Online Alternate Generator Against Adversarial Attacks

Haofeng Li, *Member, IEEE*, Yirui Zeng, Guanbin Li, *Member, IEEE*, Liang Lin, *Senior Member, IEEE*, and Yizhou Yu, *Fellow, IEEE*

***Abstract*— The field of computer vision has witnessed phenomenal progress in recent years partially due to the development of deep convolutional neural networks. However, deep learning models are notoriously sensitive to adversarial examples which are synthesized by adding quasi-perceptible noises on real images. Some existing defense methods require to re-train attacked target networks and augment the train set via known adversarial attacks, which is inefficient and might be unpromising with unknown attack types. To overcome the above issues, we propose a *portable* defense method, online alternate generator, which does not need to access or modify the parameters of the target networks. The proposed method works by online synthesizing another image from scratch for an input image, instead of removing or destroying adversarial noises. To avoid pretrained parameters exploited by attackers, we alternately update the generator and the synthesized image at the inference stage. Experimental results demonstrate that the proposed defensive scheme and method outperforms a series of state-of-the-art defending models against gray-box adversarial attacks.**

***Index Terms*— Deep neural network, adversarial attack, image classification.**

I. Introduction

In recent years, deep convolutional neural networks have obtained state-of-the-art performances on many machine learning benchmarks, since they can harvest adaptive features on large-scale training sets, in comparison to traditional methods based on handcrafted features. However, deep learning models are found to be vulnerable with *adversarial attacks*, which aim at synthesizing *adversarial samples* that are perceptually similar to real images but can mislead attacked models to yield totally incorrect labels, as shown in Fig. 1(b). Adversarial examples can be generated by applying quasi-perceptible perturbations which does not change labels recognized by human subjects. Such perturbations can be computed via constrained optimization or backward propagation with an incorrect supervision. Thus, given a pretrained *target network* that might be accessed and attacked by hackers, how to protect it from adversarial attacks remains an important problem.

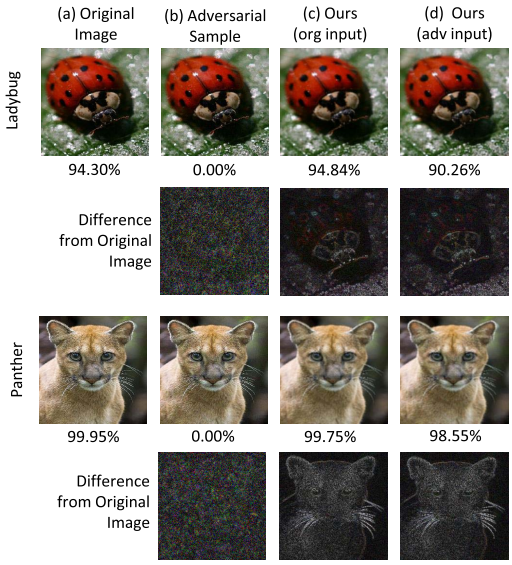


Fig. 1. The effectiveness of removing adversarial noise by our proposed method. (a) and (b) denote original images and their adversarial examples. (c) and (d) denote our generated results by taking original images and adversarial examples as reference respectively. The second row and the fourth row are normalized absolute difference between original images (a) and (b-d). Our results have different noise distribution from input adversarial examples. The predicted probability of correct labels is attached under each sample of (a-d). Our proposed defense method significantly increases the probability of predicting correct labels.

Many defense methods are developed to resist adversarial attacks on deep learning models. These defense methods can be roughly categorized into two groups. One group of defenses act as a preprocessing component which does not require accessing, modifying or re-training the attacked target network. These methods are portable and practical with different target networks or tasks since the knowledge of target networks may be confidential in real applications. These methods usually resort to image denoising and smoothing to

Manuscript received July 22, 2019; revised May 22, 2020; accepted September 11, 2020. Date of publication September 25, 2020; date of current version October 5, 2020. This work was supported in part by the National Key Research and Development Program under Grant 2020YFC2003902; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020B1515020048; in part by the National Natural Science Foundation of China under Grant 61976250, Grant 61702565, and Grant U1811463; in part by the Fundamental Research Funds for the Central Universities under Grant 18lgpy63; and in part by the CCF-Tencent Open Research Fund. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chia-Wen Lin. *(Corresponding author: Guanbin Li.)*

Haofeng Li is with the Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong at Shenzhen, Shenzhen 518172, China (e-mail: lhaof@foxmail.com).

Yirui Zeng, Guanbin Li, and Liang Lin are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: zengyr5@mail2.sysu.edu.cn; liguanbin@mail.sysu.edu.cn; linliang@ieee.org).

Yizhou Yu is with Deepwise AI Lab, Beijing 100080, China (e-mail: yizhouy@acm.org).

Digital Object Identifier 10.1109/TIP.2020.3025404

remove adversarial noises, or image transformations that could destroy adversarial noises to some extent. Another group of defense methods require to access or re-train the parameters of target network. We argue that these methods may be impractical and inefficient in real applications. For examples, adversarial training methods need to obtain the knowledge of adversarial attacks and might be unpromising to resist *unseen* attack types. Kurakin *et al.* [1] also suggest that adversarial training with single-step attacks does not confer robustness to iterative adversarial samples. Ensemble adversarial training [2] requires augmenting the train set by $N \times M$ times, designing and training N different target networks, which is inefficient. M is the number of different known adversarial attacks used in adversarial training. It is difficult to transfer DefenseGAN [3] on large images since training GAN with large images is unstable and might need to adjust the network architecture for different datasets.

Motivated by the above observations, in this article we aim at addressing such a problem: developing a *portable* defense method that protects a pretrained target network from *unseen* adversarial attacks with *images of large size*.

We define a portable defense as a method that does not need to access, modify or re-train the attacked target network. We claim that developing a portable defense is important because some parameters of the target network might be commercially confidential. Re-designing and re-training the target network could cause heavy load. A portable defense allows itself as a reusable self-contained component invoked via API. Developing a defense method against unseen attacks is critical since it is impractical to know the attack type used by attackers. On the other hand, developing a defense method working with large images, whose size is not smaller than 200×200, is practical and meaningful. Given the same L_∞ norm upper bound, the number of possible adversarial samples increases exponentially with the number of pixels. Thus defending against adversarial attacks with larger images is more difficult. Tramer *et al.* [2] also suggest that results obtained on simple datasets [4] with small images does not always generalize to harder tasks, for example, a classification benchmark with larger images.

To address the above-mentioned issues, we conceive a novel defending framework, online alternate generator. The proposed defense scheme works by online synthesizing an image that shares the same semantics with the input image but is almost free from adversarial noises. To avoid the model parameters stolen and exploited by attackers, we also propose to update the generator and the synthesized image at the inference stage, in an iterative and alternate manner. Besides, Gaussian noise is utilized as an additional perturbation in updating the synthesized image, to prevent the generator from fitting adversarial noises.

Our proposed method enjoys the following strengths. First, since the proposed method does not require any knowledge of target networks or adversarial attacks, it is a *portable* defense that can theoretically protect arbitrary target classifiers from arbitrary *unseen* adversarial attacks. Second, a Gaussian perturbation is added to yield an image, which not only introduces randomness but also decreases the probability of fitting adversarial noise on the given input. Third, as the proposed method adopts online training, its model parameters are not fixed during inference and cannot be accessed beforehand by potential attackers in real scene to synthesize an adversarial example.

