Countering Adversarial Examples: Combining Input Transformation and Noisy Training

Cheng Zhang, Pan Gao*
Nanjing University of Aeronautics and Astronautics
Nanjing
zhang927566, Pan.Gao@nuaa.edu.cn

Abstract

Recent studies have shown that neural network (NN) based image classifiers are highly vulnerable to adversarial examples, which poses a threat to security-sensitive image recognition task. Prior work has shown that JPEG compression can combat the drop in classification accuracy on adversarial examples to some extent. But, as the compression ratio increases, traditional JPEG compression is insufficient to defend those attacks but can cause an abrupt accuracy decline to the benign images. In this paper, with the aim of fully filtering the adversarial perturbations, we firstly make modifications to traditional JPEG compression algorithm which becomes more favorable for NN. Specifically, based on an analysis of the frequency coefficient, we design a NN-favored quantization table for compression. Considering compression as a data augmentation strategy, we then combine our model-agnostic preprocess with noisy training. We fine-tune the pre-trained model by training with images encoded at different compression levels, thus generating multiple classifiers. Finally, since lower (higher) compression ratio can remove both perturbations and original features slightly (aggressively), we use these trained multiple models for model ensemble. The majority vote of the ensemble of models is adopted as final predictions. Experiments results show our method can improve defense efficiency while maintaining original accuracy.

1. Introduction

Adversarial attack presents a major challenge for the prevalent deep neural networks used for image classification and recognition [37]. Several countermeasures have been proposed against adversarial examples, mainly including model-specific hardening strategies and model-agnostic defenses. Typical model-specific solutions like "adversarial training" [17, 27, 34, 33, 29] can rectify the model parameters to mitigate the attacks by using the iterative retraining procedure or modifying the inner architecture. However, it is generally believed that, network's architectural elements would matter little unless making them larger and deeper in improving adversarial robustness. In contrast, model-agnostic solutions like input dimension reduction or direct JPEG compression [8, 4], become more feasible and practical, which attempt to remove adversarial perturbations by input transformations before feeding them into neural network classifiers.

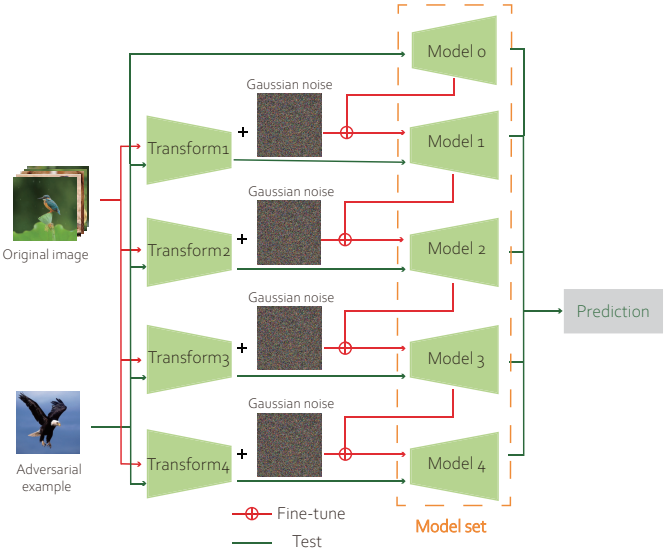


Figure 1. Overview of our combination method. Different transform modules represent different level of compression for the original images. The initial model, i.e., model 0, is the pre-trained model on benign images of ImageNet dataset, such as ResNet or Inception-v4.

For mitigating adversarial examples, standard JPEG compression has been explored in [8, 4]. But, in these works, they have shown that JPEG cannot achieve a good balance between countering adversarial examples and classifying benign images, *i.e.*, lower quality factor (QF) for JPEG compression achieves better defense efficiency but

*Corresponding author

causes a significant feature loss on benign images. To resolve this problem, we first optimize the JPEG based transformation process in this work, to improve defense efficiency against adversarial examples and maintain classification accuracy on benign images. Firstly, we analyze the distributions of the DCT coefficients for 6 color channels (i.e., R, G, B, Y, Cb, Cr) on both benign images and polluted images to find out adversarial perturbations' distribution at all 64 frequency bands. With the frequency analysis, we then divide the frequency coefficients into two types, i.e., the original favored (OF) band and the adversarial favored (AF) band. Finally, the corresponding defensive quantization parameters for these two bands are derived, where the number of the DCT coefficients that should be included in each type of band is jointly optimized.

With the purpose to further achieve both accuracy and robustness, we fine-tune the model with our own pre-processed images. Firstly, as a data augmentation, training images will be compressed using our compression algorithm. This idea shares the similar spirit of the image cropping-rescaling method proposed in [13], in which, the neural network re-trained on randomly cropped-rescaled images yields better performance than other input transformations. Secondly, Gaussian noise is added to the compressed images to mimic the adversarial perturbations (detailed in Section 3.4), which is based on the fact that strong adversaries are not necessarily needed during adversarial training as demonstrated in [33]. However, as the model is commonly noisy trained with compressed images of a certain level of quality, there still exists unavoidable trade-off between robustness and accuracy. To achieve the best balance, we generate a number of classifiers by fine-tuning the neural network using a variety of degrees of compression quality images during aforementioned pre-processing. After having obtained a set of classifiers, the final prediction value if chosen to be the label maximizes the average confidence (i.e., the output of Softmax layer) of each classifier.

