# (Compress and Restore)$^N$: A Robust Defense Against Adversarial Attacks on Image Classification

CLAUDIO FERRARI, Department of Architecture and Engineering, University of Parma/Department of Information Engineering, University of Florence
FEDERICO BECATTINI, LEONARDO GALTERI, and ALBERTO DEL BIMBO,
Department of Information Engineering, University of Florence

Modern image classification approaches often rely on deep neural networks, which have shown pronounced weakness to adversarial examples: images corrupted with specifically designed yet imperceptible noise that causes the network to misclassify. In this article, we propose a conceptually simple yet robust solution to tackle adversarial attacks on image classification. Our defense works by first applying a JPEG compression with a random quality factor; compression artifacts are subsequently removed by means of a generative model Artifact Restoration GAN. The process can be iterated ensuring the image is not degraded and hence the classification not compromised. We train different AR-GANs for different compression factors, so that we can change its parameters dynamically at each iteration depending on the current compression, making the gradient approximation difficult. We experiment with our defense against three white-box and two black-box attacks, with a particular focus on the state-of-the-art BPDA attack. Our method does not require any adversarial training, and is independent of both the classifier and the attack. Experiments demonstrate that dynamically changing the AR-GAN parameters is of fundamental importance to obtain significant robustness.

CCS Concepts: • **Computing methodologies** → **Computer vision**; **Adversarial learning;**

Additional Key Words and Phrases: Adversarial attacks, image restoration

**ACM Reference format:**
Claudio Ferrari, Federico Becattini, Leonardo Galteri, and Alberto Del Bimbo. 2023. (Compress and Restore)$^N$: A Robust Defense Against Adversarial Attacks on Image Classification. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 1s, Article 26 (January 2023), 16 pages.
https://doi.org/10.1145/3524619

## 1 INTRODUCTION

Deep Convolutional Neural Networks represent the fundamental core of most of the current state-of-the-art artificial intelligence systems in a variety of fields. Recently, it was shown that they are

This work was supported by the European Commission under European Horizon 2020 Programme, grant number 951911 - AI4Media.

Authors' addresses: C. Ferrari, University of Parma, Department of Architecture and Engineering, Parco area delle scienze 181/A, 43124, Parma, Italy; email: claudio.ferrari2@unipr.it; F. Becattini, L. Galteri, and A. D. Bimbo, University of Florence, Department of Information Engineering, Via di Santa Marta 3, 50139, Firenze, Italy; emails: {federico.becattini, leonardo.galteri, alberto.delbimbo}@unifi.it.

1551-6857/2023/01-ART26 $15.00
https://doi.org/10.1145/3524619

highly susceptible to adversarial examples [45]. These consist of slightly perturbed versions of the original examples that eventually trick the networks into outputting a wrong prediction. The peculiar characteristic of adversarial examples is that they are hardly distinguishable from their "clean" counterpart for the human eye, because they have been specifically optimized to have the minimum possible perturbation.

The most effective and challenging attacks are those that craft adversarial examples exploiting the gradient information; indeed, the gradient indicates how to perturb the input to falsify the network's prediction. To combat this class of attacks, many solutions have been proposed that aim at obfuscating the gradient information [23, 34, 48]. This goal can be achieved either by introducing non-differentiable operations [12], randomization to obtain stochastic gradients [49], or by forcing vanishing/exploding gradients [40]. However, most of these strategies have been proven vulnerable to methods that approximate the gradient [4].

To mitigate this performance breakdown, a different line of research focused on increasing robustness by means of adversarial training mechanisms, i.e., including adversarial examples directly into the training data [46]. This class of defenses represents one of the most effective against strong white-box attacks. However, one major issue of such strategies is represented by the necessity of re-training the targeted networks, which might be a non-negligible limitation in many cases. In addition, relying on some sort of adversarial process to improve the network's robustness, could lead to a poor generalization to unseen adversarial noises.

In light of the above, we propose a defense method embracing the idea of gradient obfuscation, which is independent of both any adversarial generation process and the classification network. It can be readily applied to any attacked models without needing to perform any re-training. Our main intuition is to process the input image by combining the use of JPEG compression with a randomized quality factor, and a generative model to remove compression artifacts. In the recent literature, JPEG compression has been recognized as a straightforward yet effective enough pre-processing operation to account for adversarial attacks, which has the practical advantage of being the standard for image compression. Given its potential, efforts have been put in developing learning-based solutions specifically tailored for replicating and improving the JPEG to account for its vulnerabilities e.g., [12, 35]. In our work, we propose a different solution that, instead of emulating the JPEG, tries to improve its performance by adding a generative restoration module. Despite being conceptually trivial, this solution has several advantages, the most relevant ones being the following: *(i)* it combines a non-differentiable operator and auxiliary networks, whose internal parameters change depending on the random compression factor; *(ii)* the generative model is independent of the classification network, and can be readily used without requiring any re-training of the attacked model; *(iii)* the generative model restores the quality of the compressed image, allowing us to perform multiple compression-restoration steps without impacting on the image quality and so the classification result. This also makes the gradient estimation complex, and strongly enhances the robustness to the attack, while forcing strong adversarial distortions (Figure 1). To summarize, the main contributions of this article are as follows:

- We developed a defense capable of tackling a variety of adversarial attacks, both white-box, and black-box, including the challenging BPDA [4].
- Differently from the majority of most effective defenses, our approach is independent of both the classifier and the attack. It does not require adversarial training and so knowledge of the adversarial process.
- Our method effectively combines several defense techniques i.e., input transformation, randomization, and generative restoration, in a robust framework that can be readily applied to any pre-trained classifier.