In summary, this article has the following contributions:

- We develop a novel *portable* defense framework, online alternate generator, which can resist unseen adversarial attacks for unlimited pretrained classifiers, without the knowledge of target networks or adversarial attacks.
- We propose a stopping criteria which does not require accessing any adversarial samples such that the proposed method can deal with unseen attacks.
- We verify the transferability and generalization of the proposed method with different adversarial attacks, target models and benchmarks. Different selections of hyper-parameters are also well investigated.
- This article presents extensive experimental results to verify that the proposed framework surpasses a wide range of existing state-of-the-art defense algorithms.

II. Related Work

A. Adversarial Attack

Deep convolutional neural networks have demonstrated powerful fitting capacity in solving computer vision problems for last several years, but they are threatened by adversarial attacks [5]–[16]. Fast Gradient Sign Method (FGSM) [6] synthesizes adversarial examples by adding some weighted gradients which increases the prediction loss of the attacked target network, as shown in $I_{adv} = I + \epsilon \cdot sign(\partial L(F(I, \theta), Y_I)/\partial I)$ where I denotes a real image and I_{adv} is the generated adversarial example. $sign(\cdot)$ returns 1 for positive input, and returns -1 for negative one. $L(x, y)$ denotes a loss function that estimates the difference between a prediction x and the ground truth y. $F(\cdot, \theta)$ is the attacked target neural network with parameters θ. Y_I denotes the ground truth annotation of image I. ϵ is the l_∞ norm of the adversarial perturbation and the weight of the gradient with respect to I. Attack strength can be controlled by ϵ. The above algorithm is referred as *untargeted* attack. A *targeted* variant [1] of FGSM encourages the attacked network to predict high probability at some deliberately incorrect category Y_{adv}, as formulated in $I_{adv} = I - \epsilon \cdot sign\left(\partial L\left(F\left(I, \theta\right), Y_{adv}\right)/\partial I\right)$. Iterative Gradient Sign Method (IGSM) [1], [17] iteratively apply FGSM multiple times with a small step size to locate a stronger adversarial sample, as shown in the following.

$$\begin{aligned} I'_{t+1} &= I_t - \alpha \cdot sign(\partial L(F(I_t, \theta), Y_{adv})/\partial I) \\ I_{t+1} &= clip(I'_{t+1}, I - \epsilon, I + \epsilon) \end{aligned} \tag{1}$$

where I_t is the adversarial example synthesized after t iterations and $I_0 = I$. I'_{t+1} is a temporary variable. $clip(\cdot, l, u)$ with lower bound l and upper bound u, is an element-wise operator that ensures the L_∞ of the adversarial perturbation $|I_{t+1} - I|$ within the bound ϵ. Momentum based Iterative FGSM (MI-FGSM) [12] introduces a momentum term to stabilize the gradient descent and avoid the adversarial example stuck in poor local minima, which helps to synthesize more

transferable adversarial examples. Athalye *et al.* [18] propose three tricks, Backward Pass Differentiable Approximation, Expectation over Transformation and Reparameterization, which have broken into most existing obfuscated gradients based defenses under a white-box attack setting.

B. Gray-Box Attack Setting

Adversarial attacks have multiple settings, according to how much knowledge that adversaries can access. The settings consist of white-box, gray-box and black-box. Adversaries in white-box attacks can obtain all information about target models and defense methods. In black-box setting, adversaries do not know the architecture, the training data, the parameters of target classifiers and defense methods. Gray-box attacks have more than one definition. Previous works have defined different gray-box adversarial attacks and developed defense methods under their own settings. Taran *et al.* [19] define a gray-box setting, where the attacker knows target network architecture, training/testing data and the output label for each input, but has no knowledge of the network parameters and the defense mechanism. Zheng and Hong [20] utilize another gray-box setting, where attackers know the architecture of the target network and its defense strategy, but have no knowledge of their parameters. Different from the above settings where adversaries cannot obtain the network parameters, Guo *et al.* [21] introduce the following gray-box setting: the adversary has access to the model architecture and the model parameters, but is unaware of defense strategies. In this article we adopt the gray-box setting in [21]. It is a strong gray-box attack setting since both the architecture and parameters of target networks can be accessed by adversaries.

C. Defense Against Adversarial Attack

Many methods that aim at protecting some neural model from adversarial examples are proposed recently [3], [21]–[31]. SafetyNet [22] incorporates a deep convolutional neural network with a RBF-SVM which converts the final ReLU outputs to discrete codes to detect adversarial examples. Guo *et al.* [21] utilize bit-depth quantization, JPEG compression, total variance minimization and image quilting to destory or remove adversarial noise. Xie *et al.* [23] resort to random resizing and random padding during inference stage to leverage the cue that many adversarial examples are not scale invariant. Pixel Defend [24] iteratively updates each pixel of an input image using a pretrained PixelCNN [32] that is learned to predict a pixel value based on other pixels. Defend-GAN [3] learns to model the data distribution of clean images, and solves a vector in its learned latent space to approximate an input image. The latent vector is exploited to synthesize a substitute for the input via a generative neural network. HGD [25] makes use of high-level feature extracted by the attacked target network to train an image denoiser that can remove adversarial noise. Since HGD requires accessing the intermediate outputs of target classifiers to tune the denoiser, it is not a portable defense. MagNet [33] utilizes a detector to reject inputs far from the manifold boundary of training data, and an auto-encoder as reformer to find a substitute that is close to the input on the manifold. It is not straight-forward to fairly compare MagNet with other methods that do not reject adversarial examples. MagNet is based on the assumption that samples of some classification task are on a manifold of lower dimensions, and is only evaluated with small-size images. Pixel Deflection [26] iteratively and locally swaps two randomly sampled pixels according to their positions, before applying image denoiser to destroy adversarial noise. DIP [34] online trains a CNN that reconstructs the input image from a noise map. The output of the trained CNN is sent to be classified. Different from DIP, the proposed method updates an image and a CNN from scratch in an alternate way. The CNN does not directly output the synthesized image but approximates the energy of a data distribution for input images.

III. Method

This section describes the details of our proposed defense framework, online alternate generator, and its mathematical explanation. The proposed method can be regarded as a pre-processing component that protects a *target network* or *target classifier* from adversarial samples. *Target network* is defined as a pre-trained neural model that is exposed to be attacked. The parameters of a target network can be accessed by attackers. That is to say, adversarial samples are synthesized with the target network. The proposed method is portable and practical, and it does not require accessing, modifying or re-training the parameters of the target network. Different from some existing pre-processing defenses that try to remove or destroy adversarial noises, our proposed method synthesizes an image from scratch. The synthesized image is almost identical in appearance and semantics to the original image, but contains much less adversarial perturbations, and hence achieve more robust classification.