Figure 1 shows an overview of our overall method, where we combine the input transformation and the noisy training. The pre-processing is implemented by using a proposed compression followed by adding the general Gaussian noise. The model set is realized by fine-tuning the models with compressed images at different compression level. The initial model used for fine-tuning is the pre-trained model on benign images. The models retrained with different compression levels are ultimately utilized together in an ensemble defense. Experimental results demonstrate the defense efficiency and the legitimate classification efficiency of the proposed algorithm against a variety of adversarial examples in the gray-box, black-box and white-box scenarios. The implementation code of the algorithm proposed will be made publicly available.

2. Related Works

2.1. Adversarial Attacks

One of the first and simple but quite effective attack is the **Fast gradient sign method** (FGSM [12]). It simply takes the sign of the gradient of loss function J (e.g., cross-entropy loss) w.r.t the input image x and multiply with magnitude ϵ as perturbations,

$$x_{adv} = x + \epsilon \cdot sign(\bigtriangledown_x J(\theta, x, y)). \quad (1)$$

where θ is the parameters set of neural network and y is the ground truth label of x. The parameter ϵ is the magnitude of perturbation which controls the similarity of adversarial examples and original image. By trying to find a high success rate adversarial example but has as small dissimilarity with original image as possible, [16] proposed an iterative version of FGSM, **I-FGSM**. It iteratively applies FGSM in every iteration and clips the value to ensure per-pixel perturbation below attack magnitude,

$$x_{adv}^{(i)} = Clip_{x,\epsilon}\{x_{adv}^{(i-1)} + \epsilon \cdot sign(\bigtriangledown_x J(\theta, x, y))\}. \quad (2)$$

Deepfool [25] assumes neural network as a linear classifier, and then projects x onto a linearization of the decision boundary. The distance from x to decision boundary is computed as perturbation. Since the assumption aforementioned is overly simplistic, Deepfool keeps iterating until it finds a success adversarial example. **Carlini-Wagner's L_2 attack(C&WL_2)** [2] is an optimization-based attack that adds a relaxation term to the perturbation minimization problem based on a differentiable surrogate of the model. The optimization problem are minimizing

$$\parallel x - x' \parallel + \lambda_{max}(-\kappa, Z(x')_k - max(Z(x')_{k'} : k' \neq k)) \quad (3)$$

where κ controls the confidence of predictions made by neural network, and $Z(\cdot)_k$ represents the logits value (input for softmax layer) corresponding to class k. **BPDA** [1] recurrently computes the adversarial gradient after applying defense:

$$x_{adv} = Clip(x + \epsilon \cdot sign(\bigtriangledown J_{\theta,Y}(Def(x)))) \quad (4)$$

where J represents the loss function of classifier and $Def()$ is the used defense method.

2.2. Model-Agnostic Defenses

Recently, [22] developed a deep neural network favorable JPEG-based image compression framework called Feature distillation which preprocesses images using modified quantization table in JPEG. [13] combined input transformations like TVM [19], image quilting [9] and image cropping. However, both TVM and image quilting are time-consuming. [8] empirically reported that JPEG compression can reverse only small perturbations, but the reason

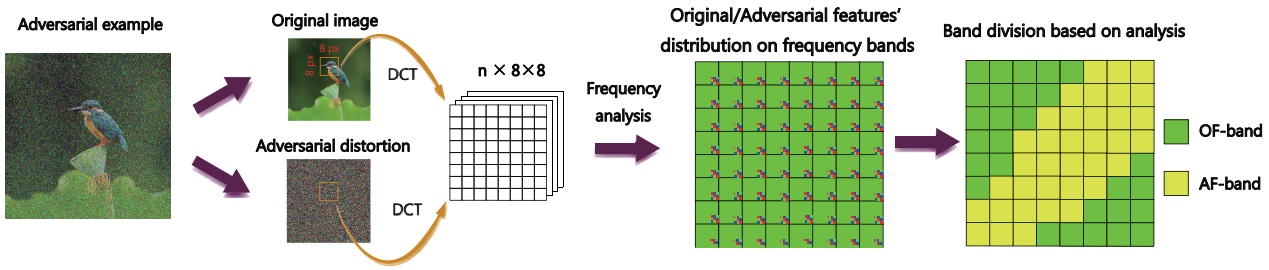


Figure 2. Analysis of original and adversarial features' distribution at frequency domain and band division.

behind is not explained. [4] proposed a JPEG compression based ensemble method, called "vaccinating", to mitigate adversarial attacks by voting the results based on a variety of compression rates. [21] proposed an adversary-concerned JPEG compression framework, which modified quantization table to eliminate the malicious perturbations. However, it analyzed the DCT coefficients of all channels as a whole, and used a heuristic design flow for quantization. In this work, we consider an analysis of the frequency of both the original images and perturbations on different channels, and design a more detailed optimization solution.