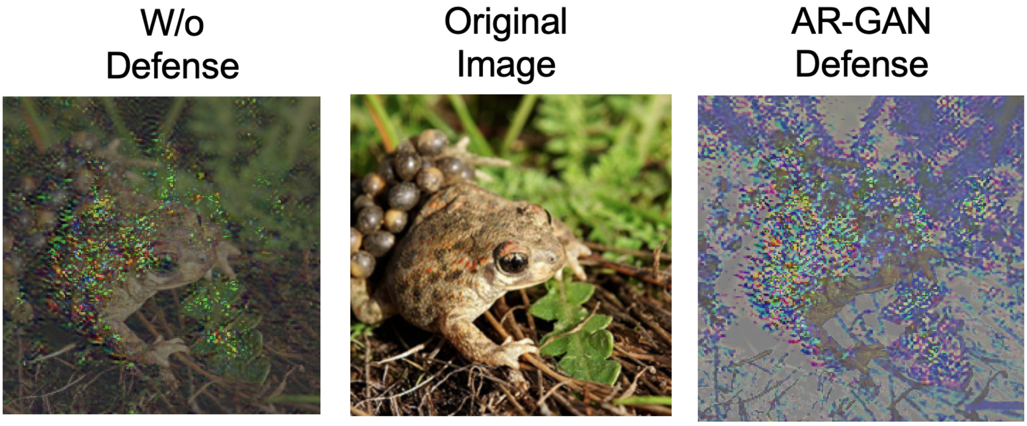

Fig. 1. A simple yet effective defense mechanism: the input image is first compressed with JPEG at a random quality factor. The latter is then restored using an ensemble of Generative Adversarial Networks (AR-GAN), each one trained for a specific quality factor. The process is repeated $N$ times so to add further complexity, while ensuring the original image is not compromised. This ultimately forces strong adversarial distortions.

## 2 BACKGROUND AND RELATED WORK

In the recent literature, a lot of effort has been put in developing both attack and defense frameworks, as they are strictly related to each other. We revise some attack and defense strategies that are most related to our proposal.

### 2.1 Adversarial Attacks

Adversarial attacks are meant to make a classifier output a wrong prediction [22]. A classifier can be denoted as a function $f(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{Y}$ that maps an input image $\mathbf{x} \in \mathcal{X}$ into a class label $y \in \mathcal{Y} = \{1, 2, \ldots, C\}$, with $C$ being the number of classes. Let $\mathbf{x}$ and $\mathbf{x}_a$ be the clean and adversarial inputs respectively, and $y^*$ the ground-truth label. Adversarial attacks simply aim at causing misclassification, and the adversarial example $\mathbf{x}_a$ should be crafted such that it is as close as possible to its clean counterpart, that is it should satisfy $\|\mathbf{x} - \mathbf{x}_a\|_p \leq \epsilon$, where $\epsilon$ is the perturbation error. Both the above can be conducted in the *white-box* and *black-box* settings.

In the former case, the attacker has full knowledge of the model, so it has access to the network architecture along with its internal parameters, the gradient w.r.t the input, and also to possible defense mechanisms. Therefore, white-box attacks commonly make use of gradient-based strategies to guide the generation of adversarial perturbations. One of the first white-box attacks is the **Fast Gradient Sign Method** (**FGSM**) [22]. This approach was further extended by [30] and [37] by refining the generation taking small iterative gradient steps via **Projected Gradient Descent** (**PGD**) or using approximated gradients with random starts. The DeepFool method [38] focuses on generating the adversarial example with the minimum possible perturbation, while **Carlini and Wagner** (**CW**) [8] proposed an approach to generate a perturbation so that the result is still a valid image but changes the classification by taking the Lagrangian form. One of the latest and most effective attacks that were proposed is the BPDA [4], which circumvents gradient obfuscation by estimating an approximation.

Differently, black-box attackers cannot rely on the knowledge of architecture and internal parameters. Thus, they need to design their attacks based on different strategies. Transfer-based approaches are those considered the most effective, which try to generate adversarial examples on a surrogate is known model using a white-box strategy, and transfer the attacks to different architectures [16, 17, 50]. Score-based attacks instead commonly rely on the output of the target architecture and use approximated gradients to generate possible adversarial examples based on the loss [27, 32, 47]. Decision-based attacks use a similar principle but the scenario is more challenging as the attacker has access only to hard-label predictions [6, 10, 18]. For a more comprehensive review, the reader can refer to [2].

### 2.2 Defense Strategies

In contrast to adversarial attacks, extensive investigations have been conducted to find effective countermeasures. By far the most effective way to protect against white-box attacks consists in performing robust *adversarial training*, which inspired a large corpus of methods [22, 36, 46, 48]. A relevant drawback is that re-training or fine-tuning is required for each attacked model. We will neglect this kind of method in our investigation as, differently, our proposed method is independent from the specific attack and does not make use of corrupted images at training time.

When dealing with gradient-based attacks, a promising direction has been identified in methods that attempt to obfuscate the gradient. A simple way to achieve this consists in randomization, which can be applied either to the input [34] or in the form of stochastic activation functions [13]. Other approaches manipulate the input image $\mathbf{x}$, so relying on an operator $g(\cdot)$ which is applied before the classification model i.e., $y = f(g(\mathbf{x}))$. Many recent works employ the JPEG compression as non-differentiable input transformation to eliminate the adversarial perturbation [11, 19, 23]. However, simple JPEG compression showed weaknesses to slightly more complex attacks [4]. To improve its robustness, some variants have been proposed. In [35], a deep network is employed to specialize the JPEG quantization step in order to mitigate accuracy reduction due to the introduction of artifacts for small quality factors. In [12], the target model is re-trained using several compressed versions of the input image, so as to "vaccinate" the model against both the attacks and JPEG compression. All these approaches resulted vulnerable to the BPDA attack [4].

Two defenses that make use of generative models are, respectively, Defense-GAN [43] and the work of Mustafa et al. [39]. The former grounds on the idea of using a GAN to remove adversarial noise from a perturbed image by finding its closest sample on the natural image manifold. This is one of the few methods partially resisting BPDA; yet, it was shown adversarial examples still exist in the GAN-generated manifold. Mustafa et al. [39] instead remove the adversarial noise by using a deep network trained for super-resolution. The network is trained on data augmented with both adversarial and super-resolved images to improve robustness. This method shares some advantages with our solution, i.e., it does not require re-training when used with different classifiers, while at the same time producing an enhanced version of the image. However, it is not really comparable with our approach since the super-resolution model is trained using an ensemble of both clean and adversarial images. Another recent approach that shares some ideas with our proposal is the method of Jia et al. [28], named ComDefend. The latter uses the same degradation-restoration principle but trains an adaptive compression model to specifically apply stronger compression to malicious features. The method is extremely robust in case the training set is augmented with adversarial images. However, if training is performed with clean images only, as we do, performance drop to some extent.