The overall pipeline of the proposed method is described in Algorithm 1. Given an input image that may be an adversarial sample, we define *reference image* (denoted as I_z in Algorithm 1) as the input image, and synthesize another image I_s to replace the original input before passing it into the target network for classification. For each input image, the parameters θ_1 of the generator F are initialized randomly and the synthesized image I_1 is filled with zeros at the beginning. T_N denotes the maximum number of outer iterations while T_I denotes the maximum number of inner iterations. Within each outer iteration, the synthesized image is updated for T_I times while the network parameters of our proposed method is updated once. Training the generator and synthesizing the image are conducted alternately. For each input image, the generator is updated for exactly T_N times while the synthesized image I_s is updated for $T_N * T_I$ times. Notice that θ_t in Algorithm 1 does not include any parameters of the target network, but only the parameters of the proposed defense method. In Line 7 of Algorithm 1, I_{T_I+1} is assigned to I_1, since we employ a circular array $I_{\{1,\ldots,T_I+1\}}$ to restore T_I+1 latest snapshots of the synthesized image. An overview of the proposed method is illustrated in Fig. 2, it is composed of an image updating procedure and a network updating

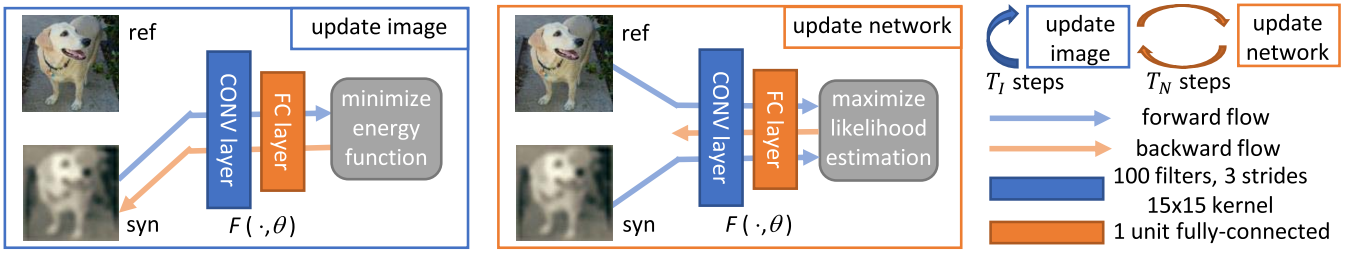


Fig. 2. A Defense Framework: Online Alternate Generator. ref denotes the reference image I_z. syn denotes the synthesized image I_s. $F(\cdot,\theta)$ is the neural network with parameters θ. The blue-boundary box represents image updating according to Eq. (3) while the orange-boundary box denotes network updating according to Eq. (5). The blue bending arrow means the inner iterations of image updating in Algorithm 1 while the orange bending arrows denotes the outer iterations of network updating.

Algorithm 1 Online Alternate Generation Algorithm

Input: I_z, reference image with potential adversarial noise
Output: I_s, synthesized image

```
Randomly initialize θ_1
I_1 = 0
for t = 1 to T_N do
    for s = 1 to T_I do
        Update I_{s+1} with I_s according to Eq. (3)
    end for
    I_1 = I_{T_I+1}
    Update θ_{t+1} with θ_t according to Eq. (5)
end for
I_s = I_{T_I+1}
return I_s
```

process. Please refer to Algorithm 1 for the detailed iteration. In the following sections, we will focus on the computation of both image updating and network updating, and the theoretical basis behind this alternative updating mechanism.

A. Image Updating

Let us discuss how to update a synthesized image initialized by zeros, with a reference image I_z and a neural network F. F denotes the proposed generator instead of the target classifier. Suppose I_z and I_s are sampled from the same data distribution denoted as $p(I;\theta) = (1/Z)e^{-U(I;\theta)}$ where I denotes an image. θ represents the parameters of the model and Z is a normalization term. e denotes exponential and U is an energy function. Then we utilize the neural network F in the proposed framework to approximate the energy function, i.e., $F(I,\theta) = -U(I;\theta)$. To maximize the probability density of the synthesized image I_s, we update I_s to minimize the energy function by $I_{s+1} = I_s - \alpha\partial U(I_s;\theta)/\partial I = I_s - \alpha(-\partial F(I_s,\theta)/\partial I)$ where I_s is the current synthesized image and I_{s+1} is the updated image. α denotes the learning rate. $\partial F(I,\theta)/\partial I$ is the gradients of neural network $F(\cdot,\theta)$ with respect to image I, and can be computed by backward propagation. In a sense, generating an image is to reconstruct the reference image. However, synthesizing an image with adversarial noise is undesirable. Thus we further introduce a noise model during image updating,

$$I_{s+1} = I_s - \alpha(-\partial F(I_s,\theta)/\partial I) + \epsilon_g D \quad (2)$$

where D denotes some noise distribution, such as a gaussian noise $N(0,1)$. ϵ_g is the strength of the noise. Adding noise during image synthesis can increase the difficulty in recovering subtle details, and thus decrease the chance of fitting adversarial noise. Langevin Dynamics, which is originally to simulate how particles move under a random force, has a similar form with Eq (2), a modified gradient decent with a Gaussian perturbation. To comprehend the relationship between α and ϵ_g, we resort to Langevin Dynamics and come up with Eq (3) following a previous work [35].

$$I_{s+1} = I_s - (\epsilon_g^2/2)(I_s - \partial F(I_s,\theta)/\partial I) + \epsilon_g N(0,1) \quad (3)$$

where ϵ_g controls the magnitude of the Gaussian noise. $\epsilon_g^2/2$ corresponds to the learning rate α. $1-\epsilon_g^2/2$ is the inertia factor of I_s. Since random fluctuation is used to generate an image, the distribution of images is changed into $p(I;\theta) = (1/Z)e^{-U(I;\theta)}(1/(2\pi)^{|S|/2})e^{-\frac{1}{2}||I||^2}$. The multiplicative term on the right is a Gaussian distribution with $\sigma^2 = 1$. $|S|$ denotes the number of elements in image I.

B. Network Updating

The following details how to update the neural network F such that the synthesized image I_s gradually approximates the reference image I_z. Notice that at the very beginning F is initialized by randomization. Thus we update F to maximize the likelihood with respect to I_z. Let $L(\theta) = log(p(I_z;\theta))$. θ is trained along the direction maximizing the log likelihood $L(\theta)$ with gradient descent: $\theta_{t+1} = \theta_t + \beta\partial L(\theta_t)/\partial\theta$, where θ_t denotes the current parameters at the time step t. θ_{t+1} is the updated parameters. $\partial L(\theta_t)/\partial\theta$ denotes the gradients of the log likelihood function w.r.t θ. As suggested in [36], the gradient is computed in:

$$\partial L(\theta)/\partial\theta = \partial F(I_z;\theta_t)/\partial\theta - E_{p(I;\theta)}[\partial F(I;\theta_t)/\partial\theta], \quad (4)$$

where $E_{p(I;\theta)}[\cdot]$ is the expectation with I following the distribution $p(I;\theta)$. The expectation is not explicitly calculated but approximated by sampling. Langevin Dynamics, which is adopted to update the image, is also a tool to sample image I from distribution $p(I;\theta)$. Thus we choose $\partial F(I_s;\theta_t)/\partial\theta$ to approximate the expectation $E_{p(I;\theta)}[\cdot]$ for simplicity. Then learning neural network F can be formulated as in Equation 5.

$$\theta_{t+1} = \theta_t + \beta(\partial F(I_z;\theta)/\partial\theta - \partial F(I_s;\theta)/\partial\theta) \quad (5)$$

For a testing image, T_N samples $\partial F(I_s;\theta)/\partial\theta$ are selected to approximate $E_{p(I;\theta)}[\partial F(I;\theta)/\partial\theta]$. In practice, T_N ranges from 200 to 300. Thus, hundreds of samples $\partial F(I_s;\theta)/\partial\theta$ are used to approximate the expectation $E_{p(I;\theta)}[\partial F(I;\theta)/\partial\theta]$ and such approximation is effective.