2.3. JPEG Compression

JPEG [32], [20] is one of the most popular lossy compression standards for digital images. First, the image is converted from RGB into a different color space called YCbCr. After color space transformation, each channel is split into 8×8 blocks. Each block of each component is converted to a frequency-domain representation, using discrete cosine transform (DCT) to obtain 64 coefficients. The coefficients are quantized by a quantization table [32]. The table is designed to preserve the low-frequency components and discard high-frequency details, because human visual system (HVS) is less sensitive to the information loss in high-frequency bands. After quantization, all the quantized coefficients are ordered into the zig-zag sequence. The differential coded DC and AC coefficients will be further compressed by entropy coding. A reversed procedure of aforementioned steps can decompress an image.

3. Our Proposed Approach

In the following, we firstly analyze the DCT coefficient distribution on the respective three channels of the two color formats, i.e., RGB and YCbCr. Then, we derive the best quantization steps for defending perturbation. Finally, we propose a noisy-based adversarial training based robustness reinforcement method. To further enhance defense efficiency, we propose an ensemble method.

3.1. Frequency Analysis

Standard JPEG compression is based on two assumptions, i.e., HVS are more sensitive to low frequency components of DCT coefficients and brightness channel than high frequency component and chrominance channels, respectively. However, neural networks learn features in a quite different way. As shown in the prior work [28], the image feature is highly related to the standard deviations ($\delta_{i,j}$) of the coefficient, and a larger $\delta_{i,j}$ means more features in band (i, j) can be learned by NN classifiers. Inspired by this, we analyze the distribution for each frequency component of the channels of the RGB and YCbCr.

Our frequency analysis is illustrated in Figure 2. We select 10k original images from ImageNet randomly and then use them to generate 10K adversarial examples. Each of adversarial examples can be split into original image and adversarial distortion which would be represented as both the RGB and YCbCr spaces. Therefore, each image (original image or adversarial distortion) can be split into 6 color channels (i.e., R, G, B, Y, Cb, Cr). Each separate channel is then partitioned into 8×8 blocks, followed by a block-wise DCT. The standard deviations of each frequency channel are computed. This statistical information can tell us the distribution of the original/adversarial features on frequency bands. Based on that distribution, we can optimize quantization table.

As our experimental results in Figure 3 show, RGB space has almost the same $\delta_{i,j}$ for the frequency components among the three channels. However, Cb, Cr channel in YCbCr space has a significantly less $\delta_{i,j}$ in comparison to each RGB channel, though Y channel aligns with RGB channel in terms of $\delta_{i,j}$. That is caused by the down-sampling step in standard JPEG process. According to previous conclusion that higher $\delta_{i,j}$ means more features, we consider the RGB to YCbCr color space transformation, would induce feature loss in the following quantization step so that there would be an accuracy decline when classified by neural network. To validate the hypothesis above, we use the same quantization table for compression in RGB and YCbCr domains. Results are shown in Figure 4. Though the defense efficiency is slightly worse when QS (quantization step) is less than 20, as QS increases, directly doing DCT in RGB domain not only can preserve a higher accuracy on benign images but also have nearly 2 times larger defense efficiency than YCbCr domain. In addition, another advantage of employing RGB space is that, since RGB space has

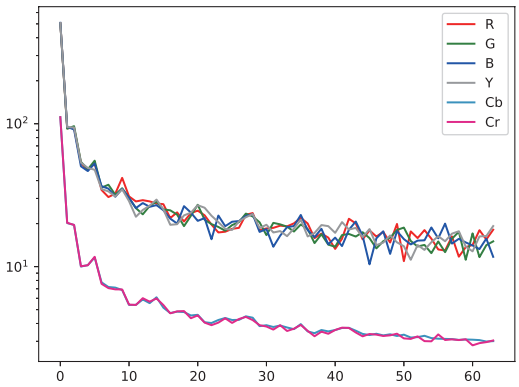


Figure 3. Statistical information about frequency of 64 component in different color channels. The y-axis represents the standard deviation ($\delta_{i,j}$) and the x-axis represents the 64 components.

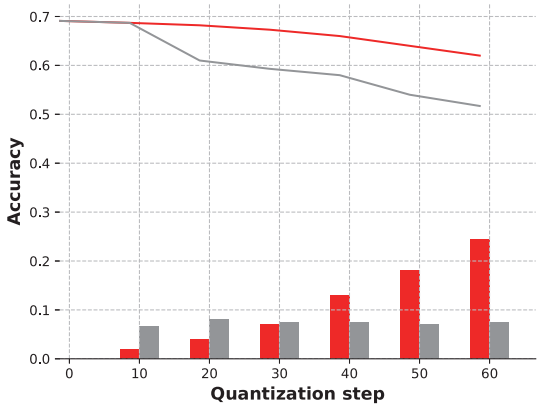


Figure 4. Comparison between doing quantization process at the RGB and YCbCr domains. The lines represent the accuracy on preprocessed original image (red for applying quantization at RGB domain, gray for YCbCr), and the histograms represent the accuracy on preprocessed adversarial examples (red for applying quantization at RGB domain, gray for YCbCr). Here, adversarial examples are generated using FGSM (ϵ = 0.008) and tested on ResNet-50.

the same statistical characteristics for the respective three channels, it allows us to design only one quantization table for the lossy coding of the DCT coefficients.

3.2. Quantization Table Design

In this section, we redesign the quantization table for the purpose of removing the perturbation in the image.