Finally, some solutions have been proposed that use ensemble strategies to achieve model-diversity. In [34], the prediction of different models are averaged after injecting random noise to each individual model. Pang et al. [40] promote model diversity by adding a regularization that encourages different predictions among different models. Similarly to these methods, we rely on an ensemble of JPEG restoration GAN models, applied in tandem with compressions at different quality factors. Ensemble defense mechanisms have been proven vulnerable to attacks that optimize with respect to the expected output **Expectation over Transformation** (**EOT**). It was also demonstrated that ensemble defenses are not much stronger than the strongest sub-component [4].

## 3 DEFENSE VIA COMPRESSION-RESTORATION

Here we first introduce the rationale behind our solution and then provide a detailed description of the method.

### 3.1 Motivation

An effective attack should be able to make the classifier predict a wrong class while simultaneously making its crafted artifacts imperceptible. Under this point of view, a defense mechanism can be considered effective whether it can prevent wrong classifications, or can make the perturbations manifest at the point they are easily recognizable. In a way, methods based on input transformations and those based on adversarial training are related with respect to the above goal. When applying an input transformation in the attempt of either obfuscating the gradient, e.g., [49], or removing adversarial artifacts, e.g., JPEG compression [23], we wish to remove or "substitute" those perturbations with some other not affecting the classifier prediction. Sticking to the JPEG example, most of the images normally used to train a deep network are indeed JPEG compressed; hence, networks gain a certain degree of robustness to such degradations. This is equivalent to mapping the image back into the learned natural image manifold [19]. Similarly, a certain degree of invariance to other image manipulations is usually achieved with data augmentation techniques. Intuitively, adversarial training mechanisms point at the same result by augmenting the training corpus with images containing adversarial artifacts. This can be viewed as the process of "enlarging" the learned manifold to include adversarial samples. However, this implies the knowledge of some adversarial generation processes, and the generalization to different such processes is not guaranteed [48]. In both the cases though, the idea is that the effort to make the attack successful implies corrupting the image with a non-negligible amount of noise. This conceptual similarity suggests that gaining protection from adversarial samples without knowledge of the adversarial process is possible [43], which would be desirable since brand new attack strategies are continuously developed. However, input transformation-based defenses have been proven vulnerable; some effective solutions that have been identified are, for example, EOT [5] or approximating the backward pass computation [4]. To counteract such strategies, strong transformations are necessary e.g., JPEG compression with very low-quality factor. Clearly, this has itself a negative impact on the classification, ultimately resulting counterproductive.

### 3.2 Proposed Method

The above considerations motivated our solution. The main idea is that of first degrading the image with a JPEG compression to account for adversarial perturbations, and then restoring it by means of a generative restoration model $g(\mathbf{x}, \theta)$ so as to recover its original quality. From a different point of view, it can be seen as a process that first uses JPEG as non-differentiable input transformation $J(\mathbf{x})$ to map the image from the adversarial manifold $\mathcal{A}$ to that $\mathcal{J}$ of compressed images i.e., $J(\mathbf{x}) : \mathcal{A} \rightarrow \mathcal{J}$; then, maps the compressed image back the natural image manifold $\mathcal{I}$ using the generative model $g(\mathbf{x}, \theta)$ that is specifically trained for that goal i.e., $g(\mathbf{x}, \theta) : \mathcal{J} \rightarrow \mathcal{I}$.

One major advantage of this solution is that we can train several, specialized, restoration models for different quality factors, each having a different set of learned parameters $\theta_q$. The advantage is two-fold: first, the probability of correctly classifying compressed and restored (non-attacked) images is higher even for lower quality factors, because each model $g(\mathbf{x}, \theta_q)$ is trained specifically for that factor. Second, we can randomly choose the quality factor, and select the most proper $g(\mathbf{x}, \theta_q)$

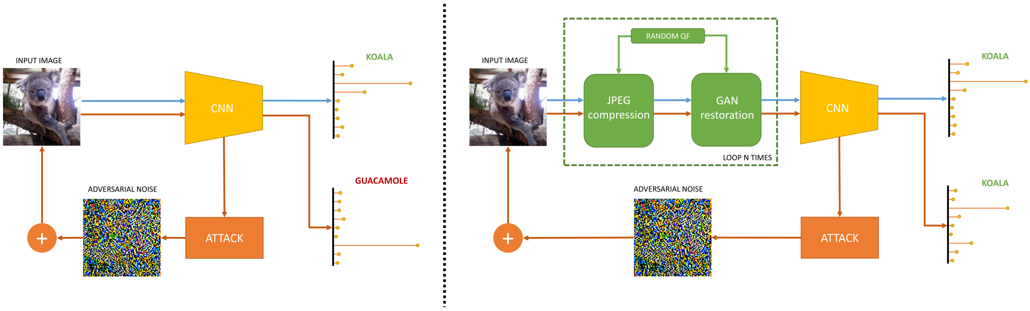

Fig. 2. Overview of the proposed pipeline. On the left, the adversarial attack process is visualized. On the right, our proposed approach. Before classification, the image goes through a sequence of JPEG compressions and AR-GAN restorations at random QFs to protect the classifier network from adversarial perturbations.

on-the-fly. In addition, operating this way it is possible to iterate the compression-restoration steps, adding a further level of complexity for the attacker. Figure 2 shows our defense strategy.

