Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 31(11): 6634–6656.
DOI: 10.3934/era.2023335
Received: 29 August 2023
Revised: 08 October 2023
Accepted: 09 October 2023
Published: 16 October 2023

Research article

EITGAN: A Transformation-based Network for recovering adversarial examples

Junjie Zhao[1], Junfeng Wu[2,*], James Msughter Adeke[1], Guangjie Liu[1] and Yuewei Dai[1,3]

[1] School of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China

[2] School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China

[3] Nanjing Center For Applied Mathematics, Nanjing 211135, China

* **Correspondence:** Email: wjf1916219959@outlook.com.

Abstract: Adversarial examples have been shown to easily mislead neural networks, and many strategies have been proposed to defend them. To address the problem that most transformation-based defense strategies will degrade the accuracy of clean images, we proposed an Enhanced Image Transformation Generative Adversarial Network (EITGAN). Positive perturbations were employed in the EITGAN to counteract adversarial effects while enhancing the classified performance of the samples. We also used the image super-resolution method to mitigate the effect of adversarial perturbations. The proposed method does not require modification or retraining of the classifier. Extensive experiments demonstrated that the enhanced samples generated by the EITGAN effectively defended against adversarial attacks without compromising human visual recognition, and their classification performance was superior to that of clean images.

Keywords: enhanced sample; Generative Adversarial Network; adversarial defense; deep learning; adversarial example; image processing

1. Introduction

Convolutional Neural Networks (CNNs) have been successfully applied to a wide range of computer vision tasks, including image clasification [1], object detection [2], and semantic segmentation [3]. However, recent studies have shown that CNNs can be deceived by images with meticulously designed small perturbations that are imperceptible to human vision [4, 5]. These perturbations that cause the CNN to misclassify an image into a different class are called adversarial perturbations, and the perturbed images are called adversarial examples. Adversarial examples pose a significant security

threat to the further adoption of advanced computer vision systems. Thus, addressing the influence of adversarial examples remains a challenging problem.

Many research have proposed strategies to defend against adversarial attacks [6, 7]. At present, adversarial defenses are mainly divided into two categories: Partial defense, which conducts early defense by only detecting adversarial examples [8], and complete defense. Complete defense is divided into model-specific and model-agnostic, resisting adversarial attacks by modifying classifiers or adversarial examples. Model-specific methods aim to regularize a specific model's parameters through adversarial training or parameter smoothing [9]. Such methods often require differentiable transformations, which not only consume high computation, but are also vulnerable to further attacks. Model-agnostic methods aim to remove adversarial perturbations from the input image domain by applying various transformations. These methods include Joint Photographic Experts Group (JPEG) compression [10], random image resizing & padding [11], total variance minimization [12], random Pixel Deflection (PD) [13], and image Super-Resolution (SR) [14]. Compared to model-specific methods, model-agnostic methods are simpler, faster, and more favorable. However, most model-agnostic methods lose part of the image information while removing adversarial perturbations, which decreases their classification performance on clean images.

In this study, we propose the Enhanced Image Transformation Generative Adversarial Network (EITGAN), an improved model-agnostic defense mechanism designed to generate enhanced samples that demonstrate superior classification performance compared to the original clean images. The defensive efficacy of the proposed EITGAN is not limited to image classification models. It can also be employed for adversarial defense and performance enhancement in other domains of image processing, such as medical image analysis models [15]. For example, Optical Coherence Tomography (OCT) images are frequently affected by noise and speckle [16, 17]. These types of noise are similar to adversarial noise, and EITGAN can help improve the performance of convolutional neural networks used in OCT image analysis.

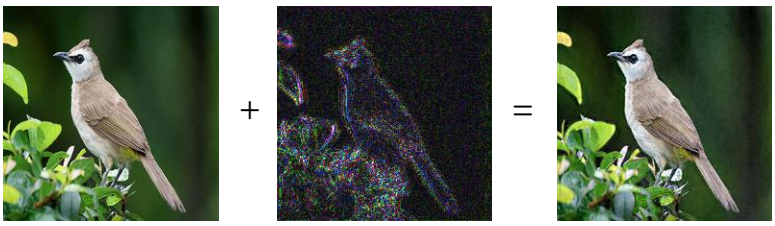

Figure 1. Example of an enhanced sample on the adversarial example.

Figure 1 illustrates an example of an enhanced sample on an adversarial example. The proposed EITGAN is used to generate enhanced samples with positive perturbations that can transform the labels of adversarial examples to the correct class. The generator of EITGAN consists of a super-resolution network and a noise network. The super-resolution network is used to generate super-resolution images to mitigate the effect of adversarial perturbations, and the noise network is used to generate positive perturbations to offset the influence of adversarial perturbations. We prove that the adversarial examples exhibit better performance than clean images after adding positive perturbations. Our main contributions are summarized as follows:

- A positive perturbation is employed to guide the attention of classifiers toward distinctive regions in the images that correspond to the correct class labels, rather than background regions.
- We propose the EITGAN to generate enhanced samples with positive perturbations, which can effectively resist adversarial attacks without affecting human vision recognition, and enhance the classification performance on transformed images higher than that of clean images.
- The proposed method is a model-agnostic defense that does not require modification or retraining of the target classifier. This can easily surpass other model-specific defenses.

The remainder of this paper is organized as follows. In Section 2, we present the related adversarial attacks and defense works that will be used in our work. In Section 3, the proposed EITGAN is introduced in detail, and the experimental results and analysis are presented in Section 4. Finally, the conclusion of this paper is provided in Section 5.

2. Related works

Here, we introduce several well-known adversarial attacks and defenses proposed in the literature that form the basis for our experiments. We only study model-agnostic defenses against non-targeted adversarial examples for image classification, although the same can be applied to other computer vision tasks.

2.1. Adversarial attack

Given a target classifier $F(\cdot)$ and a clean image x, y represents the ground-truth label corresponding to x. *Untargeted attacks* imply modifying a sample that was initially correctly classified, causing it to be randomly and inaccurately assigned to any erroneous category. For these attacks, the given adversarial perturbation ρ will make the adversarial example $\hat{x} = x + \rho$ look the same as the clean image, but the corresponding label $F(\hat{x}) \neq y$ is incorrect. *Targeted attacks* are similar, but they require to change the correct label into a specific incorrect label y_t. They seek $\hat{x}$ such that $F(\hat{x}) = y_t$ and $y_t \neq y$.

Next, we present a brief overview of several popular attacks against which we will evaluate our method.

Fast Gradient Sign Method (FGSM) [18] is a single-step attack that uses the sign of the gradient of the loss function ℓ w.r.t. the image to find the adversarial perturbation. For a given step size ϵ, FGSM is defined as Eq (2.1):

$$\hat{x} = x + \epsilon \cdot sign(\nabla_x \ell(x, y)) \tag{2.1}$$

Projected Gradient Descent (PGD) [4] is a variant of Basic Iterative Method (BIM [19]: an iterative version of FGSM) with uniform random noise as initialization, which is one of the most powerful first-order attacks. PGD projects the adversarial examples learned from each iteration into the L_p neighbor of clean images to constrain adversarial perturbation. The procedure of each iteration is as Eq (2.2):

$$x^{t+1} = \Pi_{x+S}(x^t + \alpha sign(\nabla_x \ell(x, y))) \tag{2.2}$$

where Π_{x+S} represents projecting the updated adversarial examples into the range from $x - S$ to $x + S$.

DeepFool (DF) [20] is an untargeted iterative attack that aims to minimize the L_2 norm between clean images and adversarial examples. This method approximates the classifier to a linear decision

boundary, and then looks for the smallest perturbation until the image crosses the boundary and is misclassified. The resulting perturbation is difficult for humans to detect.