Step 1. As also shown in [28], the unquantized coefficients can be approximated as normal distribution with zero mean but different standard deviations ($\delta_{i,j}$). Since the adversarial example is a linear combination of two normal-distributed variables, i.e., original DCT coefficient and the perturbation, its δ can be also considered as a linear combination of the δs of these two variables. Since $\delta_{i,j}$ of adversarial example is a linear summation of original image and

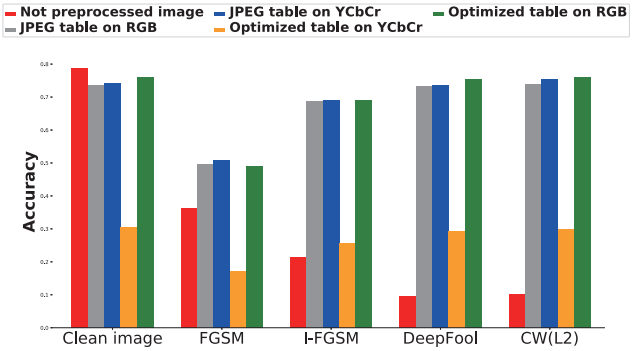


Figure 5. Ablation study on how the quantization table and color transform contribute to robustness. Clean images are from ImageNet validation set.

perturbations, this statistical information can tell us the relationship between each frequency component and features. Assume that $\delta^b_{i,j}$ indicates $\delta_{i,j}$ of DCT coefficients for the benign image, and $\delta^a_{i,j}$ indicates $\delta_{i,j}$ of DCT coefficients for adversarial distortions. We then import $\delta^{'}_{i,j}$=$\delta^a_{i,j}$/$\delta^b_{i,j}$ to characterize correspondence between two contrary features for the frequency component. The $\delta^{'}_{i,j}$ is thus a measure of how many perturbation is introduced into the current considered frequency component, i.e., coefficients of those frequency components with larger $\delta^{'}_{i,j}$ contains greater proportion of adversarial features. With the $\delta^{'}_{i,j}$ calculated for all the coefficients, we sort the DCT coefficients in a way that the associated $\delta^{'}_{i,j}$ increases. After the ascending sorting for the DCT coefficients, we can partition 64 coefficients to OF (origial-favored) band and AF (Adversarial-favored) band. The number of DCT coefficients that is included in each type of band is determined by the optimization algorithm, which will be detailed in the following.

Step 2. For these two types of bands, we use different quantization parameters to balance the defense efficiency and the testing accuracy of the legitimate examples. When adversarial distortion ρ_x is added with intensity ϵ ($\epsilon = \rho_x/255$) into original 8×8 block x, the formed adversarial block can be represented as:

$$x_{adv} = x + \rho_x = (x/255 + \epsilon) \times 255 \tag{5}$$

Input block will be linearly separated by 2D-DCT transformation as:

$$DCT(x_{adv}) = DCT(x) + DCT(\rho_x) = C_x \cdot B + C_{\rho_x} \cdot B \tag{6}$$

In (6), C_x and C_{ρ_x} are DCT coefficients, and B denotes the DCT basis function. The maximum magnitude of C_{ρ_x} can be calculated by the sum of all 64 frequency components and each term is bounded by $cos(\theta) \cdot \epsilon$, where θ is dependent on the DCT basis generating function [32]. Thus we have $-8 \cdot \epsilon < C_{\rho_x} < 8 \cdot \epsilon$. Generally speaking, the quantization and dequantization steps in JPEG provides an opportunity for filtering adversarial perturbations. If we want to

eliminate the perturbation C_{ρ_x}, we need a QS to satisfy:

$$Round((C_x + C_{\rho_x})/QS) \times QS \approx Round((C_x)/QS) \times QS \tag{7}$$

Noting such a process cannot fully recover the original value, even though the perturbation is removed. Since $(Round(C_x)/QS) \cdot QS \neq C_x$, let $\eta = Round(C_x)/QS$, we can get $\hat{\theta} = |C_x - \eta \cdot QS|$, which is the remainder of C_x/QS. With these notations, we have the following:

$$Round(C_x/QS) = (C_x - \hat{\theta})/QS$$

$$Round((C_x + C_{\rho_x})/QS) = Round((C_x - \hat{\theta} + \hat{\theta} + C_{\rho_x})/QS)$$

$$Round((C_x + C_{\rho_x})/QS) = \eta + Round((\hat{\theta} + C_{\rho_x})/QS) \tag{8}$$

As can be observed from the above derivation, either too small or too large QS can induce large rounding error in the quantization and dequantization processes. Thus, in order to make the second term in the right-hand side of (8) equals zero, we choose QS=16·ϵ, which is used for quantizing the OF bands to preserve original features.