*3.2.1 Restoration GAN.* While JPEG compression is a standard image pre-processing operation, there exist several approaches to restore a compressed image [9, 14, 44] that rely on architectures based on convolutions and skip connections, thus learning layers of feature representations and propagating informations from the initial layers to the final output. In this work, we make use of the GAN-based network first proposed by Galteri et al. in [20] (AR-GAN) and extended in [21], which is specifically tailored for removing JPEG compression artifacts as it brings remarkable results in terms of perceptual quality. Differently from other solutions, this approach tries to estimate a distribution that approximates the real distribution of data and consists of two different parts, a generator network and a discriminator one. In this scenario, the networks are trained in an alternate fashion, where the objective of the generator is to produce a restored output as close as possible to the target distribution, and the discriminator is trained to recognize reconstructed images from the real ones.

The architecture of the generator is based on [24], a very common structure that comprises a sequence of residual blocks and convolutional layers. In this work, we train the generator to reconstruct the residual image rather than the whole image, that is we add a simple skip connection between the JPEG image in input and the output of the last convolutional layer. This mechanism makes the training more stable and significantly faster.

We train several AR-GANs with different JPEG quality factors, as we can observe that when we vary this parameter, different kinds of artifacts appears inside the images. For each setting, we train the model with batches of 16 $256 \times 256$ patches randomly drawn from the training images of DIV2K dataset [1] with some standard data augmentation techniques, such as image flipping and random rotation. We use Adam [29] optimizer with the standard learning rate $10^{-4}$ and momentum 0.9. Every 5 epochs we halve the learning rate and we train for a total of 20 epochs.

*3.2.2 Defense Mechanism.* We designed our defense to deal with a specific attack, that is the BPDA white-box attack [4]. BPDA is capable of effectively breaking defenses that either shatter the gradients with non-differentiable transforms, use randomization, or encourage vanishing/exploding gradients by performing optimization loops. Whereas BPDA can circumvent those mechanisms separately, using them jointly can help in preventing the attack.

*Compressed Image Restoration.* The first step of our pipeline consists in compressing the image with a JPEG operator. For each image, we perform compression using a random **quality factor**

(**QF**). However, the AR-GAN proved effective when trained to restore specific quality factors. Despite this could seem a limitation, it swings in our favor; in fact, in this way we can change the parameters of the AR-GAN at different quality factors at inference time. The AR-GANs for each QF have been trained independently, so they can be readily plugged in as needed. This is incidentally the key novel component of our approach, and has several advantages. We can change the quality factor significantly, making the approximation of the JPEG gradient more difficult while maintaining an almost unchanged classification accuracy on clean images. If the model parameters are known to the attacker as in white-box settings, it will try to use them to forge the attack. However, the parameters change continuously according to the quality factor. So, the gradient that is used at one iteration of the attack, either approximated or exact, is likely to be invalid for the subsequent iterations.

*Multiple Iterations.* The AR-GAN, which is applied after JPEG compressions, allows removing artifacts and restoring the original image quality. Thus, it is possible to perform an arbitrary number $T$ of subsequent compression and restoration steps without degrading the final image. Simultaneously, it simulates a complex, very deep network. Using a different quality factor for each image has the additional effect of changing the AR-GAN input, so that the gradient that is backpropagated changes at each iteration.

This strategy is not going to make the BPDA attack completely ineffective though; this is because the image that is restored will still lie on the GAN manifold, where it was shown adversarial samples still exist [4]. However, we force the attack to heavily corrupt the image in order to drive the network towards either predicting the desired class or making the attacked model misclassify.

## 4 EXPERIMENTAL RESULTS

In the following we report results of an extensive set of experimental validations. First, we conduct an ablation study to investigate the effects of each component of our defense, that are evaluated under the generalized BPDA attack. Then, we provide a comprehensive view by reporting results on three different white-box attacks, that is DeepFool [38], CW [8], and a modified version of BPDA equipped with EOT, that we refer to as EOT-BPDA [4]. The latter is specifically tailored to address all aspects of our defense; in particular, we use BPDA to approximate the non-differentiable JPEG operator, in conjunction with EOT to tackle the randomization applied on the quality factors. We finally explore two strong black-box attacks, namely Nattack [32] and SquareAttack [3]. We performed the evaluation using ResNet50 [24] pre-trained on ImageNet. For BPDA and EOT-BPDA, in oder to showcase the versatility of our defense, we also evaluate DenseNet [26] and MobileNet [25]. To perform the attacks, we used the implementations of the *DeepRobust* PyTorch library [31], except for DeepFool and SquareAttack, for which we used the original codes.

All the experiments are conducted under the most difficult possible setting. For white-box attacks i.e., BPDA, C&W, DeepFool, the attacks have full access to all the network parameters and gradients, including the classification module and the AR-GAN (for each QF). Black-box attacks instead do not exploit gradients; however, we provide the attacker with the model including the defense.

### 4.1 Dataset and Common Evaluation Metrics

We evaluate our defense strategy on the ILSVRC validation set [42]. Following the protocol of previous works e.g., [32, 39], we randomly choose a subset of 1,000 images (one image per class) from the validation set such that the respective classifier achieves rank-1 100% accuracy on clean, non-attacked images. As noted in [39], it would be pointless to use already misclassified images to evaluate the defense, since an attack on a misclassified image is successful by definition.

To assess the effectiveness of our approach, we evaluate it using various, standard metrics. First, we evaluate the *degradation*, that is the impact of our defense strategy on the classification accuracy when applied to clean images. To obtain a comprehensive view of the attacks and defense performance, when possible we follow the evaluation criteria suggested in the recent works of Dong et al. [15] and Carlini et al. [7]: for white box attacks, we let the attack run until a fixed perturbation budget is reached, and report results in terms of *accuracy* vs. *perturbation budget*. For black-box attacks, we fix the perturbation budget and let the attack run until a predefined amount of queries is reached.

In order to show that our defense works properly for both the widely used $\ell_2$ and $\ell_\infty$ metrics, we use both depending on the attack. Following the standard convention e.g., [4, 15], we use the $\ell_2$ norms normalized by the total number of pixels. The standard thresholds used for such metrics are, $\ell_2 = 0.005$ and $\ell_\infty = 0.031 = 8/255$. To demonstrate the robustness of our defense, we consider a less strict constraint for the attacks, and use also $\ell_2 = 0.01$ and $\ell_\infty = 0.062 = 16/255$. The detailed configurations of each attack are clarified in the following.