C. Analysis

This subsection presents explanation on why the synthesized image I_s approximate the reference image I_z after alternately updating I_s and the network F. In the following, we consider F as a very simple prototype. The prototype model contains a convolution layer, a ReLU layer and a summation operator. The prototype model contains a convolution layer, a ReLU layer and a summation operator. I, the input of prototype model, could be reshaped as a vector of shape $1 \times chw$. The convolution layer has K kernels and the kernel size is $r_h \times r_w$. The weight of the convolution is denoted as W of size $cr_hr_w \times K$, while the bias B is of size $1 \times K$. For simplicity, we let $h = r_h$ and $w = r_w$ so that the convolution operator is only applied on a position. The output of the convolution is $IW + B$, of size $1 \times K$. The ReLU function λ is applied on each element of $IW + B$. The summation operator sums up all elements of $\lambda(IW + B)$. Thus $F(I)$ outputs a single scalar and is formulated as:

$$F(I) = \sum\nolimits_{k=1}^{K} \lambda(IW_k + B_k) \tag{6}$$

According to the definition of ReLU, $\lambda(x) = max(0, x)$. ReLU function can be represented as a multiplication between the input and a Heaviside step function u. For $x > 0$, $u(x)$ equals 1. Otherwise, $u(x)$ is 0. Thus $\lambda(x) = u(x)x$. Then we formulate Eq (6) as:

$$\begin{aligned} F(I) &= \sum\nolimits_{k=1}^{K} (u(IW_k + B_k)(IW_k + B_k)) \\ &= \sum\nolimits_{k=1}^{K} (u(I,\theta)(IW_k + B_k)), \end{aligned} \tag{7}$$

where $u(IW_k + B_k)$ is a single scalar depending on I and θ. We denote $u(IW_k + B_k)$ as $u(I,\theta)$ for simplicity.

Assume that the synthesized image I_s and the reference image I_z belong to the same data distribution $p(I;\theta)$. Since we introduce Gaussian noise to synthesize I_s, we assume that

$$p(I;\theta) = \frac{1}{Z} e^{-U(I;\theta)} \frac{1}{(2\pi)^{|S|/2}} e^{-||I||^2/2} \tag{8}$$

Let I^* denote the image that maximizes the probability $p(I;\theta)$. The proposed method is to synthesize an image I_s that approximates to I^*. Let $F(I;\theta)$ approximate $-U(I;\theta)$. To maximize the probability, we need to maximize $-U(I;\theta) - ||I||^2/2$ as:

$$\begin{aligned} &F(I) - ||I||^2/2 \\ &\quad = -||I||^2/2 + I \sum\nolimits_{k=1}^{K} (u(I,\theta)W_k) + \sum\nolimits_{k=1}^{K} u(I,\theta)B_k \\ &\quad = -||I - \sum\nolimits_{k=1}^{K} (u(I,\theta)W_k)||^2/2 + C(u(I,\theta),\theta) \end{aligned} \tag{9}$$

where $C(u(I,\theta),\theta)$ depends on I and θ. When we update image I_s as I_{s+1}, we can set $u(I,\theta)$ as $u(I_s,\theta)$ and fix the network parameters θ. Then we come up with $I^* = \sum_{k=1}^{K}(u(I_s,\theta)W_k)$ to maximize Eq (9). Recall how we update I_{s+1} as shown in Eq (3):

$$\begin{aligned} I_{s+1} &= I_s - (\epsilon_g^2/2)(I_s - \partial F(I_s,\theta)/\partial I) + \epsilon_g N(0,1) \\ &= (1 - \epsilon_g^2/2)I_s + \epsilon_g^2/2 \cdot \partial F(I_s,\theta)/\partial I + \epsilon_g N(0,1) \\ &= (1 - \epsilon_g^2/2)I_s + \epsilon_g^2/2 \sum\nolimits_{k=1}^{K} (u(I_s,\theta)W_k) + \epsilon_g N(0,1) \\ &= (1 - \epsilon_g^2/2)I_s + \epsilon_g^2/2 \cdot I^* + \epsilon_g N(0,1) \end{aligned} \tag{10}$$

As Eq (υ), updating I_{s+1} is to do linear interpolation between I_s and I^* with a Gaussian perturbation. Thus the pixel values of the synthesized image I_s will approximate to those of I^*, which is the peak of highest probability. Since I_z is a sample of high probability in the same data distribution with I^*, I_s will also be visually similar to I_z. When updating the network, the proposed method tunes θ to guarantee that the probability of sampling I_s is as high as sampling I_z.

In our real implementation (shown in Fig. 2) of the proposed method, the neural network F, which contains a convolution, a fully-connected layer and a non-linear activation, can be approximated by the above-mentioned model. Therefore the above analysis also works for our actual implementation. Here we discuss the time complexity of the proposed online alternate generation algorithm (shown in Algorithm 1). The proposed algorithm consists of two nested loops. The outer loop contains T_N iterations while the inner loop has T_I iterations. Assume that computing Eq. (3) takes the same constant time as computing Eq. (5). The overall time complexity is $\mathcal{O}(T_N(T_I + 1)) = \mathcal{O}(T_N \cdot T_I)$. In practice, it takes about 30 seconds to process an input image of size 224×224.

D. Stopping Criteria

The proposed online alternate generation terminates after T_N outer iterations as shown in Algorithm 1. On one hand, larger network step T_N leads to unnecessary computations. On the other hand, with smaller T_N, the proposed generator could fail to reproduce the semantics of an input image, which will seriously drop the classification accuracy of target networks. More importantly, to develop a portable defense against unseen attacks, we need to determine T_N without tuning it on known adversarial samples. Here, we propose to utilize images perturbed by Gaussian noises to choose the network steps T_N. It is based on an assumption that the online alternate generating process of adversarial samples and those with Gaussian noises are similar. The experimental details and results are presented in Section IV-E.