Step 3. For determination of the number of OF bands in the partition process and the QS for AF bands, we develop an iterative algorithm. In general, more DCT coefficients included in the OF bands can lead to better classification accuracy on the benign images. But, it will result in degraded defense efficiency for the polluted images. To better balance the classification accuracy and the defense efficiency, we partition the DCT coefficient-sorted block according to the right diagonal line following the zigzag scanning order starting from the DC coefficient. The top-left corner DCT coefficients belong to OF bands, while the remaining are AF bands. As there are 15 diagonal lines for a 8×8 block, we have a total of 15 different partition patterns (P_Pattern). To determine which P_Pattern is appropriate for against adversarial attacks, we test the defense efficiency and the classification efficiency for each P_Patterns. During each test, we also conduct an exhaustive search of the QS for AF (QS_AF) bands given the QS for OF (QS_OF) being $16 \times 255\epsilon$. The search range of the QS_AF is from 1 to 121 with step size of 5. We repeat the P_Pattern search and the QS_AF search until the defense efficiency converges and the decrease of the classification efficiency on the benign image is no more than 1%. The detailed implementation of the proposed iterative algorithm is summarized in Algorithm 1. Based on our quantization table optimization algorithm, we adopt 16 as QS_OF for OF frequency bands, 50 as QS_AF for AF frequency bands.

3.3. Ablation Study

As the optimized quantization table is designed, we conduct the ablation survey to figure out how skipping color transform and optimized quantization table contributes to the robustness. We consider 4 combinations: (1) traditional JPEG (original quantization table + color transform), (2) original quantization table + skipping color transform, (3) optimized quantization table + color transform, (4) optimized quantization table + skipping color transform. Four attacks aforementioned are adopted for testing on Inception-v4 [30]. As Figure 5 shows, if we maintain the quantization table of traditional JPEG, skipping the color transform process can outperform on both benign image and adversarial examples. When applying our optimized quantization table, accuracy drop drastically with color transform. However, our optimized quantization table utilized on RGB space has better defense efficiency than the first three methods (except against FGSM) while achieving higher accuracy on benign images. Since we design an identical optimized table for all three channel instead of different table for each channel in traditional JPEG, we use our quantization table in next subsection.

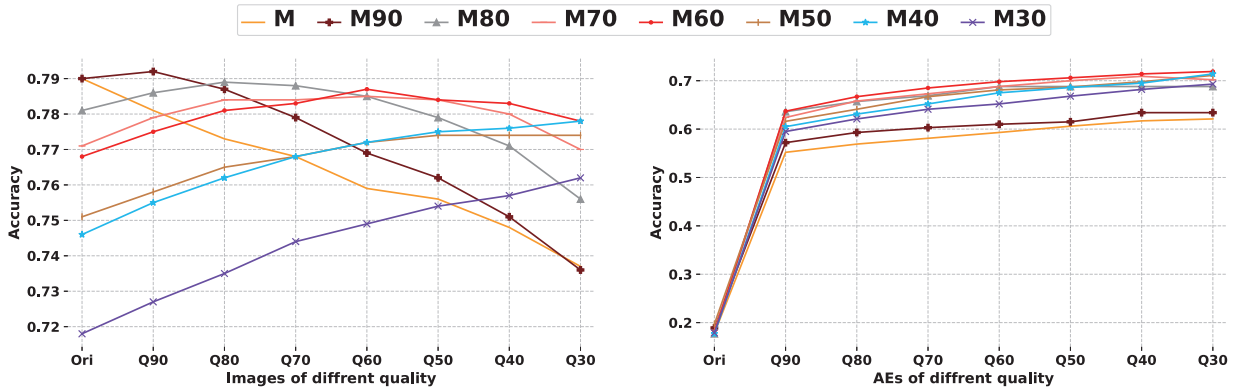
Algorithm 1: Iterative algorithm to optimize QS

```
QS_AF = 1#Initial QS for AF bands
QS_OF = 16#QS for OF bands, we adopt ϵ=0.004
for k in range(1,15):
  Initialize P_Pattern[k]
#select 1000 benign images
X_ben = Dataset(ImageNet,1000)
#generate 200 AEs from correctly classified images
X_adv = Adv(X_benign,200)
 for QS_AF in range(1,121,5) and k in range(1,15):
      X_ben_de = update_jpeg(X_ben, k, QS_AF, QS_OF)
      X_adv_de = update_jpeg(X_adv, k, QS_AF, QS_OF)
      model.predict(X_ben_de, X_adv_de)
#Accuracy decline on benign images
      Acc_dec = Cal_acc_dec()
#Defense efficiency on adversarial examples
      Def = Cal_def()
 [QS_AF*, k*] = argmax_[QS_AF, k](Def)
                    s.t.  Acc_dec<1%
Return P_Pattern[k*], QS_AF*
```

3.4. Noisy Training

At present, the vanilla adversarial training is that, in each iteration, the network is trained with those adversarial examples generated online [35, 7]. In this way, the trained network can defend against the attack of the sample to a certain extent. However, there are two disadvantages in traditional adversarial training: firstly, there is no theoretical guarantee for this kind of defense, that is, we do not know whether the attacker can design more intelligent attack methods to bypass this defense. Secondly, this kind of adversarial training can be extremely time-consuming since the calculation of adversarial examples is quite complicated.

More recently, adversarial training [33] proposed to ac-



(a) (b)

Figure 6. The performance of each individual model, each model is fine-tuned using certain but different quality images: (a) Benign images from ImageNet validation set are compressed at different quality and then tested on each model. (b) Adversarial examples (AEs) generated using I-FGSM ($\epsilon = 0.008$) are compressed at different quality and then tested on each model. Here x-axis represents images at 8 different compression qualities, and y-axis represents the test accuracy.

Table 1. Performance of image classifiers trained with adversarial training and proposed method.