### 4.2 Ablation Study

In this section we analyze each module of our defense separately considering the BPDA attack as reference. BPDA is a powerful attack that estimates obfuscated gradients by approximation. These are then used to craft adversarial images in a gradient descent fashion. Even though it can be extended with EOT in the case of randomized defenses, here we chose to use the generalized BPDA attack. In fact, the goal of this ablation study is showing that the BPDA attack is strong enough to break the JPEG compression defense in all the cases, and the JPEG is followed by the AR-GAN restoration module as well. In order to make our contribution explicit, we will show that changing the AR-GAN parameters randomly depending on the current quality factor is of fundamental importance to make the defense robust.

We evaluate the robustness of our solution by applying the defense and classifying adversarial images generated at different perturbation thresholds to provide a more thorough analysis. We set the maximum perturbation to be $\ell_\infty = 0.062 = 16/255$. In particular, we let the attack run until the budget or the maximum number of iterations (200) is reached, using a step size of $10^{-3}$. We collect the results of the defense at each iteration of the attack. These experiments have been conducted using ResNet50 on a subset of 200 random correctly classified images of the ILSVRC validation set. In all the cases, the gradient of the JPEG is approximated by implementing the backward pass as the identity function, as done in the original article [4].

*4.2.1 JPEG Compression.* First, we analyze the effect of different quality factors when using the sole JPEG compression as a defense. As discussed, BPDA approximates the backward pass of the JPEG operator with the identity function. This trick is effective whenever, for a non-differentiable operator $h()$, we have $h(\mathbf{x}) \approx \mathbf{x}$. So, for breaking this attack one must ensure that $h(\mathbf{x})$ deviates largely from $\mathbf{x}$. One can achieve this by applying a strong compression (e.g., QF = 5); however, as reported in Figure 3 (left), in this case the accuracy on clean images drops dramatically. Overall, this verified that BPDA can circumvent JPEG, and that approximating the backward pass with the identity function is indeed effective.[1]

**Multiple Compression Steps:** Before analyzing the effect of introducing AR-GAN, we first aim at verifying how performing multiple JPEG compression impacts on the results. It can be observed from Figure 3 (middle) that iterating the JPEG compression $J(\mathbf{x})$ does not provide any

[1] Our results with JPEG are better than those in [4] as they use the setting of [23], where the QF is set to 75.

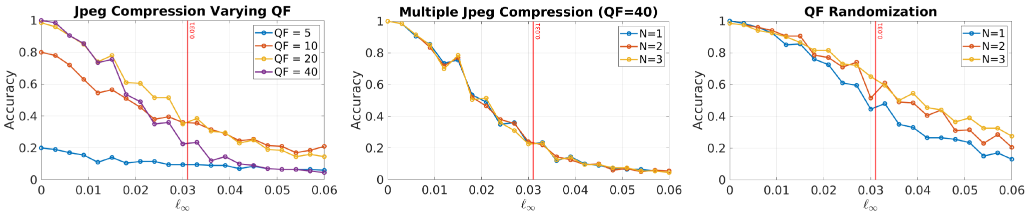

Fig. 3. Accuracy of ResNet50 under the generalized BPDA attack defending with JPEG compression at different QFs (left), performing multiple compression steps (QF = 40) (middle), and adding randomization (right). $N$ indicates the number of compression-restoration steps.

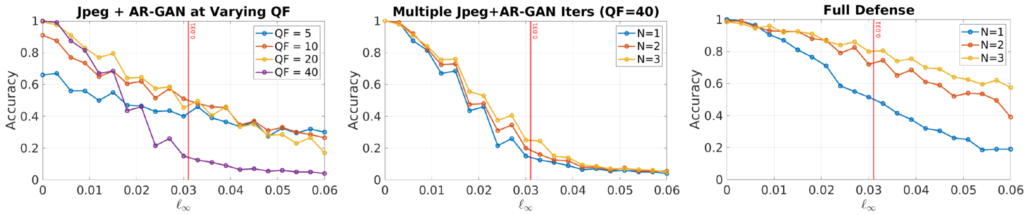

Fig. 4. Classification accuracy of ResNet50 under the generalized BPDA attack defending with JPEG followed by AR-GAN restoration for different QF (left), with multiple JPEG+AR-GAN iterations (middle), and our full defense including random quality factors (right). $N$ indicates the number of compression-restoration steps.

improvement. In fact, by applying a cascade of 3 compressions, we still have $J(J(J(\mathbf{x}))) \approx \mathbf{x}$, and the identity approximation performed by BPDA circumvents the defense.

**Randomization:** The other module of our defense consists in randomizing the JPEG quality factor at each iteration. We randomize the QF within the range $\delta_q = [20, 60]$. Differently from the previous case, adding the randomization results effective, which suggests our intuition is valid. Still, the sole JPEG does not yet provide enough robustness.

*4.2.2 AR-GAN Restoration.* In this section, we explore how restoring the compressed images with AR-GAN impacts on the defense.

**Different Quality Factors:** In Figure 4 (left) we show that for lower quality factors e.g., 5 or 10, the restoration step improves the accuracy for strong perturbations, although we still cannot achieve a perfect classification on clean images, making this strategy not useful. The curves instead drop faster for larger quality factors, despite an almost perfect accuracy on non-attacked samples. This evidences that BPDA can effectively approximate the input transformation and break the defense.

**Iterating the process:** Figure 4 (middle) shows the effect of iterating the compression-restoration steps with a fixed quality factor (QF = 40). Differently from the case of sole JPEG, repeating the compression-restoration process instead provides a slight increase of robustness. However, the defense accuracy is still low i.e., around 25% classification accuracy at $\ell_\infty = 0.031$.

**Full Defense:** Finally, Figure 4 (right) shows the effect of randomizing the quality factor for the JPEG compression and changing the AR-GAN parameters accordingly. We randomly sample the quality factors in a range $\delta_q = [20, 60]$ at each iteration of the defense. To restore the image, we use three different AR-GANs, trained with quality factors of 20, 40, and 60. At each iteration, we pick the one that is closest to the current compression factor. The complete defense with ResNet50 obtains an accuracy of 80.3% at a perturbation of $\ell_\infty = 0.031$, and of 59.4% at a perturbation of $\ell_\infty = 0.062$.