IV. Experiments

A. Experimental Setting

1) Dataset: We evaluate the performance of our method on ILSVRC 2012 dataset [39] and Oxford Flower-17 dataset [40]. Most images in ILSVRC 2012 dataset are large images whose size is not smaller than 200×200. Most images in Oxford Flower-17 dataset are larger than 500×500. We claim that it is practical and meaningful to develop a defense method that works with large images. Because experimental results obtained on simple dataset such as MNIST [4] do not

TABLE I

TOP-1 ACCURACY OF DIFFERENT DEFENSE METHODS AGAINST FGSM, IGSM, MI-FGSM AND C&W ON ILSVRC 2012 DATASET AND OXFORD FLOWER-17 DATASET. ∗ INDICATES THAT THE METHOD NEEDS TO BE TRAINED BEFORE INFERENCE

Defense Methods	ILSVRC 2012				Oxford Flower-17			
	FGSM	IGSM	MI-FGSM	C&W	FGSM	IGSM	MI-FGSM	C&W
None	8.35%	0.05%	0.30%	0.00%	27.65%	7.35%	0.59%	0.00%
Mean Filter [37]	39.65%	67.65%	38.50%	75.55%	60.59%	73.24%	47.06%	66.47%
JPEG [38]	19.70%	54.55%	0.85%	67.50%	45.88%	47.94%	20.88%	28.82%
TVM [21]	41.00%	66.70%	53.45%	69.00%	69.12%	83.82%	70.88%	80.29%
Pixel Deflection [26]	41.70%	55.80%	51.10%	57.00%	71.47%	82.06%	70.59%	78.82%
Randomization [23]	40.05%	67.15%	44.80%	68.30%	72.65%	83.24%	62.94%	75.88%
∗Pixel Defend [24]	20.15%	55.25%	20.05%	66.75%	48.82%	53.53%	27.06%	37.35%
Ours	49.10%	75.25%	55.90%	77.95%	76.76%	87.35%	73.82%	84.41%

always generalize to harder tasks, as suggested by [2]. As suggested in [23], it is less meaningful to attack misclassified images, we randomly choose 2000 correctly classified images (2 images/class) from the validation set to perform experiments.

2) Target Network: We choose ResNet18 [41] as the attacked target network on Oxford FLower-17 dataset, and utilize ResNet50 on ILSVRC 2012 dataset. To demonstrate the transferability of out proposed defense method, we further conduct experiments on Oxford Flower-17 dataset with different target networks, including VGG11 [42], MobileNet_v2 [43] and DenseNet121 [44].

3) Attack Methods: We exploit FGSM [6] and IGSM [1] to construct targeted adversarial examples with randomly selected target categories. We utilize MI-FGSM [12] and C&W attack [45] to synthesize untargeted adversarial samples. In this article, we adopt gray-box attacks [21] in which attackers can access the target network and its parameters, but have no knowledge of defense methods. L_∞ norm is adopted to bound adversarial perturbations. ϵ denotes the upper bound of L_∞ norm. We choose ϵ for each attack type such that the adversarial attack is strong enough and its resulting perturbation is imperceptible. For example, we attack a ResNet18 model by targeted IGSM, as shown in Fig. 3. As ϵ grows from 0 to 5, the top-1 accuracy of the attacked model drops rapidly. For ϵ larger than 5, the top-1 accuracy becomes stable. In such case, larger perturbations do not lead to stronger attack but only synthesize noisy and undesirable images. For FGSM, IGSM and MI-FGSM, ϵ is selected respectively as {6, 6, 2} on ILSVRC 2012 dataset and {6, 6, 4} on Oxford Flower-17 dataset. The number of iterations is set as $\min(\epsilon + 4, \text{ceil}(1.25\epsilon))$, according to [1]. ceil$(\cdot)$ denotes rounding up to an integer. The step size is set as 1.

4) Defense Methods: In this article, we compare our method with six state-of-the-art portable defenses, including Mean Filter [37], JPEG compression and decompression [38], TVM [21], Pixel Deflection [26], Randomization [23] and Pixel Defend [24]. All of these methods play a role as preprocessings. In addition, the first five methods can be used without training. Although Pixel Defend needs to train a PixelCNN [32] network on the training set, it does not need to access the parameters of adversarial attacks. Pixel Defend has been successfully attacked by Athalye *et al.* [18] in white-box setting, but it is still meaningful to compare it with our proposed method in the gray-box setting.

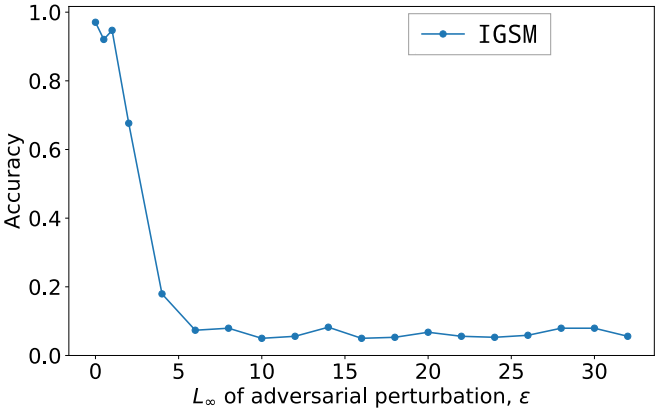


Fig. 3. Top-1 accuracy of ResNet18 attacked by targeted IGSM with different ϵ (l_∞ norm of adversarial perturbations) on Oxford Flower-17 dataset.

B. Comparison With the State-of-the-Art

We choose existing portable defenses that can work with unseen attack types and large images for comparison. These defense methods do not require accessing target networks or attack types. This experiment is conducted with gray-box attack setting. The parameters of all defenses are invisible to attackers. As can be observed in TABLE I, our proposed defense method achieves the best top-1 accuracy on both ILSVRC 2012 dataset and Oxford Flower-17 dataset, against four kinds of attacks, including FGSM, IGSM, MI-FGSM and C&W. On ILSVRC 2012 dataset, The proposed method outperforms the second best Pixel Deflection by 7.4% against FGSM, and surpasses the second best Mean Filter by 7.6% against IGSM. Our proposed method outperforms the second best method TVM by 2.45% top-1 accuracy on MI-FGSM and Mean Filter by 2.4% top-1 accuracy on C&W. On Oxford Flower-17 dataset, the top-1 accuracy of our method is 4.1% higher than the second best against FGSM and 3.5% higher than TVM against IGSM. The proposed method outperforms

TABLE II

INVESTIGATION OF THE PROPOSED METHOD WITH ENSEMBLE ADVERSARIAL TRAINING. ADVERSARIAL EXAMPLES ARE SYNTHESIZED USING MI-FGSM ON ILSVRC 2012 DATASET

Attacks With	IncV3ens	IncV3 + ours	IncV3ens + ours
IncV4	71.80%	73.30%	78.45%
IncV3	34.70%	39.60%	71.70%

the second best method, TVM, by 2.94% top-1 accuracy on MI-FGSM and 4.12% top-1 accuracy on C&W attack.

TABLE I shows that top-1 accuracy on IGSM is better than FGSM, while the results in [17] suggest that IGSM is stronger than FGSM on white-box attacks with the same ϵ. We claim that our experimental results is reasonable because our experiments is tested on gray-box attacks. In our cases, defense methods shown in TABLE I have modified the adversarial perturbations in adversarial samples. Since adversarial perturbations computed in iterative methods 'fit' the target network better than single-step methods, minor changes on iterative adversarial samples could reduce more attack effects than single-step adversarial samples. It can be regarded as an evidence supporting that stronger adversaries decrease transferability [17]. Experimental results in Pixel Deflection [26] also show that classifier accuracy of IGSM is higher than FGSM on gray-box attacks. Notice that the numeric results of Pixel Deflection in our experiment may differ from those in [26]. It is due to that we adopt L_∞ norm following the original FGSM/IGSM while Prakash *et al.* [26] use L_2 norm. Different norm may result in different distribution of adversarial noises and attack effects.