Carlini and Wagner (C&W) [21] is an optimization-based attack that combines a differentiable surrogate of the model with a relaxation term to solve the perturbation minimization problem. The optimization is expressed as Eq (2.3):

$$\|x - \hat{x}\|_p + \lambda max(Z(\hat{x})_y - max\{Z(\hat{x})_{y_t} : y_t \neq y\}, -k) \tag{2.3}$$

where $Z(\hat{x})_{y_t}$ denotes the logit value (the output before the softmax layer) corresponding to class y_t, and k is the margin parameter.

2.2. Adversarial defense

Given a target classifier $F(\cdot)$ and an image $\tilde{x}$, which may be a clean image x or adversarial example $\hat{x}$. Adversarial defense is a method that aims to make the prediction $F(\tilde{x})$ on image $\tilde{x}$ equal to the one $F(x)$ on clean image x. The model-specific defense mentioned earlier modifies the classifier as $F'(\cdot)$ such that $F'(\tilde{x}) = F(x)$, and the model-agnostic defense uses transformation $G(\cdot)$ to change the image such that $F(G(\tilde{x})) = F(x)$. In this study, we focus on the study of the model-agnostic defense.

Recently, adversarial defenses against the input image transformation domain have been proposed. Luo et al. [22] proposed a foveation-based mechanism, which crops the image around the object with ground-truth coordinate data, then scales it back to the original size. The JPEG compression defense [10] removes adversarial perturbations by compressing high-frequency noise information that is invisible to the human eye. However, this method is effective only for very small perturbations. Guo et al. [12] proposed image transformation using quilting and Total Variance Minimization (TVM). Image quilting refers to replacing input image patches with similar patches drawn from a bank of images. However, image quilting alone is often insufficient. Therefore, it is combined with TVM, which minimizes the total variance by optimizing the construction of substitute images. Xie et al. [11] performed image transformation by randomly resizing and padding the images. The randomness property is also used in the work of Prakash et al. [13], which uses a wavelet-based pixel deflection transform to denoise the perturbation. Based on this, Mustafa et al. [14] proposed a method that combines the wavelet denoising and image super-resolution against adversarial attacks.

The main shortcoming facing most model-agnostic defenses is that the transformation degrades the quality of clean images, which leads to the loss of important information and decreases the classified performance.

3. EITGAN

This section presents the specific implementation details of the proposed enhanced model-agnostic defense against adversarial attacks. In Subsection 3.1, we introduce the overall architecture of the EITGAN and explain the implementation process in detail. In Subsection 3.2, we describe the design of the generator, which is the core component of EITGAN. In Subsection 3.3, we briefly introduce the design of the discriminator, which is also a part of EITGAN. Finally, we summarize the overall algorithm design of the EITGAN in Subsection 3.4.

3.1. Overview of EITGAN

In this work, EITGAN is used to transform adversarial examples and enhance them, and the same can also be applied to clean images. The proposed method can effectively resist adversarial attacks and improve the accuracy of the clean images. Figure 2 illustrates the overall architecture of EITGAN, which is mainly composed of the generator (G), the discriminator (D), and the target classifier (F). It should be noted that the target classifier is independent of EITGAN; therefore, the training process does not involve modifying the parameters of the target model. The classifier utilized in EITGAN is primarily employed to predict the label of the generated sample.

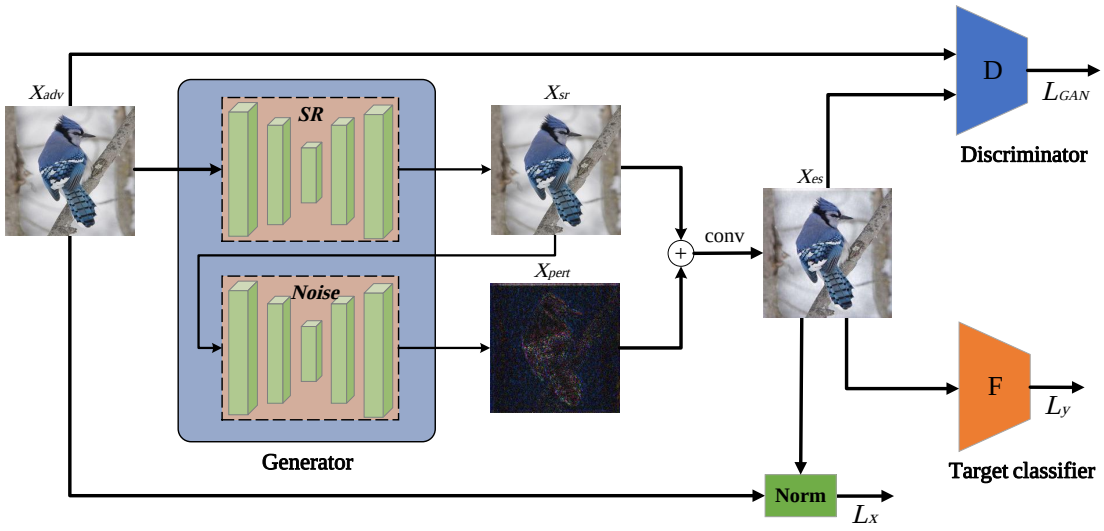

Figure 2. Overall architecture of the EITGAN.

As shown in Figure 2, the input of the model is the adversarial example X_{adv} and the output is the corresponding enhanced sample X_{es}. The generator of the EITGAN consists of two parts, the image SR network and a noise network ($Noise$). The SR network draws on the idea of Mustafa et al. [14], which uses image SR technology to generate recovered adversarial examples X_{sr}. However, the difference is that we do not just train an SR network, but also use it as a part of the generator and train it together with the noise network. The image generated by the SR network is not a common SR image, but an image of the same size as the input image. Keeping the output image size unchanged enables the image to better merge with later images while saving training resources and time. The $Noise$ network is used to generate positive perturbations X_{pert}, which can offset the influence of adversarial perturbations and enhance the performance of the classifier. Finally, the enhanced sample is generated by adding the SR image X_{sr} with the positive perturbation X_{pert} in a specific ratio.

The generation of enhanced samples can be expressed as Eq (3.1):

$$\begin{aligned} X_{es} &= G(X_{adv}) \\ &= conv(\alpha SR(X_{adv}) + \beta Noise(SR(X_{adv}))) \\ &= conv(\alpha X_{sr} + \beta Noise(X_{sr}))) \\ &= conv(\alpha X_{sr} + \beta X_{pert}) \end{aligned} \tag{3.1}$$

where α and β represent the ratios of SR images and positive perturbations, respectively. In this work, we set $\alpha = 0.6$ and $\beta = 0.4$, which are empirical values. *conv* in Eq (3.1) represents a 1×1 convolution layer. Because the features after simple addition exhibit relative independence, a 1×1 convolution is employed to facilitate interaction between channels, enhancing the fitting of these features.

For a given clean image X and its corresponding ground-truth label Y, when the input is an adversarial example, the sample enhanced by EITGAN can enable the target classifier to classify it into the correct category (i.e., $F(G(X_{adv})) = Y$ and $F(X_{adv}) \neq Y$). When the input is a misclassified clean image, EITGAN can also make it be classified correctly (i.e., $F(G(X)) = Y$ and $F(X) \neq Y$). The optimization loss L_{GAN}, L_x, and L_y, as shown in Figure 2, will be introduced in Subsection 3.4.

3.2. Generator of EITGAN

As the core component of the EITGAN, the generator mainly consists of the SR and *Noise* networks. Hence, we will provide a detailed introduction to the specific structures and parameters of these two networks.

3.2.1. SR network

Inspired by enhanced deep SR (EDSR) network [23], we design the SR network as shown in Figure 3. The SR network has twenty layers. The first layer is a down-sampling layer, the middle seventeen layers are composed of eight residual blocks and one convolutional layer, and the last two layers are up-sampling layers. In Figure 3, the solid line represents the implementation process of the network, and the dashed line represents the specific structure of the corresponding module. The three sets of data in the convolution brackets represent the input channel, output channel, and kernel size, respectively. Here, the kernel size of all convolution layers is 3×3.

