90–93.

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *International Conference on Machine Learning*, 274–283.

Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2017. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397* .

Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2018. Synthesizing Robust Adversarial Examples. In *International Conference on Machine Learning*, 284–293.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, 387–402. Springer.

Buckman, J.; Roy, A.; Raffel, C.; and Goodfellow, I. 2018. Thermometer encoding: One hot way to resist adversarial examples .

Carlini, N.; Athalye, A.; Papernot, N.; Brendel, W.; Rauber, J.; Tsipras, D.; Goodfellow, I.; Madry, A.; and Kurakin, A. 2019. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705* .

Carlini, N.; and Wagner, D. 2017a. Magnet and" efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478* .

Carlini, N.; and Wagner, D. 2017b. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.

Das, N.; Shanbhogue, M.; Chen, S.-T.; Hohman, F.; Li, S.; Chen, L.; Kounavis, M. E.; and Chau, D. H. 2018. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 196–204.

Dong, Y.; Liao, F.; Pang, T.; Hu, X.; and Zhu, J. 2017. Discovering adversarial examples with momentum. *arXiv preprint arXiv:1710.06081* .

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* .

Gu, S.; and Rigazio, L. 2014. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068* .

Guo, C.; Rana, M.; Cisse, M.; and Van Der Maaten, L. 2017. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117* .

Guo, C.; Rana, M.; Cisse, M.; and van der Maaten, L. 2018. Countering Adversarial Images using Input Transformations. In *International Conference on Learning Representations*.

Hore, A.; and Ziou, D. 2010. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*, 2366–2369. IEEE.

Huang, R.; Xu, B.; Schuurmans, D.; and Szepesvári, C. 2015. Learning with a strong adversary. *arXiv preprint arXiv:1511.03034* .

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* .

Lee, H.; Han, S.; and Lee, J. 2017. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387* .

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

Liu, Z.; Liu, Q.; Liu, T.; Xu, N.; Lin, X.; Wang, Y.; and Wen, W. 2019. Feature distillation: DNN-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 860–868. IEEE.

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

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

Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; and Swami, A. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, 582–597. IEEE.

Prakash, A.; Moran, N.; Garber, S.; DiLillo, A.; and Storer, J. 2018. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8571–8580.

Raff, E.; Sylvester, J.; Forsyth, S.; and McLean, M. 2019. Barrage of random transforms for adversarially robust defense. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6528–6537.

Ross, A. S.; and Doshi-Velez, F. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*.

Shaham, U.; Yamada, Y.; and Negahban, S. 2018. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing* 307: 195–204.

Song, Y.; Kim, T.; Nowozin, S.; Ermon, S.; and Kushman, N. 2017. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766* .

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

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

Tramer, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347* .

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204* .

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2018. Mitigating Adversarial Effects Through Randomization. In *International Conference on Learning Representations*.

Xu, W.; Evans, D.; and Qi, Y. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.

Yang, Y.; Zhang, G.; Katabi, D.; and Xu, Z. 2019. ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation. In *International Conference on Machine Learning*, 7025–7034.