Method	ϵ	Epochs	Standard acc.	PGD + 1 restarts	PGD + 10 restarts	Total time(hrs)
Fast [33]	2	15	60.90%	43.46%	43.43%	12.14
Free [29]	2	15	64.37%	43.31%	43.28%	52.20
Proposed	2	14	65.10%	40.06%	40.01%	11.51
Proposed	2	28	63.10%	42.23%	42.17%	22.11
Fast [33]	4	15	55.45%	30.28%	30.18%	12.14
Free [29]	4	15	60.42%	31.22%	31.08%	52.20
Proposed	4	14	64.02%	28.43%	28.40%	11.51
Proposed	4	28	63.01%	30.89%	30.85%	22.11

celerate training from three aspects, (1) starting from a non-zero initial perturbation, regardless of the actual initialization. (2) one of the reasons why FGSM and R+FGSM [31] may have failed previously is due to the restricted nature of the generated examples (each dimension is perturbed only by either 0 or $\pm\epsilon$). (3) defenders do not need strong adversaries during training. Thus, fast adversarial training (FAT) is designed to modify the simple FGSM (instead of more complicated PGD [24]) adversarial training by replacing its zero initial with random initial. However, since we do not need strong adversaries, why not to use a simpler Gaussian noise as adversaries? Also, [11, 38] have suggested that Gaussian data augmentation could supplement or replace adversarial training and proposed using a network's robustness to Gaussian noise as a proxy for its robustness to adversarial perturbations. Inspired by this, we proposed a noisy training based method. With just an adding operation (without computation of gradients), we consider our training computation complexity is a little less than that of [33]. Specifically, noisy based adversarial training combined with our method uses compression as the regularizer for original cost function, which is as follows

$$J'(\theta, x, y) = \xi J(\theta, x, y) + (1 - \xi) J(\theta, x_q, y) \quad (9)$$

where $\xi \in [0, 1]$. The x_q represents compressed image at quality q of original image x adding Gaussian noise ρ_g. In our experiments, we use $\xi = 0.9$ to achieve diversity of networks' parameters [26]. Note that the proposed noisy training is different from randomized smoothing proposed in [18, 3, 39], where Gaussian noise is injected into original images rather than the compressed images as done in this paper. As will be verified by our experiments later, combining noisy training with compression based transformation can provide both L_2 and L_∞ robustness.

We adopt the method in [4, 5] to fine-tune the network using the pre-trained weights for faster convergence. We use stochastic gradient descent (SGD) with a learning rate of 0.005, with a decay of 94% over 14 / 28 epochs. Training images are compressed at qualities from 90 to 30 (with a step size of 10). Hence we get a total of 8 network models sharing the same architecture but having different weights ($M, M_{90}, M_{80}, M_{70}, M_{60}, M_{50}, M_{40}, M_{30}$), where M represents the original network model and M_x the network model fine-tuned with images of compression quality of x. While fine-tuning, the initial weights of M_x is from the intermediately proceeding one, i.e., M_{x+10}, and the initial weights of M_{90} is from the initial pre-trained model M. For an ablation study, our proposed noisy training is compared with Fast [33] and Free [29] training in terms of computation efficiency and robust accuracy, and the results are

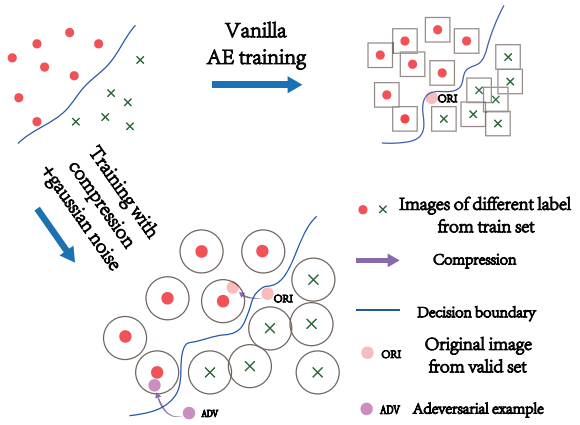


Figure 7. Comparison between vanilla adversarial example (AE) training and the proposed training algorithm.

listed in Table 1. We can see the proposed method maintains higher accuracy on benign image (Standard acc.) than Fast training and has similar robustness to PGD attack. Compared with Free training, we achieve the same level of performance while consuming much less time.

Further, we test the performance of each model in our model set. Figure 6 shows the results of classification accuracy on benign images and adversarial examples. In Figure 6 (a), the results on benign images show fine-tuning with certain compression quality images can indeed improve network's accuracy. However, this improvement is limited to certain compression quality that was used for training, e.g., M_{90} achieves best performance on images at quality 90 (Q90). Despite those network models fine-tuned in later stage (M_{50}, M_{40}, M_{30}) have been learned from high quality images, those models do not perform well on high quality images. For example, the M_{50} is indirectly fine-tuned from the M, but M_{50} performs bad on original images. In Figure 6 (b), as the compression ratio increases, fine-tuned models with compressed images all can achieve better performance on defending adversarial examples than the original model M. But defense efficiency becomes better when compression quality is lower. So, using one single network to model seems to be difficult to maintain good performance on both high and low quality images. To save inference time, finally we choose $M, M_{90}, M_{70}, M_{50}, M_{30}$ to consist our network model set which is used as an ensemble for defense.