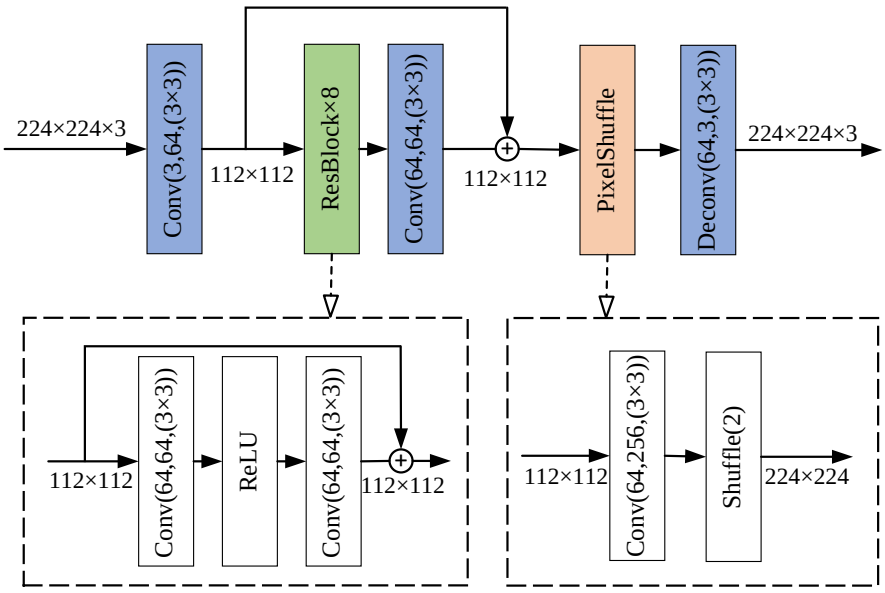

Figure 3. Architecture of the SR network.

The input samples are adversarial examples or clean images, which are sized 224 × 224 × 3. Figure 3 shows that the input samples are first passed through the down-sampling layer, which halves the sample size from 224 × 224 to 112 × 112. We use the method of down-sampling and then up-sampling to avoid the convolution of high-resolution images, thereby reducing the computational complexity. Subsequently, they are inputted into eight residual blocks and one convolutional layer, which keeps the sample size and channels unchanged. The output is then added to the output of the previous down-sampling and inputted into the up-sampling layer. We use pixelshuffle as the up-sampling, which is adopted from the Efficient Sub-Pixel Convolutional Neural Network (ESPCN) [24]. The pixelshuffle is mainly composed of a convolution layer and a shuffle operation. Assuming that the channel of the input sample is C, the size is $H \times W$, and the upsampling factor is r, the convolution operation reshapes the sample as $(r^2C,\ H,\ W)$, and the shuffle operation further reshapes the sample as $(C,\ rH,\ rH)$. In this work, we set the upsampling factor $r = 2$, doubling the size of the sample and resulting in a sample size of 224 × 224. Finally we pass the output of pixelshuffle into the deconvolution layer to reduce the number of channels from 64 to 3, thereby obtaining an SR image with the same size as the input sample.

3.2.2. Noise network

The structure of the *Noise* network is shown in Figure 4, which is divided into three parts. The top, middle, and bottom parts represent the encoder, residual block, and decoder, respectively. The *Noise* network has sixteen layers. Both the encoder and decoder consist of four layers, while the middle section incorporates four residual blocks, each comprising two layers.

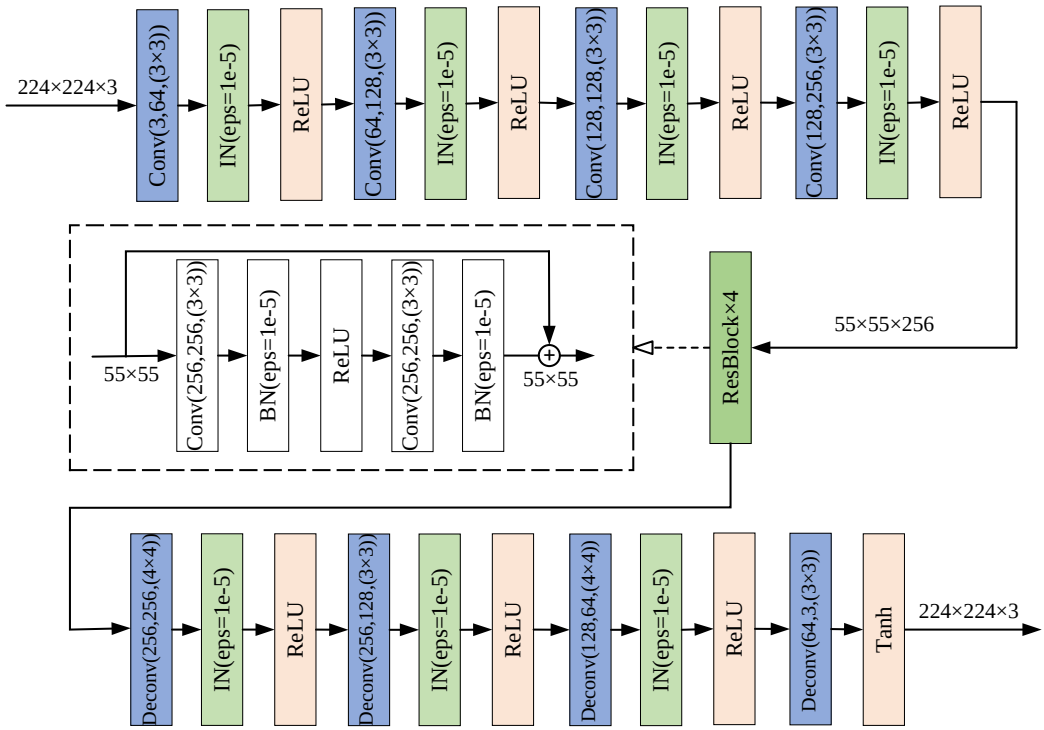

Figure 4. Architecture of the noise network.

The input of the noise network is the output of the SR network, which has the same shape as the original sample ($224 \times 224 \times 3$). As shown in Figure 4, the input samples are encoded first. The encoder is composed of four convolutions, each followed by Instance Normalization (IN) and Rectified Linear Unit (ReLU), and the convolution kernel size is 3×3. After the first and third convolution layers, the sample size was reduced by half. The second layer reduces the sample size by two pixels, and the fourth layer keeps the size unchanged. During encoding, the sample channels increase exponentially, resulting in the encoded samples sizing at $55 \times 55 \times 256$. Subsequently, the encoded samples are inputted into four residual blocks for residual convolution, and each group of residual blocks maintains the sample size and channels unchanged. It is worth mentioning that the residual block differs from that in the SR network. Batch Normalization (BN) is used after each convolution layer to accelerate the network convergence. The output of the residual convolution is then inputted into the decoder for decoding. The decoder's structure mirrors that of the encoder, each comprising four convolutional layers. However, the decoder employs deconvolution, with the final deconvolution layer being succeeded by the hyperbolic tangent function (Tanh). The kernel sizes of the first and third layers of the decoder are both 4×4, and those of the second and fourth layers are both 3×3. After the first and third deconvolution layers, the sample size is doubled, the second layer increases the sample size by two pixels, and the fourth layer keeps the sample size unchanged. During decoding, the sample's channels decrease exponentially, resulting in positive perturbations that have the same shape as the input samples of $224 \times 224 \times 3$.