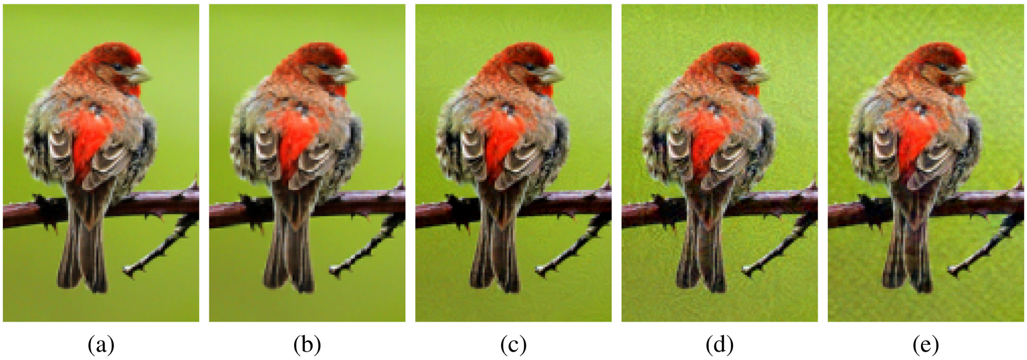

Fig. 5. Adversarial perturbation induced by BPDA. In (a), the clean image, in (b), the result of applying our full defense on the clean image. Despite the multiple compression-restoration steps, the quality of the image is maintained so that the classification is not degraded. In (c) the attacked image without defense, while in (d–e) the result of applying simple JPEG compression or our full defense, respectively. It is clear that our defence forces the attacker to inject a significant distortion.

To aid the reader with understanding the entity of the distortions, in Figure 5 we show some examples of images corrupted by the attack when different defense methods are applied. Our solution forces the attacker to inject a significantly stronger noise to make the network predict a wrong class (Figure 5(e)). In addition, we might observe an interesting side-effect; differently from the uniform noise induced by JPEG, the adversarial noise resulting from our defense forms small cross shaped patterns, that are easier to spot. Finally, in Figure 5(b) we show the effect of our defense applied to the clean image. The defense does not degrade the image quality, preserving its details and allowing us to perform a correct classification.

### 4.3 White-box Attacks

In this section, we present results on three white-box attacks, namely BPDA and EOT-BPDA [4], C&W [8], and DeepFool [38] using our full defense. All the following experiments are conducted on the full set of 1,000 images.

**EOT-BPDA.** Our defense includes the following components: non-differentiable operators, i.e., JPEG, a deterministic module with its own parameters, i.e., the AR-GAN, and a randomization criterion acting on both the quality factors and the parameters of the GAN. When using a defense mechanism that applies a random input transformation $h()$, drawn from a distribution $\mathcal{T}$, before the classifier $f()$, EOT optimizes with respect to the expectation over the transformations $\mathbb{E}_{h\sim\mathcal{T}} f(h(\mathbf{x}))$. A PGD-like attack, such as BPDA, can then be applied observing that $\nabla\mathbb{E}_{h\sim\mathcal{T}} f(h(\mathbf{x})) = \mathbb{E}_{h\sim\mathcal{T}} \nabla f(h(\mathbf{x}))$. The expectation is approximated by sampling from the distribution of $h()$.

Randomizing the quality factor can easily result ineffective against this strategy; the result of applying JPEG to an input image is deterministic once the QF is fixed. So, an attacker who applies EOT sampling from a set of quality factors can get meaningful gradients. The same applies when restoring the compressed image with the AR-GAN, as its output is conditioned on the image, and so in turn on the quality factor. However, we argue that changing the parameters $\theta$ of $g(\mathbf{x}, \theta)$ prevents an effective estimation of the EOT. A GAN $g(\mathbf{x}, \theta)$ can be seen as a process that generates samples from an underlying data distribution, that is parameterized by $\theta$. So, changing the parameters $\theta$ is equivalent to sampling from a different data distribution. This implies that EOT should approximate the expectation from multiple distributions $g(\mathbf{x}, \theta q1), \ldots, g(\mathbf{x}, \theta qn)$.

Table 1. EOT-BPDA and Generalized BPDA

| | | Random | $\ell_\infty = 0.031$ | $\ell_\infty = 0.062$ |
|---|---|---|---|---|
| EOT-BPDA | Ours - ResNet50 | ✓ | **91.4** | **77.7** |
| | | ✗ | 31.7 | 8.9 |
| | Ours - DenseNet | ✓ | **90.1** | **77.4** |
| | | ✗ | 28.4 | 8.6 |
| | Ours - MobileNet | ✓ | **87.4** | **74.4** |
| | | ✗ | 24.9 | 6.3 |
| BPDA | Bit Depth [51] | ✗ | 0.0 | - |
| | Quilting [23] | ✗ | 0.0 | - |
| | TVM [23] | ✗ | 0.0 | - |
| | DNN-JPEG [35]* | ✗ | 60.0 | - |
| | Ours - ResNet50 | ✓ | 86.5 | 60.1 |
| | Ours - DenseNet | ✓ | 87.2 | 56.9 |
| | Ours - MobileNet | ✓ | 84.1 | 57.1 |
| BPDA-ID | Ours - ResNet50 | ✓ | 89.2 | 66.3 |
| | Ours - DenseNet | ✓ | 89.5 | 62.8 |
| | Ours - MobileNet | ✓ | 88.9 | 72.5 |

Each defense (JPEG+GAN) is applied 3 times, fixing the QF at 40 or drawing it at random. Different levels of $\ell_\infty$ distortion are reported. Results for * are taken from the original article and are obtained from 100 correctly classified images out of 1,000. For BPDA-ID, we approximate both the JPEG and AR-GAN backward with the identity function. Bold values indicate the best performing method.