We discuss two reasons why the proposed method are better than the previous works. First, the proposed method synthesizes a new image with less adversarial noises to replace the original input. In Sec III-C we have shown that the synthesized image I_s approximates to the original input I_z. The introduced Gaussian perturbation in Eq (3) avoids recovering the adversarial noises of I_z. Second, the transferability among CNN models may be a reason. When updating image I_s, F has also been updated for hundreds of times. That is to say, I_s is synthesized based on hundreds of different CNN models (a model F with different parameters). The pixel values of I_s could be suitable for another CNN model (such as the target classifier) to extract effective features.

C. Investigation With Adversarial Training

This section presents how our proposed method work with Ensemble Adversarial Training method [2]. Noted that adversarial training based methods are not *portable* defense as they require to re-train the attacked target networks. As shown in TABLE II, IncV3ens, IncV3+ours and IncV3ens+ours respectively denote an ensemble adversarial training IncV3 (Inception-V3) [46] model, a plain IncV3 model defended by our proposed method and the Ensemble Adversarial Training model defended by our proposed method. 'Attacks With' denotes the pre-trained models used to synthesize adversarial examples. The pre-trained IncV3 model has the same architecture but different parameters from IncV3ens. Thus attacks with IncV4 (Inception-V4) [47] and IncV3 are in black-box setting and gray-box setting respectively. The results indicate that the proposed method outperforms Ensemble Adversarial Training by 1.5% and 4.9% top-1 accuracy in black-box and gray-box setting respectively. Besides, our proposed method significantly enhances the top-1 accuracy of the IncV3ens model by 6.65% and 37% top-1 accuracy against attacks with IncV4 and IncV3 respectively. Notice that Ensemble Adversarial Training and the proposed method obtain a relatively lower accuracy (less than 40%) on gray-box setting. Such situation could be due to that the IncV3 model is relatively more sensitive to the MI-FGSM attack than other models such as ResNet. Besides, we find that combining the proposed method with Ensemble Adversarial Training can achieve acceptable top1-accuracy of 71.70%.

TABLE III

TOP-1 ACCURACY OF DIFFERENT DEFENSE METHODS AGAINST IGSM ON OXFORD FLOWER-17 DATASET WITH DIFFERENT TARGET NETWORKS. $*$ INDICATES THAT THE METHOD REQUIRES OFFLINE TRAINING

Defense Methods	**ResNet18**	**VGG11**	**MobileNet_v2**	**DenseNet121**
None	7.35%	9.71%	5.59%	5.29%
Mean Filter	73.24%	69.12%	82.65%	60.00%
JPEG	47.94%	50.00%	58.53%	27.56%
TVM	83.82%	67.94%	88.53%	73.24%
Pixel Deflection	82.06%	77.06%	82.65%	82.56%
Randomization	83.24%	72.65%	86.75%	72.53%
$*$Pixeldefend	53.53%	50.88%	61.47%	35.29%
Ours	87.35%	80.29%	91.18%	87.65%

D. Transferability Analysis

This section investigates whether our defense method is transferable to protect different kinds of target networks. Experiments are conducted on Oxford Flower-17 dataset. Adversarial examples are generated using IGSM in this section. The target networks includes VGG11, MobileNet_v2 and DenseNet121. These target networks are initialized with weights pretrained on ImageNet, and then fine-tuned on the training set of Oxford FLower-17 dataset. The top-1 accuracy of VGG11, MobileNet_v2 and DenseNet121 are 91.47%, 97.35% and 96.47% respectively on clean images from the test set. We determine ϵ according to the criteria discussed in the above section, and select $\epsilon = 6$ for ResNet18, $\epsilon = 8$ for VGG11, $\epsilon = 6$ for MobileNet_v2 and $\epsilon = 8$ for DenseNet121.

As shown in TABLE III, the proposed method demonstrates higher top-1 accuracy than other state-of-the-art defense algorithms. Our method exceeds TVM by 3.5% top-1 accuracy on ResNet18, and outperforms Pixel Deflection by 3.2% top1-accuracy on VGG11. On MobileNet_v2, the proposed method surpasses the second best TVM by 2.6% top-1 accuracy. The top1-accuracy of our method is 5.1% higher than the second best Pixel Deflection on DenseNet121. Notice that the proposed method does not need offline training beforehand, and therefore does not obtain any knowledge or response of the attacked target networks. Tuning hyper-parameters is

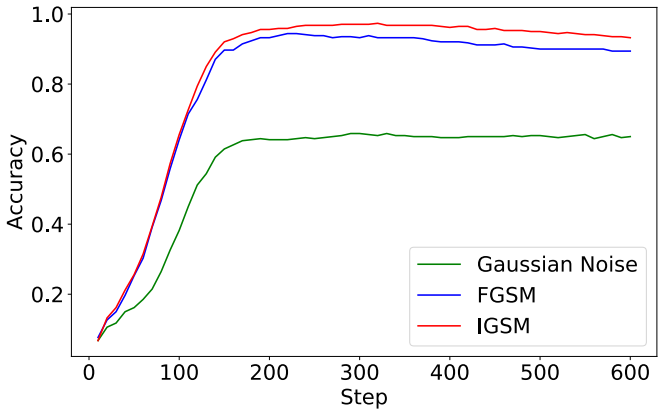


Fig. 4. Comparison among reference images with different noises. 'Accuracy' denotes top-1 accuracy. 'Step' corresponds to Network Steps. **Gaussian Noise** means that images are degraded by Gaussian Noise. **FGSM** and **IGSM** denote adversarial noises generated by FGSM and IGSM.

also independent of attacked target models. Thus the proposed method is not biased towards any specific models, but enjoys superior transferability with various types of target networks.

E. Ablation Study

Our proposed defense method contains several essential hyper-parameters, including Network Steps T_N, Image Steps T_I and kernel size. We investigate how different settings of these hyper-parameters affect the performance of our method on Oxford Flower-17 dataset. When investigating a hyper-parameter, other hyper-parameters are set as default values suggested in Section IV-A.

1) Network Steps T_N: To determine T_N, the number of iteration (in the outer loop), a straightforward way is to generate adversarial samples with training set, run the proposed method on these adversarial samples, and find out a suitable value for T_N. However, as a defender we usually do not know the exact attack type in testing stage. Thus it is meaningful to determine T_N without the knowledge of adversarial attacks. Since our proposed method does not estimate the attack type of an input image, we assume that using samples with non-adversarial noises also can determine the iteration number. Then we conduct an experiment to verify the assumption. We generate three sets of noisy samples for training set. One set is generated by adding Gaussian noises ($\mu = 0$, $\sigma = \pm 0.25$). The other two sets are generated by adding adversarial noises of FGSM and IGSM. We run the proposed method for different iterations on these three sets of data. We use the target image classifier to classify the images synthesized by the proposed method at different iterations. The results is shown in Fig 4 that has taken average of the whole training set. It does reflect most of cases. The resulting curve of IGSM, FGSM and Gaussian noises are marked with red, blue and green color in Fig 4. As can be seen, these three curves have almost the same trend, which converge and become flat after around 200 steps. Thus we can barely rely on the curve corresponding to Gaussian noise to determine a suitable Network Steps. On ILSVRC 2012 dataset, Network Steps is selected as 600 in the same way.