3.2.3. Generation of enhanced samples

It is not sufficient to only use SR images generated by the SR network or positive perturbations generated by the *Noise* network. Therefore, the positive perturbations are added to the SR images. The positive perturbations can neutralize adversarial perturbations in SR images and further enhance their classification performance. In the EITGAN, SR images focus on improving the visual quality, and positive perturbations focus on improving the classification performance. To avoid affecting the visual quality of the enhanced samples after addition, the ratio of the SR image is set to 0.6, and the ratio of the positive perturbation is set to 0.4. However, the features after simple addition are relatively independent, so a layer of 1×1 convolution is used to make the features interact between channels, thereby enhancing the expression of features and obtaining the final enhanced samples.

3.3. Discriminator of EITGAN

As a part of EITGAN, the main purpose of the discriminator is to conduct adversarial learning with the generator, and to identify as accurately as possible whether or not the input sample is the original sample. We refer to the encoder in the *Noise* network to design the discriminator, which has five layers, including four convolutional layers and one fully connected layer, as shown in Figure 5. The upper part of Figure 5 shows the output results of each convolutional layer, and the lower part shows the specific structure of the convolutional network.

The inputs of the discriminator are the original samples and the corresponding enhanced samples. As shown in Figure 5, after each convolutional layer, the sample size is reduced twice. Except for the first layer, BN and LeakyReLU with a negative slope of 0.2 are performed after each convolution layer. Finally, the output of the convolution is inputted into the fully connected layer to obtain the

value of one neuron, and sigmoid activation is performed to constrain the value between 0 and 1. In the discriminator, an output value of 0 is judged as fake, and the value of 1 is judged as real.

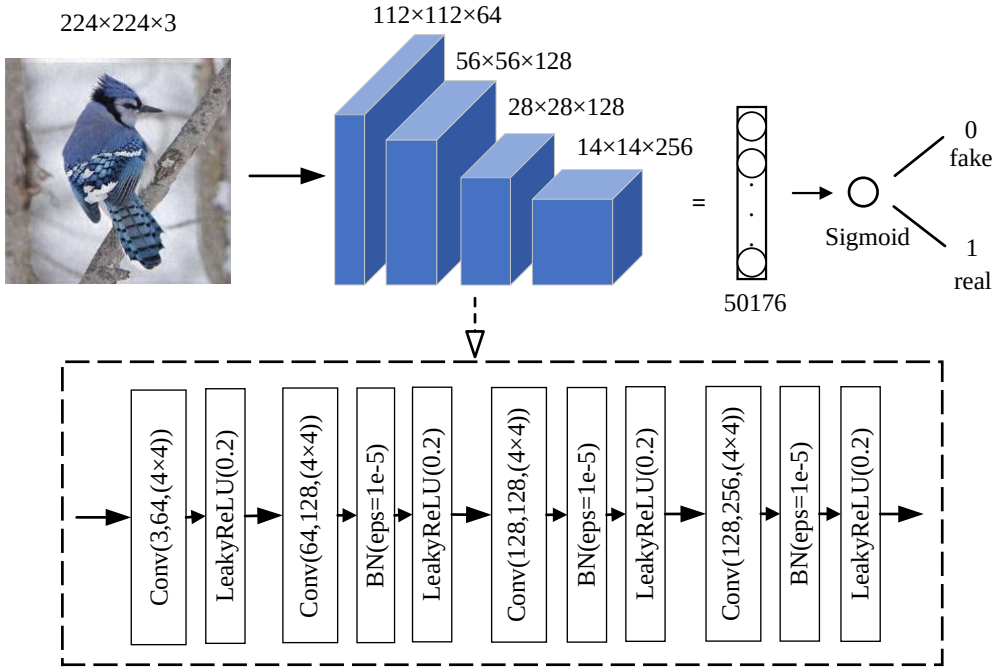

Figure 5. Architecture of the discriminator.

3.4. Algorithm description

An algorithm description of the proposed enhanced defense scheme is provided in Algorithm 1. As depicted in Algorithm 1, the implementation of EITGAN is mainly divided into two parts: The generation process and the training process. The generation process has been introduced in Subsection 3.2, so we only discuss the training process here. During the training process, when the number of training epochs was set to 20, the results were better and more stable. Therefore, in order to save training time, the training epoch is set to 20.

The overall optimization function for training the EITGAN is shown in Eq (3.2):

$$L(G, D, F) = L_{GAN} + \lambda L_x + \mu L_y \tag{3.2}$$

where L_{GAN} represents the GAN loss, which comprises of $L(D)$ (line 9) and $L(G)$ (line 11) in Algorithm 1. L_x (line 12) and L_y (line 13) represent pixel loss and category loss, respectively. The λ and μ in Eq (3.2) denotes the proportional coefficients corresponding to L_x and L_y. In this work, in order to enhance the classified performance of samples, we set $\lambda = 0.01$ and $\mu = 100$. The optimization of L_{GAN} can be described as Eq (3.3):

$$\begin{aligned} L_{GAN} &= \min_G \max_D L(D, G) \\ &= E_{X_{adv}}[logD(X_{adv}) + log(1 - D(G(X_{adv})))] \end{aligned} \tag{3.3}$$

The GAN loss L_{GAN} makes D and G play the minimax game to ensure that the generator can generate samples as realistically as possible, so that the discriminator cannot identify whether it is the original sample. The pixel loss L_x uses the L_2 norm to minimize the distance between the original sample and the generated sample, so that the generated sample looks the same as the original sample. The category loss L_y uses the cross-entropy loss l_{CE} to minimize the distance between the target classifier's predicted label of the generated sample and the ground-truth label, so that the predicted label can be closer to the ground-truth label. Undergoing 20 epochs training, we get the final generator that can generate the enhanced samples to be classified correctly.

Algorithm 1 Implementation of EITGAN

Input: The adversarial examples X_{adv}

Output: The enhanced samples X_{es}

Given clean image X and ground-truth label Y;
for *number of training epochs* **do**
  // Generation process
  generate the super-resolution image of the adversarial example: $X_{sr} = SR(X_{adv})$
  generate the positive perturbation of the super-resolution image: $X_{pert} = Noise(X_{sr})$
  generate enhanced samples with X_{sr} and X_{pert}: $X_{es} = conv(\alpha X_{sr} + \beta X_{pert}) = G(X_{adv})$
  // Training process
  update the discriminator D with parameters θ_d:
  $maxL(D) = \nabla_{\theta_d} E_{X_{adv}}[logD(X_{adv}) + log(1 - D(G(X_{adv})))]$
  update the generator G with parameters θ_g:
  $minL(G) = \nabla_{\theta_g} E_{X_{adv}}[log(1 - D(G(X_{adv})))]$
  $minL_x = \nabla_{\theta_g} E_{X_{adv}} \|X_{adv} - G(X_{adv})\|_2$
  $minL_y = \nabla_{\theta_g} E_{X_{adv}}[l_{CE}(F(G(X_{adv})), Y)]$
end for
return X_{es}

4. Experiments and analysis

In this section, we prove the existence of positive perturbations through experiments and verify the feasibility and effectiveness of the proposed method. In Subsection 4.1, the experimental setup is introduced in details. The experiment in Subsection 4.2 evaluates the performance of EITGAN on different attacks and classifiers. The experiment in Subsection 4.3 compares EITGAN with the state-of-the-art model-agnostic defense methods. The experiment in Subsection 4.4 evaluates the generalization performance of the EITGAN across different attacks and classifiers. Subsection 4.5 evaluates the performance of EITGAN on ImageNet-A.

4.1. Experimental setup

Datasets: Our experiments are performed on a dataset with 30,000 training images and 20,000 test images. This dataset is randomly chosen from ImageNet [25] and corresponds to 5 classes. Each class contains 10,000 images, and all these images have a size of $224 \times 224 \times 3$. We choose 5 classes to reduce training costs and ensure high accuracy.

Networks: To evaluate the proposed EITGAN, we use Inception-V3 [26], ResNet-50 [27] and Inception ResNet-V2 [28] as the target classifiers. These classifiers are trained on our dataset and their parameters are saved for training EITGAN. To better evaluate the EITGAN, we set the classifiers' accuracy on the test images to 80%. While training the EITGAN, we refrain from conducting any re-training or fine-tuning on these classifiers.