Finally, we use a example, i.e., Figure 7, to geometrically illustrate the key difference between the vanilla adversarial training and the proposed training method. As studied in [33], vanilla adversarial training can lead to "catastrophic overfitting", which improves robustness, however, at the cost of accuracy on original test images. This issue is shown in the top-right of Figure 7. In contrast, our proposed training method can improve both accuracy on benign images and defense efficiency on adversarial examples as verified in Figure 6. This means that, our proposed method can correctly recognize more image examples. That is, some original images and adversarial examples could be pushed back to correct decision region, as shown in the bottom of Figure 7. It should be noted here, although combined with compression and Gaussian nosie, our proposed method may also have the problem of overfitting.

4. Performance Evaluation

4.1. Experiment Setup

Our experiments are conducted on the Tensorflow DNN computing framework [23]. We choose the large-scale ImageNet [6] dataset as our benchmark to better illustrate the proposed method. Five types of adversarial example attacks including FGSM [12], I-FGSM [16], Deepfool [25], C&W(L_2) [2] and BPDA [1], are simulated using popular cleverhans package on validation set of ImageNet (50k images) in our experiment for evaluating the efficiency of each defense. Feature Distillation (FD) [21] and Total variance minimization (TVM) [13] are selected as our defense benchmarks to compare with our proposed method. To the best of our knowledge, these two methods yield the state-of-the-art performance on countering the adversarial attack among model-agnostic methods. We adopt the ResNet-v2-50 [14, 15] for Gray-box and Black-box setting, Inception-v4 [30] for White-box setting.

4.2. Gray-box Scenario

In this scenario, the adversary has the access to the model's architecture and parameters but is unaware of defense strategy. We use the parameters of the model which is trained with benign images to generate adversarial examples. As Table 2 shows, when facing with benign images, the proposed method can maintain accuracy at the same level as original model. Traditional JPEG, FD [21] and TVM [13] all introduce 5% drop on benign image (No attack). When attacking, Deepfool and C&W are both very effective, because they can achieve beyond 90% success rate while maintaining a much lower L_2 dissimilarity with original image. However, these two low intensity but effective attacks can be defended very largely by all four transformation defense. Moreover, our proposed methods can achieve the defense efficiency as high as benign images' accuracy. When combating with FGSM & I-FGSM, which generate adversarial examples with stronger perturbation intensity, our proposed methods achieve around 10% higher performance than other three methods.

4.3. Black-box Scenario

In this scenario, the adversary is unaware of neither the architecture nor the parameters about the network. In this setup, we intend to evaluate the transferability of at-

Table 2. Summary of model accuracy (in %) for all defenses in Gray-box and Black-box scenarios.

Gray-box	No attack	FGSM [12]			I-FGSM [16]			Deepfool [25]	C&W(L_2) [2]
		$\epsilon=2$	$\epsilon=4$	$\epsilon=8$	$\epsilon=2$	$\epsilon=3$	$\epsilon=4$		
No Defense	65.4	17.5	14.4	13.2	15.1	12.4	9.5	9.7	9.3
JPEG (75) [32]	60.5	41.1	29.4	21.1	57.1	51.6	50.0	60.0	60.3
FD [21]	59.5	44.1	30.5	21.4	56.5	54.4	53.3	58.8	58.7
TVM [13]	58.2	44.7	33.7	22.7	55.1	51.9	48.8	57.6	58.8
Proposed	**66.5**	**56.4**	**43.1**	**28.2**	**68.0**	**65.2**	**64.8**	**67.1**	**65.4**

Black-box	No attack	FGSM [12]			I-FGSM [16]			TI-FGSM [10]		DI-FGSM [36]	
		$\epsilon=2$	$\epsilon=4$	$\epsilon=8$	$\epsilon=2$	$\epsilon=3$	$\epsilon=4$	$\epsilon=2$	$\epsilon=4$	$\epsilon=2$	$\epsilon=4$
No Defense	65.4	45.0	34.6	25.3	53.5	55.8	54.7	**56.1**	**48.4**	45.6	38.4
JPEG (75) [32]	60.5	50.9	40.1	28.7	55.8	56.8	56.2	52.9	47.0	51.4	42.3
FD [21]	59.5	50.9	41.2	28.5	56.5	56.7	55.6	51.3	46.9	50.4	44.1
TVM [13]	58.2	59.3	42.9	32.6	52.3	55.7	55.7	51.5	47.0	**52.1**	45.1
Proposed	**66.5**	**62.1**	**50.7**	**34.9**	**62.0**	**63.1**	**66.7**	55.1	48.0	52.0	**45.3**

tacked images which are generated using ResNet-v2-101 and tested on ResNet-v2-50. Because the attacks with extremly small perturbation magnitude like Deepfool and C&W usually have limited transferability, we adopt recently proposed black-box attack methods [36] and [10] into evaluation. Compared to employing the same attack parameters in Gray-box setting, adversarial examples have certain degree of transferability. As we can see in Table 2, though these adversarial examples have weak effect on networks, traditional JPEG, FD and TVM can hardly defend these transferred adversarial examples, especially for stronger perturbation intensity. Accuracy even declines when facing TI-FGSM [10] attack. As can be seen, TI attacks actually have worse transferability performance than FGSM. This is probably because the TI attack may be counterproductive if the discriminative regions of normally trained model and defense models are not that different in the case of low-strength attack. Further, the attack algorithm should generate adversarial examples with smaller dissimilarity while adversarial perturbations with $\epsilon = 8, 16$ as set in [10] have poor visual quality, which can be easily identified by human eyes. So we did not consider them in test. When facing larger intensity FGSM attacks, our proposed method can still achieve 8% higher accuracy than other defenses.