TABLE IV
COMPARISON AMONG DIFFERENT IMAGE STEPS.

Image Steps T_I	10	20	30	40
Top-1 Accuracy	87.05%	87.35%	87.35%	**87.65%**

TABLE V
COMPARISON AMONG DIFFERENT KERNEL SIZES.

Kernel Sizes	7×7	11×11	15×15	21×21
Top-1 Accuracy	85.29%	86.18%	**87.35%**	86.18%

2) Image Steps T_I: We evaluate four different Image Steps with kernel size set to 15×15 and Network Steps set to 300. As shown in TABLE IV, the top-1 accuracy increases as more Image Steps is adopted. However, as larger Image Steps leads to higher time cost, we select $T = 20$ to make a trade-off between performance and efficiency.

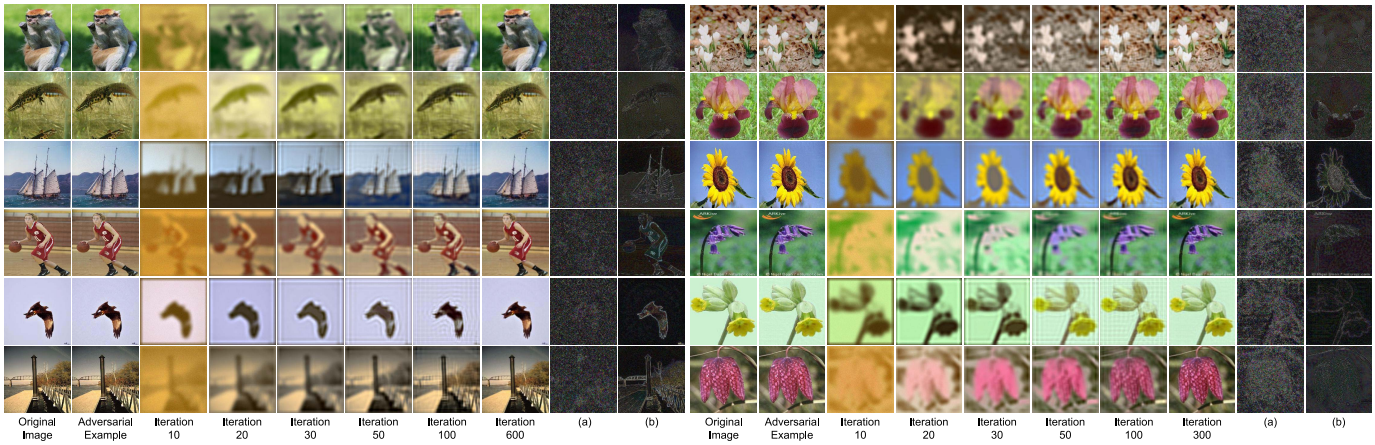
3) Kernel Size: We evaluate four different kernel sizes of the first convolution layer, with Image Steps set to 20 and Network Steps set to 300. TABLE V shows that our method performs the best with kernel size 15×15. Small receptive field is susceptible to adversarial perturbations. Convolutions with smaller receptive field benefit the reconstruction of image details, even including the adversarial perturbation on the given inference image. On the contrary, images synthesized by convolutions with larger receptive field may lower the quality of small patterns and details, which also degrades the classification accuracy.

F. Investigation With Natural/Clean Images

We present the results on both clean and adversarial samples, as shown in TABLE VI. The results are obtained on Oxford Flower-17 with C&W as attack and ResNet18 as target classifier. 'None' denotes the target classifier without any defenses. In TABLE VI, JPEG and Pixel Defend are the best on clean images, but their top-1 accuracy are less than 40% and the worst on adversarial images. TVM is worse than our method on both clean and adversarial samples. Pixel Deflection is close to our method on clean images but our method exceeds it by 5.59% top-1 accuracy against the adversarial attack. Randomization surpasses our method by 2.9% on clean images but our method outperforms it by 8.6% on adversarial examples. Comparing with DIP, our proposed method obtains 1.5% higher top1-accuracy against adversarial samples, and 2.4% lower accuracy on clean images. Overall, the proposed method achieves the state-of-the-art trade-off between natural images and adversarial examples.

G. Visualization of Online Alternate Generation

This section visualizes the synthesized image of our defense method during the alternate generation. As shown in Fig. 5, the columns captioned with 'Iteration k' are the intermediate synthesized images of our proposed defense method at iteration k. The maximum number of iterations corresponds to *Network Steps* that controls how many times the neural



i). ILSVRC 2012 dataset ii). Oxford Flower-17 dataset

Fig. 5. The visualization of our defense method on ILSVRC 2012 and Oxford Flower-17 dataset under targeted Iterative FGSM attack. We exhibit images generated by our defense method at different iterations. (a) visualizes the difference between adversarial examples and original images. (b) visualizes the difference between our results and original images. As can be observed, our results have different noise distribution from input adversarial examples.

TABLE VI
INVESTIGATION WITH NATURAL/CLEAN IMAGES

Defense Methods	Clean Images	Adversarial
None	97.06%	0.00%
Mean Filter [37]	94.71%	66.47%
JPEG [38]	97.06%	28.82%
TVM [21]	90.88%	80.29%
Pixel Deflection [26]	92.35%	78.82%
Randomization [23]	95.00%	75.88%
Pixel Defend [24]	96.76%	37.35%
DIP [34]	94.41%	82.94%
Ours	92.06%	84.41%

model in our method is updated. As the iteration number increases, the synthesized image become clearer and sharper. The synthesized image at the last iteration is visually similar to the original image. The column captioned with (a) is the *residual map* of an adversarial example (synthesized by Iterative FGSM). The residual map of a synthesized image is defined as a normalized pixel-wise absolute difference between the synthesized image and its corresponding original image. The column captioned with (b) is the residual map of the synthesized image in our method. Comparing column (a) with column (b), the residual map of an adversarial examples differs from that of the synthesized image in our method, which suggests that the noise distribution between an adversarial example and its corresponding synthesized image is quite distinct. The synthesized images could be less affected by the adversarial noises.

H. Whether Alternate Update Is Necessary

To understand whether the alternate update scheme in our proposed method is necessary, let us consider a simple auto-encoder. The auto-encoder employs online training strategy as the proposed method to achieve portable defense, but does not adopt two-step update. Given an image, we first randomly initialize the parameters of the auto-encoder, and then online tune its parameters by taking the image as input and supervision. After T_N-step updates, the auto-encoder takes the image as input and outputs another image. The output serves as a substitute to be sent into a target classifier. For fair comparison, we instantiate the auto-encoder in a symmetric form, and its encoder part as the same as the proposed generator. The auto-encoder consists of a convolution layer, two fully-connected layers, a non-linear activation and a deconvolution layer. The fully-connected layers are in between the convolution and the deconvolution while the non-linear layer is located in between these two fully-connected layers. The first fully-connected layer encodes a tensor into a scalar while the second one decodes a scalar into a tensor. The auto-encoder achieves 82.94% on Oxford Flower-17 benchmark with C&W as attack and ResNet18 as target classifier. Our proposed method using alternate update obtains 84.41%, slightly better than the auto-encoder using one-step update. It may be due to that the alternate update with Langevin Dynamics could better sample a representative point in some latent space, and hence recover more accurate semantics for an input image.