Attacks: We use four adversarial attacks to generate adversarial examples, including FGSM [18], PGD [29], DF [4], and C & W [20]. For FGSM, we set the step size as $\epsilon = 2$. PGD divides a single-step attack into multiple small-step attacks, where we set the small step size as $\alpha = 2$, the iteration as $t = 20$, and the maximum step size is restricted to 8. DF is a non-parametric attack that optimizes the amount of perturbation to misclassify an image. For C & W, we set the margin parameter as $k = 0$. All adversarial examples were generated for undefended classifiers.

Defenses: We compare the proposed EITGAN with a number of recently introduced state-of-the-art model-agnostic defense methods. These include JPEG compression [10], random image resizing & padding (Resize & Pad) [11], TVM [12], random PD [13], and image SR [14]. All experiments run on the same dataset and against the same attacks for a fair comparison.

Metrics: In the experiments, well-known metrics such as accuracy, recall, and precision [30,31] are used to evaluate the performance of the proposed model. Accuracy is used to represent the proportion of correct samples among all samples, as shown in Eq (4.1). Recall is the proportion of correctly classified samples in the positive class and can be calculated using Eq (4.2). Precision represents the proportion of correct in the predicted positive class and is obtained using Eq (4.3).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

In Eqs (4.1)–(4.3), TP, FP, TN and FN represent true positive, false positive, true negative and false negative, respectively. In this work, TP refers to the samples that are correctly recognized as the current class. The sum of TP and FP represents all samples recognized as the current class. The sum of TP and FN represents all the samples of the current class.

Environment: The hardware environment used in the experiments of this paper includes an Nvidia 2080Ti Graphic Processing Unit (GPU), a Ryzen 3600X Central Processing Unit (CPU), and 32 GigaByte (GB) Dual Data Rate 4 (DDR4) memory. The software environment comprises Windows 10, Python 3.8, and PyTorch 1.7.

4.2. Performance evaluation

4.2.1. Results and analysis

Table 1 shows the performance of the enhanced samples generated by the EITGAN on different classifiers and adversarial attacks. The first column in Table 1 lists the target classifiers. The second column shows different adversarial attacks, among which the 'None' indicates that no adversarial attack is performed. The third and fourth columns show the accuracy of the original sample and the

corresponding enhanced sample, respectively. The fifth column indicates the improved accuracy of the enhanced sample. The last two columns represent the PSNR values of the enhanced sample X_{es} and original sample (clean image X or adversarial example X_{adv}), respectively.

As shown in Table 1, the enhanced accuracy for the adversarial examples generated by FGSM is the highest across the three classifiers, which are 97.5%, 93.4%, and 94.1%, respectively. The improved accuracy of the adversarial examples generated by PGD is better. The last two columns show that although the PSNR of the enhanced sample is lower than that of the original sample, it generally meets the image evaluation standard. It should be noted that all PSNRs are compared to the original clean images. As shown in Table 1, EITGAN can improve not only the accuracy of adversarial examples as well as the accuracy of clean images. It should be emphasized that the accuracy of the enhanced recovered adversarial examples is higher than that of the original clean images. Compared with the results of directly processing clean images, the results of processing the corresponding adversarial examples are better. This indicates that the positive perturbation generated after merging with the adversarial perturbation can better highlight the characteristics of the target discriminative regions.

Table 1. The performance of enhanced samples generated by EITGAN on different classifiers and adversarial attack methods.

Classifiers	Attacks	Original accuracy	Enhanced accuracy	Improved accuracy	PSNR (X_{es})	PSNR (X/X_{adv})
Inception-V3	None	80.0%	82.9%	2.9%	23.188	100.000
	FGSM	4.6%	97.5%	92.9%	29.459	40.084
	PGD	0.0%	92.7%	92.7%	27.015	34.165
	Deepfool	0.4%	83.9%	83.5%	26.663	60.909
	C & W	2.5%	84.0%	81.5%	24.605	51.393
ResNet-50	None	80.0%	85.0%	5.0%	26.477	100.000
	FGSM	19.2%	93.4%	74.2%	27.793	40.047
	PGD	0.1%	90.4%	90.3%	26.089	34.380
	Deepfool	0.6%	86.8%	86.2%	26.812	53.025
	C & W	16.4%	84.6%	68.2%	27.033	56.943
Inception ResNet-V2	None	80.0%	83.5%	3.5%	28.627	100.000
	FGSM	10.7%	94.1%	83.4%	28.974	40.088
	PGD	0.2%	90.8%	90.6%	25.762	34.684
	Deepfool	0.6%	84.2%	83.6%	26.845	53.136
	C & W	14.4%	84.2%	69.8%	27.776	56.541

To better analyze the performance of the EITGAN, we evaluate the precision and recall rates of the enhanced samples in Tables 2 and 3, respectively. The structure of Table 2 is the same as that of Table 3. The first three columns represent classifiers, adversarial attack methods, and defense methods, respectively. 'None' indicates that no processing is performed. 'Class 0' to 'Class 4' respectively represent the labels corresponding to the five classes. We use Class N (N: 0–4) to represent the column of 'Class 0' to 'Class 4'. The data in Table 2 indicates the probability that all samples predicted to

be class N are actually class N, and the calculation is described in Eq (4.3). The data in Table 3 indicates the probability that all samples that are actually in class N are predicted to be in class N, and the calculation is shown in Eq (4.2). The data in Tables 2 and 3 are calculated based on the EITGAN obtained from the 20th training epoch. The last column shows the overall accuracy of the corresponding data in each row. Bold font indicates the data of clean images that have not been attacked or defended.

Table 2 shows that the distribution of the enhanced sample is relatively balanced across the five classes, thus explaining that the high accuracy is not because one of the classes is over-recognized and proving that the experiments are of practical significance. The last column of Table 2 shows that the overall accuracy of enhanced samples generated by EITGAN is higher than that of the original clean images, which corresponds to the data in Table 1. In general, the precision of Inception-V3 for class 0 samples is higher, ResNet-50 has better precision for class 1 samples and the precision of Inception ResNet-V2 for the five classes is relatively average. As shown in Table 3, since the recall rate for the class 1 sample of the original clean images is the lowest, the recall rate of the class 1 sample generated by EITGAN is relatively low. Besides, the overall recall rate of the enhanced samples is higher than that of the original samples. In addition, since the number of TP in Table 3 is the same as in Table 2, the overall accuracy of both is equal. Table 3 is similar to Table 2, both of which verify the validity of our experiments and reflect the feasibility and high performance of the EITGAN. It can be concluded from Tables 2 and 3 that EITGAN can not only effectively resist adversarial attacks but also further enhance the classified performance of samples, which has considerable practical significance.

Table 2. The *Precision* (%) of enhanced samples generated by EITGAN.

Classifiers	Attacks	Defense	Class 0	Class 1	Class 2	Class 3	Class 4	Accuracy
Inception-V3	None	None	**87.9**	**81.4**	**74.4**	**86.7**	**72.7**	**80.0**
	None	EITGAN	88.5	85.0	81.9	86.9	70.0	81.6
	FGSM		97.8	96.4	97.6	98.2	97.0	97.4
	PGD		94.1	94.7	92.8	92.3	89.9	92.7
	Deepfool		90.3	87.2	82.2	88.5	71.6	83.1
	C & W		89.3	84.0	83.7	86.8	72.3	82.6
ResNet-50	None	None	**70.1**	**91.1**	**85.4**	**80.3**	**80.3**	**80.0**
	None	EITGAN	80.8	88.6	86.2	88.6	78.5	84.2
	FGSM		91.5	96.1	96.1	92.4	90.6	93.2
	PGD		91.7	94.5	87.0	93.6	85.7	90.3
	Deepfool		88.9	85.4	87.2	89.0	83.5	86.8
	C & W		85.1	88.6	86.4	87.2	77.0	84.6
Inception ResNet-V2	None	None	**75.0**	**80.7**	**78.5**	**89.0**	**79.2**	**80.0**
	None	EITGAN	86.4	84.6	74.7	90.7	77.8	82.2
	FGSM		96.2	92.0	92.1	95.6	93.4	93.8
	PGD		88.8	88.7	91.8	93.4	91.1	90.7
	Deepfool		93.1	92.6	72.6	85.2	74.5	82.3
	C & W		85.7	89.0	83.8	86.9	76.7	84.1

Table 3. The *Recall* (%) of enhanced samples generated by EITGAN.