We perform EOT in our experiments by applying the defense for 10 repetitions at each iteration of the attack, so to average gradient information over different parameterizations $\theta$. Results are reported in Table 1. It turns out evidently that using EOT, BPDA can break our defense when using a fixed quality factor (QF = 40). Instead, when randomizing the quality factor, and so using multiple AR-GANs, the robustness increases substantially, proving our proposal is fundamental to achieve robustness to this attack. Note that we could not randomize the QF without changing the AR-GAN as each one is trained for a specific factor.

We report in Table 1 also the accuracy obtained against generalized BPDA, which proved more effective than EOT against our defense. This reinforces our claim that changing the AR-GAN iteratively leads to a difficulty in estimating the expectation from multiple distributions. Our defense cannot completely protect from BPDA though; as discussed in [4], similarly to natural images, adversarial examples can still be found in a GAN manifold; indeed, iterating the compression-restoration steps with fixed QF provided only a marginal improvement. However, when using multiple AR-GANs, finding an effective adversarial sample i.e., lying in the intersection of the different manifolds, is much more complex, and leads to significant image distorion as shown in Figure 5. Other than baseline results reported in Figures 3 and 4, we also report results of some other previous methods, namely Bit Depth reduction [51], Quilting and **Total Variation Minimization** (**TVM**) [23], and the DNN-oriented JPEG compression method of [35]. The latter is particularly related to our method as it tries to specialize the JPEG operator by increasing the quantization of malicious features. While methods based on simple input transformations are completely circumvented by BPDA, the DNN-oriented JPEG method shows a certain degree of robustness to the attack. Still, our method outperforms it, obtaining a higher classification accuracy.

Our defense mechanism tries to obfuscate the gradient by changing the underlying AR-GAN parameters at each iteration. An attacker that is aware of this strategy could try to circumvent

Table 2. (*Left*) Classification Accuracy Against the C&W Attack for Different Levels of $l_2$ Distortion, (*right*) Classification Accuracy Against DeepFool Attack

| Model | Defense | No attack | $\ell_2 = 0.005$ | $\ell_2 = 0.01$ |
|---|---|---|---|---|
| ResNet50 | No defense | **100.0** | 16.4 | 7.1 |
| | JPEG 40 | 99.4 | 52.2 | 33.9 |
| | Ours | 98.8 | **97.4** | **96.2** |
| IncResV2 | ComDefend [28]* | 77.0 | 61.0 | - |

| Model | Defense | No attack | DeepFool |
|---|---|---|---|
| ResNet50 | No defense | **100.0** | 0.0 |
| | JPEG 40 | 99.4 | 12.2 |
| | Ours | 98.8 | **88.8** |

Results of * are taken from the original article, and refer to testing on 1,000 random images from ILSVRC. (*right*) Classification accuracy against DeepFool attack.
Bold values indicate the best performing method.

it by ignoring the gradient and applying the same solution used for JPEG i.e., approximating the backward pass with the identity function. We report results for this attempt in Table 1 (BPDA-ID). As expected, in this scenario the defense is more robust, and the classification accuracy increases.

**Carlini & Wagner (C&W)** [8] is a strong iterative attack that aims at finding adversarial samples by minimizing the $\ell_2$ perturbation with respect to an auxiliary variable instead of the original image directly. A constant $c$ controls the tradeoff between perturbation and effectiveness of the attack, which is usually found by grid-search. We found the $c$ value resulting in perturbations within the range $0.005 - 0.01$ being $c = 0.1$, and used it in our experiments. Results are reported in Table 2 (left). The proposed defense demonstrates robust, comparing to the baseline. We remark that we experimented with the defense in full white-box, differently from other approaches that cannot be directly compared as they were conceived to be used in a gray-box setting [23, 39]. We instead report results obtained with the recent ComDefend [28] method. It uses a trained network instead of simple JPEG to compress the image, and then restores it with an additional trained module. In [28], the networks are trained either including adversarial samples in the training, or not. For a fair comparison, since we do not use adversarial images to train the AR-GAN, we report the results obtained in the latter case. ComDefend demonstrates fairly robust to the attack, reporting a loss of classification accuracy of 16 points. In comparison, our method only loses 1.4% accuracy with respect to non-attacked images, which demonstrates a substantial improvement.

**DeepFool** [38] aims at minimizing the $\ell_2$ between the image and the adversarial counterpart. It is specifically designed to apply the minimum possible perturbation to make the classifier commit a mistake. Because of its particular design, we cannot force it to reach a given perturbation budget. Results are reported in Table 2 (right). Our full defense attains good robustness compared to the baseline.

## 4.4 Black-Box Attacks

Our method was specifically designed to deal with the white-box attacks that makes use of gradient information to craft adversarial examples. However, following the standard practice, we also evaluate it against black-box attacks. We choose two recent state-of-the-art attacks, namely Nattack [32] and SquareAttack [3]. For both, even though they do not exploit gradient information, we provide the attacker with the model including the defense to perform the queries.

**Nattack** [32] is a state-of-the-art attack that aims at finding a probability density distribution over a small region centered around the input, such that a sample drawn from this distribution is likely an adversarial example. We set the hyperparameters suggested in [32], that is the variance of the Gaussian $\sigma^2 = 0.01$ and the learning rate $\eta = 0.008$. The maximum iterations are set to 300, and the sample size to 100. We use the $\ell_\infty$ version, with a perturbation budget of $\ell_\infty = 0.062$.

Table 3. Accuracy Against Nattack for Different $\ell_\infty$ Distortions

| Model | Defense | No attack | $\ell_\infty = 0.031$ | $\ell_\infty = 0.062$ |
|---|---|---|---|---|
| ResNet50 | No defense | **100.0** | 0.0 | 0.0 |
| | Ours | 98.8 | 97.2 | **95.8** |
| Inception | Denoiser [33]* | 79.1 | 4.5 | - |
| | Randomization [49] * | 77.8 | 3.5 | - |
| | Deflection [41]* | 69.1 | 0.0 | - |

Results in * are taken from [32].
Bold values indicate the best performing method.