V. CONCLUSION

In this article we develop a novel portable defense framework that reconstructs an image with less adversarial noise and almost the same semantics as an input image. The reconstructed image acts as a substitute to defend against adversarial attacks. The hyper-parameters of our proposed defense do not need to be tuned on any adversarial examples, which avoid a bias towards some known attacks. The proposed defense does not access, modify any parameters or intermediate outputs of target models, which allows the defense portably transferable to a wide range of target classifiers. Experimental results show that our method obtains the state-of-the-art performance.

REFERENCES

[1] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–17.
[2] F. Tramer, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. Mcdaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–22.

[3] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–17.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[5] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–10.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–11.

[7] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[8] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1378–1387.

[9] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer, "Universal adversarial perturbations against semantic image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2774–2783.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 86–94.

[11] M. M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured visual and speech recognition models with adversarial examples," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6977–6987.

[12] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.

[13] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, "Generative adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4422–4431.

[14] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.

[15] A. Arnab, O. Miksik, and P. H. S. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 888–897.

[16] C. Xiao, J. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–29.

[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–28.

[18] A. Athalye, N. Carlini, and D. A. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283.

[19] O. Taran, S. Rezaeifar, and S. Voloshynovskiy, "Bridging machine learning and cryptography in defence against adversarial attacks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 267–279.

[20] Z. Zheng and P. Hong, "Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7924–7933.

[21] C. Guo, M. Rana, M. Cisse, and L. V. Der Maaten, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[22] J. Lu, T. Issaranon, and D. Forsyth, "SafetyNet: Detecting and rejecting adversarial examples robustly," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 446–454.

[23] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. L. Yuille, "Mitigating adversarial effects through randomization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[24] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–20.

[25] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1778–1787.

[26] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8571–8580.

[27] N. Akhtar, J. Liu, and A. Mian, "Defense against universal adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3389–3398.

[28] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–12.

[29] H. Li, G. Li, and Y. Yu, "ROSA: Robust salient object detection against adversarial attacks," *IEEE Trans. Cybern.*, early access, May 17, 2019, doi: 10.1109/TCYB.2019.2914099.

[30] X. He, S. Yang, G. Li, H. Li, H. Chang, and Y. Yu, "Non-local context encoder: Robust biomedical image segmentation against adversarial attacks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8417–8424.

[31] J. Yang *et al.*, "An adversarial perturbation oriented domain adaptation approach for semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12613–12620.

[32] A. V. Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1747–1756.

[33] D. Meng and H. Chen, "Magnet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2017, pp. 135–147. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134057

[34] A. Kattamis, T. Adel, and A. Weller, "Exploring properties of the deep image prior," in *Proc. Adv. Neural Inf. Process. Syst. Workshop*, 2019, pp. 1–7.

[35] Y. Lu, S. Zhu, and Y. N. Wu, "Learning FRAME models using CNN filters," in *Proc. Nat. Conf. Artif. Intell.*, 2016, pp. 1902–1910.

[36] J. Xie, W. Hu, S.-C. Zhu, and Y. N. Wu, "Learning sparse FRAME models for natural image patterns," *Int. J. Comput. Vis.*, vol. 114, nos. 2–3, pp. 91–112, Sep. 2015.

[37] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5775–5783.

[38] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of JPG compression on adversarial images," 2016, *arXiv:1608.00853*. [Online]. Available: http://arxiv.org/abs/1608.00853

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[40] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1447–1454.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[44] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[45] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[47] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–12.

Haofeng Li (Member, IEEE) received the B.Sc. degree from the School of Data and Computer Science, Sun Yat-sen University, in 2015, and the Ph.D. degree from the Department of Computer Science, The University of Hong Kong, in 2020. He is currently a Research Scientist with the Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong at Shenzhen. His current research interests include computer vision, image processing, and deep learning. He was a recipient of the Hong Kong Ph.D. Fellowship. He has been serving as a Reviewer for IEEE TRANSACTIONS ON IMAGE PROCESSING, *Pattern Recognition*, *Neurocomputing*, and IEEE ACCESS.

Yirui Zeng received the B.S. degree from the School of Electronics and Information Engineering, Chongqing University, in 2017. She is currently pursuing the master's degree with the School of Electronics and Information Engineering, Sun Yat-sen University. Her current research interests include computer vision and deep learning.

Guanbin Li (Member, IEEE) received the Ph.D. degree from The University of Hong Kong in 2016. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University. His current research interests include computer vision, image processing, and deep learning. He has authorized or coauthorized on more than 60 papers in top-tier academic journals and conferences. He was a recipient of the ICCV 2019 Best Paper Nomination Award. He serves as an Area Chair of the Conference of VISAPP. He has been serving as a Reviewer for numerous academic journals and conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IJCV, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CYBERNETICS, CVPR, ICCV, ECCV, and NeurIPS.

Liang Lin (Senior Member, IEEE) was a Postdoctoral Fellow with the University of California at Los Angeles, Los Angeles, from 2008 to 2010. From 2014 to 2015, as a Senior Visiting Scholar, he was with The Hong Kong Polytechnic University and The Chinese University of Hong Kong. He is currently a Full Professor with Sun Yat-sen University. He is the Excellent Young Scientist of the National Natural Science Foundation of China. He leads the SenseTime Research and Development teams to develop cutting-edge and deliverable solutions on computer vision, data analysis and mining, and intelligent robotic systems. He has authored or coauthored more than 100 papers in top-tier academic journals and conferences. He is a fellow of IET. He was a recipient of the Best Paper Runners-Up Award at ACM NPAR 2010, the Google Faculty Award in 2012, the Best Paper Diamond Award at IEEE ICME 2017, and the Hong Kong Scholars Award in 2014. He has served as an Area/Session Chair of numerous conferences, including ICME, ACCV, and ICMR. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON HUMAN-MACHINE ASYSTEMS, *The Visual Computer*, and *Neurocomputing*.

Yizhou Yu (Fellow, IEEE) received the Ph.D. degree from the University of California at Berkeley in 2000. He was a Faculty Member with the University of Illinois at Urbana–Champaign for 12 years. He is currently a Chief Scientist with Deepwise AI Laboratory. His current research interests include computer vision, deep learning, biomedical data analysis, computational visual media, and geometric computing. He was a recipient of the 2002 U.S. National Science Foundation CAREER Award, the 2007 NNSF China Overseas Distinguished Young Investigator Award, and the ACCV 2018 Best Application Paper Award. He has served on the Editorial Board of *IET Computer Vision*, *The Visual Computer*, and IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS. He has also served on the Program Committee of many leading international conferences, including SIGGRAPH, SIGGRAPH Asia, and International Conference on Computer Vision.