Classifiers	Attacks	Defense	Class 0	Class 1	Class 2	Class 3	Class 4	Accuracy
Inception-V3	None	None	**78.0**	**68.0**	**93.1**	**82.5**	**79.2**	**80.0**
	None	EITGAN	79.3	68.1	89.6	83.4	87.9	81.6
	FGSM		97.2	97.1	98.3	97.5	96.8	97.4
	PGD		92.4	87.5	94.7	95.0	94.0	92.7
	Deepfool		82.1	71.7	90.1	83.8	88.2	83.1
	C & W		79.7	72.3	89.3	84.4	87.7	82.6
ResNet-50	None	None	**93.1**	**58.4**	**84.6**	**89.5**	**74.5**	**80.0**
	None	EITGAN	90.8	74.9	88.2	84.4	82.8	84.2
	FGSM		97.8	89.0	91.2	95.1	93.0	93.2
	PGD		93.9	82.8	93.7	90.6	90.4	90.3
	Deepfool		90.7	85.8	87.7	87.5	82.2	86.8
	C & W		89.7	76.1	86.9	85.8	84.6	84.6
Inception ResNet-V2	None	None	**90.0**	**72.9**	**87.2**	**75.3**	**75.1**	**80.0**
	None	EITGAN	85.8	74.8	92.0	77.1	81.7	82.2
	FGSM		97.0	95.0	96.0	89.5	91.7	93.8
	PGD		95.0	91.6	89.5	86.7	90.7	90.7
	Deepfool		83.2	71.9	90.7	81.4	84.3	82.3
	C & W		89.2	75.5	86.4	84.1	85.4	84.1

To further diversify the experimental scenarios, we also test EITGAN with Adversarial Transformation Network (ATN) [32] and Adversarial Example Generative Adversarial Network (AdvGAN) [33]. Table 4 presents the obtained test accuracy. In the table, "None" denotes the direct use of adversarial samples, while "EITGAN" indicates the use of samples generated by EITGAN. The experimental data demonstrates that EITGAN continues to exhibit significant defense capabilities against GAN-based adversarial attack methods (ATN and AdvGAN).

Table 4. Test accuracy on ATN and AdvGAN.

Attacks	ResNet-50		Inception-V3		Inception ResNet-V2	
	None	EITGAN	None	EITGAN	None	EITGAN
ATN	21.9%	80.5%	20.4%	90.3%	22.0%	74.6%
AdvGAN	21.1%	85.9%	5.7%	84.2%	18.1%	76.8%

4.2.2. Grad-CAM visualization

The Class Activation Map (CAM) [34] is a weakly supervised localization technology that helps explain the prediction of the CNN model by providing visualization of the discriminative area in the image. The CAM needs to replace the last fully connected layer with a Global Average Pooling (GAP) layer, which modifies and retrains the most existing model structure. To reduce training costs, we choose to use Grad-CAM [35], an improved version of the CAM. Grad-CAM calculates the weight

through the global average of the gradient and then sums the feature maps with weights. Subsequently, the ReLU is used to consider only the pixels that have a positive impact on the target class. Finally, the corrected sample is up-sampled to the size of the original image and superimposed on it to obtain the required heat maps. The heat maps range from blue to red, with red indicating a higher level of attention and a more significant impact on the results. Figure 6 shows the Grad-CAM of the prediction by Inception-V3 on the clean, attacked, and recovered images. The clean images in the upper half of Figure 6 are classified correctly, and those in the lower half are classified incorrectly. The CAM in Figure 6 indicates Grad-CAM.

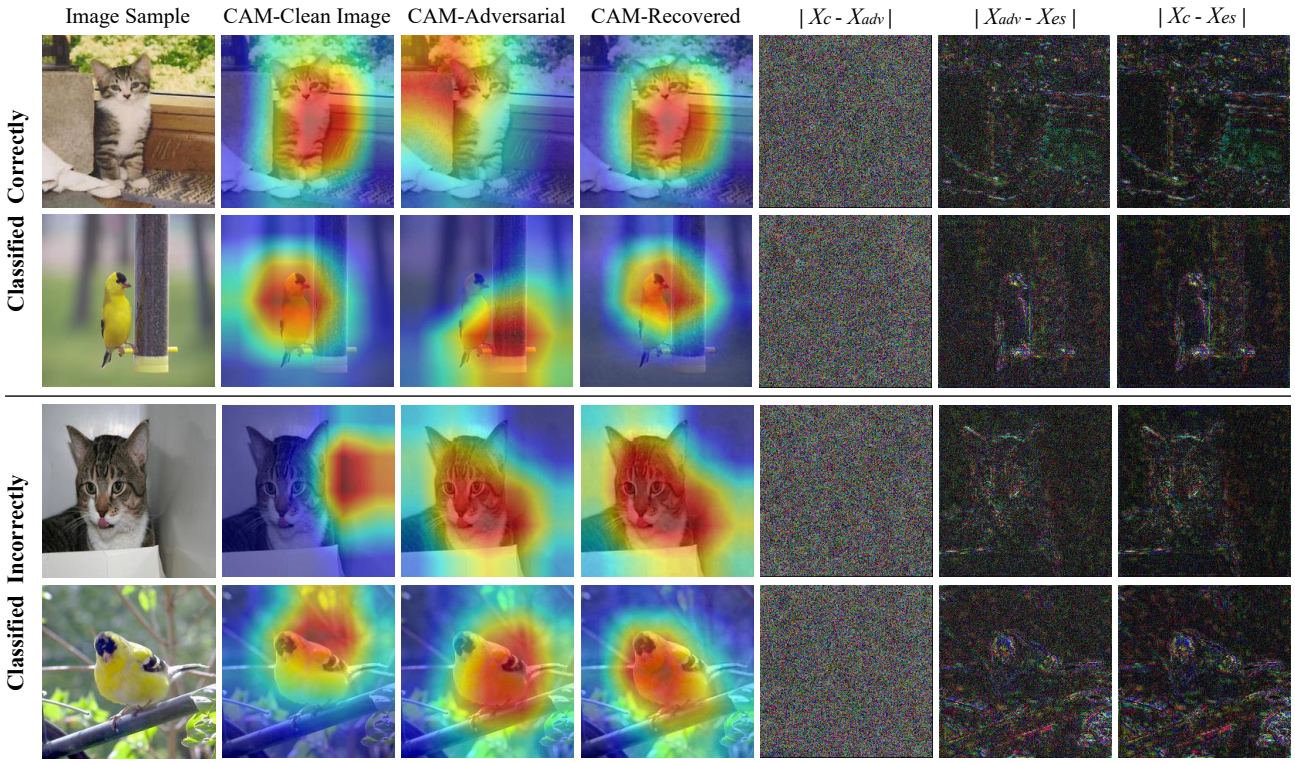

Figure 6. Visualization of EITGAN against PGD attack on clean images that are classified correctly and incorrectly. The first column shows the clean images. The subsequent three columns show the CAM on clean, PGD-attacked, and recovered images. The fifth column shows the perturbations (magnified by 20x) added to the clean image by PGD and the sixth column shows the perturbations (magnified by 5x) added to the adversarial example by EITGAN. The last column shows the difference between the clean and recovered images (magnified by 5x).