Table 4. Accuracy Against SquareAttack for Different $\epsilon_2$ Distortion

| Model | Defense | No attack | $\epsilon_2 = 5$ | $\epsilon_2 = 10$ |
|---|---|---|---|---|
| ResNet50 | No defense | **100.0** | 0.7 | 0.0 |
| | JPEG 40 | 99.4 | 5.9 | 0.0 |
| | Ours | 98.8 | **98.3** | **94.4** |

Bold values indicate the best performing method.

Results in Table 3 show the effectiveness of our defense, which protects even for larger distortions with respect to other defenses.[2]

**Square Attack** [3] is a state-of-the-art, score-based attack based on a randomized search scheme, which selects localized square-shaped updates at random positions so that at each iteration the perturbation is situated approximately at the boundary of the feasible set. We run the attack implementing the $\ell_2$ versions from the original code.[3] We use the hyperparameters suggested in [3], that is $p = 0.1$, and perturbation budget of $\epsilon_2 = 5$ (note that we use the $\epsilon_2$ notation here as the attack considers the non-normalized version of $\ell_2$). For consistency with the previous settings, we also use a larger perturbation budget, that is $\epsilon_2 = 10$. We let the attack run for 5,000 iterations, i.e., queries, and collect the generated adversarial images. We then apply the defense and report the classification accuracy. From the results in Table 4, it turns out clearly that our defense is extremely robust against this attack, even for a larger distortion.

## 5 CONCLUSION

In this article, we proposed a method to protect against gradient-based adversarial attacks. Our method works by iteratively compressing the image with JPEG at a random QF, and then restoring it using an ensemble of AR-GANs. The process can be iterated without degrading the image quality thanks to the restoration module. The core of our proposal consists in using a different AR-GAN at each iteration, chosen randomly depending on the QF. This strategy allows us to change the internal parameters of the restoration model, so that the underlying data distribution changes continuously, ultimately resulting in noticeable robustness to both white-box and black-box attacks.

## ACKNOWLEDGMENTS

The Titan Xp GPUs used for this research were donated by the NVIDIA Corporation.

[2] Results indicated with * are taken from [32], and are collected on the subset of correctly classified images among 1,000 randomly selected.

[3] https://github.com/max-andr/square-attack.

## REFERENCES

[1] Eirikur Agustsson and Radu Timofte. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 126–135.

[2] Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access* 6 (2018), 14410–14430.

[3] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: A query-efficient black-box adversarial attack via random search. In *Proceedings of the European Conference on Computer Vision*. Springer, 484–501.

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

[5] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Synthesizing robust adversarial examples. In *Proceedings of the International Conference on Machine Learning*. PMLR, 284–293.

[6] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*.

[7] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. arXiv:1902.06705. Retrieved from https://arxiv.org/abs/1902.06705.

[8] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy*. IEEE, 39–57.

[9] Lukas Cavigelli, Pascal Hager, and Luca Benini. 2017. CAS-CNN: A deep convolutional neural network for image compression artifact suppression. In *Proceedings of the 2017 International Joint Conference on Neural Networks*. IEEE, 752–759.

[10] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. 2018. Query-efficient hard-label black-box attack: An optimization-based approach. In *International Conference on Learning Representation (ICLR)*.

[11] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. 2017. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. arXiv:1705.02900. Retrieved from https://arxiv.org/abs/1705.02900.

[12] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, and Duen Horng Chau. 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.

[13] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. 2018. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*.

[14] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. 2015. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision*. 576–584.

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

[16] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9185–9193.

[17] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4312–4321.

[18] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. 2019. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7714–7722.

[19] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. 2016. A study of the effect of jpg compression on adversarial images. arXiv:1608.00853. Retrieved from https://arxiv.org/abs/1608.00853.

[20] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. 2017. Deep generative adversarial compression artifact removal. In *Proceedings of the IEEE International Conference on Computer Vision*. 4826–4835.

[21] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. 2019. Deep universal generative adversarial compression artifact removal. *IEEE Transactions on Multimedia* 21, 8 (2019), 2131–2145.

[22] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. arXiv:1412.6572. Retrieved from https://arxiv.org/abs/1412.6572.

[23] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. 2017. Countering Adversarial Images Using Input Transformations. In *International Conference on Learning Representations*.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[25] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. Retrieved from https://arxiv.org/abs/1704.04861.

[26] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. 2014. Densenet: Implementing efficient convnet descriptor pyramids. arXiv:1404.1869. Retrieved from https://arxiv.org/abs/1404.1869.

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

[28] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. 2019. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6084–6092.

[29] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR* (Poster).

[30] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. arXiv:1607.02533. Retrieved from https://arxiv.org/abs/1607.02533.

[31] Yaxin Li, Wei Jin, Han Xu, and Jiliang Tang. 2020. DeepRobust: A PyTorch library for adversarial attacks and defenses. arXiv:2005.06149. Retrieved from https://arxiv.org/abs/2005.06149.

[32] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. 2019. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 3866–3876.

[33] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. 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.

[34] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. 2018. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision*. 369–385.

[35] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. 2019. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 860–868.

[36] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

[37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations*.

[38] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 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.

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

[40] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. 2019. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*. PMLR, 4970–4979.

[41] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8571–8580.

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[43] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*.

[44] Pavel Svoboda, Michal Hradis, David Barina, and Pavel Zemcik. 2016. Compression artifacts removal using convolutional neural networks. arXiv:1605.00366. Retrieved from https://arxiv.org/abs/1605.00366.

[45] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. arXiv:1312.6199. Retrieved from https://arxiv.org/abs/1312.6199.

[46] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2018. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*.

[47] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. 2018. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*. PMLR, 5025–5034.

[48] Chang Xiao and Changxi Zheng. 2020. One man's trash is another man's treasure: Resisting adversarial examples by adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 412–421.

[49] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2018. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*.

[50] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2730–2739.

[51] Weilin Xu, David Evans, and Yanjun Qi. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv:1704.01155. Retrieved from https://arxiv.org/abs/1704.01155.

Received 14 September 2021; revised 26 January 2022; accepted 8 March 2022