4.4. White-box Scenario

In this scenario, we evaluate our methods against white-box BPDA attack, which generates adversarial examples using defense methods iteratively. We implement the evaluation experiment on the released code at GitHub [2], using Inception-v4 [30] as test model, and 1000 iterations and 0.1 learning rate as attack parameters. The accuracy of various methods on adversarial examples (Acc_{ae}) and benign images (Acc_{raw}) are reported in Table 3. As can be observed, BPDA attack almost achieves 100% success rate. Due to our multiple-model setting, we can defend BPDA attack to a large extent. This is because BPDA works by finding a differentiable approximation ($f(\cdot)$) for a non-differentiable preprocessing transformation (*e.g.*, quantization) or network layer, $g(\cdot)$. Apparently, our proposed ensemble model set has various non-differentiable $g(\cdot)s$, and there would be extremely difficult for BPDA to find *one* $f(\cdot)$ to simultaneously approximate our *five* different $g(\cdot)s$, if not impossible. So our method is an easy and effective way to defend BPDA attack, and provides a new angle to redsign input-based defense to balance the accuracy of benign image and defense efficiency with defensive frequency domain quantization.

Table 3. Accuracy results (in %) in white-box scenario.

	None	JPEG (75)	FD	TVM	Proposed
Acc_{raw}	78	74	73	74	78
Acc_{ae}	0	1	2	0	60

5. Conclusion

In this paper, we combine the compression based input transformation method with re-training the compressed images. Firstly, we propose an optimized JPEG compression process to remove perturbations as much as possible and preserve the original features at the same time, by discarding the color space transformation and introducing an iterative algorithm to optimize the quantization table. Secondly we use images compressed with certain level as a strategy of data augmentation and generate a set of different network models. The final prediction will be the average results voted by each model in the set. Experimental results show that, our proposed method can improve defense efficiency while maintaining the classifying accuracy for benign images.

References

[1] A. Athalye, N. Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pages 274–283, 2018. 2, 7

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 2, 7, 8

[3] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 6

[4] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression. *CoRR*, abs/1705.02900, 2017. 1, 3, 6

[5] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using JPEG compression. *CoRR*, abs/1802.06816, 2018. 6

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 7

[7] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Max-margin adversarial (MMA) training: Direct input space margin maximization through adversarial training. *ICLR*, 2020. 5

[8] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. A study of the effect of JPG compression on adversarial images. *CoRR*, abs/1608.00853, 2016. 1, 2

[9] Alexei Efros and William Freeman. Image quilting for texture synthesis and transfer. In *Proc. SIGGRAPH*, pages 341–346, 2001. 2

[10] Y. Dong et al. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019. 8

[11] Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. Adversarial examples are a natural consequence of test error in noise. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2280–2289. PMLR, 09–15 Jun 2019. 6

[12] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. 2, 7, 8

[13] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. *ICLR*, 2018. 2, 7, 8

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 7

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV (4)*, pages 630–645, 2016. 7

[16] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR*, 2017. 2, 7, 8

[17] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *ICLR*, 2017. 1

[18] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019. 6

[19] Stanley Osher Leonid Rudin and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259 – 268, 1992. 2

[20] Zhijing Li, Christopher De Sa, and Adrian Sampson. Optimizing jpeg quantization for classification networks. *arXiv preprint arXiv:2003.02874*, 2020. 3

[21] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: DNN-oriented jpeg compression against adversarial examples. In *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868. IEEE, 2019. 3, 7, 8

[22] Zihao Liu, Tao Liu, Wujie Wen, Lei Jiang, Jie Xu, Yanzhi Wang, and Gang Quan. Deepn-jpeg: A deep neural network favorable jpeg-based image compression framework. In *Proceedings of the 55th Annual Design Automation Conference*, page 18. ACM, 2018. 2

[23] M.Abadi, P. Barham, and *et al*. Tensorflow:a system for large-scale machine learnig. *OSDI*, 16, 2016. 7

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 6

[25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 7, 8

[26] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *ICML*, pages 4970–4979, 2019. 6

[27] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, 2016. 1

[28] R. C. Reininger. Distributions of the two-dimensional dct coefficents for images. *IEEE Trans. Commun.*, 31(6):835–839, 1983. 3, 4

[29] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, pages 3353–3364, 2019. 1, 6

[30] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. 5, 7, 8

[31] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. 6

[32] G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 37, Feb 1992. 3, 4, 8

[33] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020. 1, 2, 5, 6, 7

[34] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020. 1

[35] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 501–509, 2019. 5

[36] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 8

[37] X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824, 2019. 1

[38] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, New York, NY, USA, 2017. Association for Computing Machinery. 6

[39] Tianhang Zheng, Di Wang, Baochun Li, and Jinhui Xu. A unified framework for randomized smoothing based certified defenses. 2019. 6