As shown in Figure 6, column 5 shows the perturbations added to the clean images by the PGD attack, which are called adversarial perturbations. Column 6 shows the perturbations added to the adversarial examples by EITGAN, which we refer to as positive perturbations. Column 7 presents the result of combining positive perturbations with adversarial perturbations. Comparing the combined perturbation with the positive perturbation, its distinctive regions are more prominent, and the random noise is lower. It can be observed from Figure 6 that EITGAN can recover the CAM on the

adversarial examples to be consistent with that on the clean images, when the clean images are classified correctly. In the lower half, when the clean images are classified incorrectly, the EITGAN can also recover the CAM on the adversarial examples to the discriminative regions corresponding to the correct class labels. It can be concluded that positive perturbations can effectively offset the impact of adversarial perturbations and make the CAM on the recovered images consistent with that of the target discriminative regions. The enhanced samples selectively add positive perturbations, which effectively neutralize the adversarial perturbation and eventually help in recovering the model attention toward discriminative regions corresponding to the correct class labels.

4.3. Comparison and analysis

Table 5. The accuracy (%) of different defense method.

Classifiers	Clean	FGSM	PGD	DeepFool	C&W
No Defense					
Inception-V3	80.0	4.6	0.0	0.4	2.5
ResNet-50	80.0	19.2	0.1	0.6	16.4
Inception ResNet-V2	80.0	10.7	0.2	0.6	14.4
JPEG (Das et al. [10])					
Inception-V3	79.2	51.8	54.2	75.5	76.8
ResNet-50	79.1	41.7	35.3	30.2	74.5
Inception ResNet-V2	77.9	48.6	49.7	32.9	75.2
Resize & Pad (Xie et al. [11])					
Inception-V3	78.0	63.8	57.3	74.8	75.6
ResNet-50	77.5	58.0	48.9	56.0	75.0
Inception ResNet-V2	79.9	57.7	46.1	55.7	77.4
TVM (Guo et al. [12])					
Inception-V3	75.8	46.5	44.3	71.8	72.5
ResNet-50	72.2	41.4	26.8	32.5	67.7
Inception ResNet-V2	74.4	36.9	26.2	25.9	69.4
PD (Prakash et al. [13])					
Inception-V3	66.3	44.9	46.9	62.4	63.4
ResNet-50	64.0	42.4	37.1	35.9	60.6
Inception ResNet-V2	62.7	45.8	46.9	36.3	60.7
SR (Mustafa et al. [14])					
Inception-V3	78.4	51.6	53.2	74.7	75.7
ResNet-50	78.4	44.5	39.5	38.1	74.6
Inception ResNet-V2	77.9	50.8	53.3	38.7	75.5
EITGAN (Ours)					
Inception-V3	82.9	97.5	92.7	83.9	84.0
ResNet-50	85.0	93.4	90.4	86.8	84.6
Inception ResNet-V2	83.5	94.1	90.8	84.2	84.2

We compare EITGAN with various defense mechanisms, as shown in Table 5. The first column shows the target classifier, and the second column shows the accuracy of the clean images. The third to sixth columns represent the accuracy of the adversarial examples generated by the four adversarial attack methods. 'No Defense' in Table 5 shows the performance of classifiers on original clean images and adversarial examples.

In Table 5, we compare the five defense mechanisms. It can be observed that these contrast defenses have a certain recovery effect on adversarial examples, and the recovered results for C & W are relatively good compared with those of the other three adversarial attacks. However, the accuracy of clean images decreases after these defenses, and the accuracy of the recovered samples still has a certain gap compared to that of clean images. In comparison, the proposed EITGAN can improve not only the accuracy of adversarial examples but also clean images. It should be emphasized that the accuracy of the enhanced samples exceeds that of the original clean images, which represents a great advantages compared with other model-agnostic defenses. Table 5 shows that EITGAN has better recovery results for adversarial examples generated by FGSM, and the overall recovery performance of the Inception-V3 model is better.

Figure 7. Comparison of recovered images on different defenses against FGSM attack. The first column shows five clean images corresponding to the five classes. The next five columns display the recovered samples of the five contrasting approaches. The last column show the enhanced samples generated by EITGAN.

Table 5 shows that EITGAN outperforms the contrast defense methods. In order to further verify the superiority of the proposed method, we compare the recovered images generated by the contrast defenses against the FGSM attack under the Inception-V3 model, as shown in Figure 7. It can be seen from Figure 7 that the sample after 'Resize & Pad' has black borders, the sample after 'TVM' is relatively blurry, and the sample after 'PD' has apparent noise. The samples generated by EITGAN and the other defense methods (JPEG compression, image SR) are similar to the clean images. It can be proven that the proposed EITGAN does not affect the visual quality of the sample while enhancing the accuracy of the recovered samples.

4.4. Evaluation across attacks and classifiers

To better reflect the performance of the EITGAN, we separately evaluate its generalization ability across different attacks and classifiers.

Table 6. The accuracy (%) of EITGAN across different classifiers.

Classifiers	Defense	Inception-V3					ResNet-50					Inception ResNet-V2				
		Clean	FGSM	PGD	DF	C & W	Clean	FGSM	PGD	DF	C & W	Clean	FGSM	PGD	DF	C & W
Inception-V3	✖	80.0	4.6	0.0	0.4	2.5	83.3	82.4	79.5	81.3	83.2	81.3	80.3	77.9	78.3	81.1
	✔	82.9	97.5	92.7	83.9	84.0	79.7	82.3	80.1	79.2	80.5	79.5	78.6	76.5	75.8	78.3
ResNet-50	✖	80.7	79.6	75.7	80.6	80.6	80.0	19.2	0.1	0.6	16.4	78.8	76.7	73.6	74.3	78.6
	✔	76.3	79.9	74.0	77.8	79.8	85.0	93.4	90.4	86.8	84.6	77.9	79.0	75.8	73.2	76.7
Inception	✖	85.1	84.6	81.3	85.1	85.1	85.9	84.7	80.7	83.3	85.7	80.0	10.7	0.2	0.6	14.4
ResNet-V2	✔	81.7	82.4	81.7	81.5	79.9	85.2	83.7	82.4	80.2	84.0	83.5	94.1	90.8	84.2	84.2

Table 7. The accuracy (%) of EITGAN across different attacks.

Train \ Test	Clean	FGSM	PGD	DF	C & W	Clean	FGSM	PGD	DF	C & W	Clean	FGSM	PGD	DF	C & W
	Inception-V3					ResNet-50					Inception ResNet-V2				
None	80.0	4.6	0.0	0.4	2.5	80.0	19.2	0.1	0.6	16.4	80.0	10.7	0.2	0.6	14.4
Clean	82.9	73.9	68.9	80.8	81.0	85.0	39.3	25.4	37.5	79.6	83.5	34.2	23.7	32.4	76.7
FGSM	80.2	97.5	91.2	83.7	83.2	81.4	93.4	86.8	83.3	83.1	80.1	94.1	89.1	82.2	82.2
PGD	79.5	96.3	92.7	82.7	82.1	81.3	92.7	90.4	81.1	83.2	78.7	92.9	90.8	81.8	80.7
DeepFool	81.3	88.4	74.9	83.9	82.4	83.4	92.7	86.4	86.8	85.4	75.9	92.5	81.1	84.2	80.1
C&W	81.6	87.5	75.3	82.8	84.0	83.9	87.8	73.8	81.5	84.6	82.3	91.7	68.9	76.4	84.2

We evaluate the generalization performance of EITGAN across different classifiers as shown in Table 6. The first column represents the target classifiers. The first row demonstrates the target classifiers for the generalization experiments, and the second row represents the clean images and adversarial examples corresponding to the current classifier. The second column shows the state of the current sample. The row of '✖' indicates the accuracy of the original sample, and the row of '✔' indicates the accuracy of the enhanced sample generated by EITGAN. It can be seen from Table 6 that the generalization ability of the original adversarial examples between models is relatively poor, which directly affects the generalization of the proposed method between models. However, the application of EITGAN does not have a major impact on the original sample, and some results are better than the original. For example, the result of Inception-ResNet-V2 on the enhanced sample,

which is generated by EITGAN trained on ResNet50 under PGD attack, is 1.7% higher than that of the original sample.

Table 7 shows the generalization performance of the EITGAN across different attacks. The first column represents different training sets used to train the EITGAN. The first row represents different test sets, where adversarial examples are generated by FGSM, PGD, DF, and C & W. We evaluate the EITGAN performance on three target classifiers. The results in each row of Table 7 indicate the accuracy of enhanced samples generated on different test sets by EITGAN, and EITGAN is trained on the training set where the row is located. The row of 'None' represents the result without the EITGAN. As shown in Table 7, the overall generalization performance of EITGAN is better, and each result across attacks is higher than the accuracy of the original clean image. The EITGAN trained on different training sets all have the best generalization performance on the FGSM test set, and the EITGAN trained on the FGSM dataset also has the better generalization performance on different test sets. The overall experiments show that the proposed method not only resists adversarial attacks, but also generalizes well.

4.5. Application on more classes

We also do experiments on ImageNet-A [36], a dataset of natural adversarial examples that fool current ImageNet classifiers. We use the well-trained Inception-V3 [26], ResNet-50 [27] and Wide ResNet-50 [37] as the target classifiers. These classifiers are trained using ImageNet-1000. Table 8 compares the performance of EITGAN with other defense methods on ImageNet-A.

Table 8. The accuracy (%) of different defenses on ImageNet-A.

	ResNet-50	Inception-V3	Wide ResNet-50
None	0.4	0.9	0.3
JPEG	0.3	0.9	0.3
Resize & Pad	0.4	0.6	0.4
TVM	0.2	0.4	0.4
PD	0.2	0.5	0.2
SR	0.3	1.0	0.2
EITGAN	2.5	2.1	2.6

As shown in Table 8, the first column lists different defense methods, where 'None' means no defense. The first row lists three well-trained target classifiers. The data in Table 8 is the recognition accuracy of classifiers on samples processed by different defense methods. It can be seen from Table 8 that most of the five defenses compared cannot improve the accuracy of the samples. Even if they could, the improvement effect is limited, and the maximum is approximately 0.1%. However, EITGAN can improve the accuracy of the three target classifiers, and the highest can be improved by 2.3%. This shows that EITGAN can also be applied to more classes and has better performance than other model-agnostic defenses.

5. Conclusions

Motivated by problems such as CNN being vulnerable to adversarial attacks and most model-agnostic defenses decreasing the accuracy of clean images, we proposed an enhanced defense mechanism EITGAN. The EITGAN is also a model-agnostic defense, which does not need to modify or retrain the target classifier. In this work, we used image SR to mitigate the effect of adversarial perturbations, as well as positive perturbation to further enhance the classified performance of the recovered sample. Extensive experiments showed that the proposed EITGAN outperformed the state-of-the-art defenses, which cannot only improve the accuracy of clean images but also the accuracy of recovered adversarial examples higher than that of the original clean images, greatly improving the defense performance of the target classifier. The enhanced recovered images generated by EITGAN also have good visual quality and generalization performance.

However, as the number of classes in the target classifier increases, the effectiveness of EITGAN in defending against adversarial samples gradually diminishes. The classification accuracy on a thousand-class task only improved by 2.3%. This performance was relatively less effective compared to the successful defense in a five-class scenario. In future work, we will explore the impact of the number of classes on defense effectiveness. We will also try to further improve the performance of EITGAN on more classes and further explore the application of positive perturbation to clean images.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62072250, U1636117, U1804263 and 61802212), the Zhongyuan Science and Technology Innovation Leading Talent Project of China (Grant No. 214200510019), and the Plan for Scientific Talent of Henan Province (Grant No. 2018JR0018).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. X. Li, J. Wu, Z. Sun, Z. Ma, J. Cao, J. Xue, Bsnet: Bi-similarity network for few-shot fine-grained image classification, *IEEE Trans. Image Process.*, **30** (2021), 1318–1331. https://doi.org/10.1109/TIP.2020.3043128
2. X. Chen, C. Xie, M. Tan, L. Zhang, C. J. Hsieh, B. Gong, Robust and accurate object detection via adversarial learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 16622–16631.

3. X. Li, H. He, X. Li, D. Li, G. Cheng, J. Shi, et al., Pointflow: Flowing semantics through points for aerial image segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 4217–4226.
4. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in *International Conference on Learning Representations*, ICLR, (2018), 1–23.
5. P. Mangla, S. Jandial, S. Varshney, V. N. Balasubramanian, Advgan++: Harnessing latent layers for adversary generation, *arXiv preprint*, (2019), arXiv:1908.00706. https://doi.org/10.48550/arXiv.1908.00706
6. X. Li, L. Chen, J. Zhang, J. Larus, D. Wu, Watermarking-based defense against adversarial attacks on deep neural networks, in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, (2021), 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534236
7. Y. Zhu, X. Wei, Y. Zhu, Efficient adversarial defense without adversarial training: A batch normalization approach, in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, (2021), 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533949
8. H. Kwon, Y. Kim, H. Yoon, D. Choi, Classification score approach for detecting adversarial example in deep neural network, *Multimedia Tools Appl.*, **80** (2021), 10339–10360. https://doi.org/10.1007/s11042-020-09167-z
9. B. Huang, Z. Ke, Y. Wang, W. Wang, L. Shen, F. Liu, Adversarial defence by diversified simultaneous training of deep ensembles, in *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI, (2021), 7823–7831. https://doi.org/10.1609/aaai.v35i9.16955
10. N. Das, M. Shanbhogue, S. T. Chen, F. Hohman, S. Li, L. Chen, et al., Shield: Fast, practical defense and vaccination for deep learning using jpeg compression, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Association for Computing Machinery, (2018), 196–204.
11. C. Xie, J. Wang, Z. Zhang, Z. Re, A. Yuille, Mitigating adversarial effects through randomization, in *International Conference on Learning Representations*, ICLR, (2018), 1–16.
12. C. Guo, M. Rana, M. Cisse, L. V. D. Maaten, Countering adversarial images using input transformations, in *International Conference on Learning Representations*, ICLR, (2018), 1–12.
13. A. Prakash, N. Moran, S. Garber, A. DiLillo, J. Storer, Deflecting adversarial attacks with pixel deflection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 8571–8580.
14. A. Mustafa, S. H. Khan, M. Hayat, J. Shen, L. Shao, Image super-resolution as a defense against adversarial attacks, *IEEE Trans. Image Process.*, **29** (2020), 1711–1724. https://doi.org/10.1109/TIP.2019.2940533
15. R. K. Meleppat, K. E. Ronning, S. J. Karlen, M. E. Burns, E. N. Pugh, R. J. Zawadzki, In vivo multimodal retinal imaging of disease-related pigmentary changes in retinal pigment epithelium, *Sci. Rep.*, **11** (2021), 16252. https://doi.org/10.1038/s41598-021-95320-z

16. R. K. Meleppat, C. R. Fortenbach, Y. Jian, E. S. Martinez, K. Wagner, B. S. Modjtahedi, et al., In Vivo Imaging of Retinal and Choroidal Morphology and Vascular Plexuses of Vertebrates Using Swept-Source Optical Coherence Tomography, *Transl. Vision Sci. Technol.*, **11** (2022), 11. https://doi.org/10.1167/tvst.11.8.11

17. K. M. Ratheesh, L. K. Seah, V. M. Murukeshan, Spectral phase-based automatic calibration scheme for swept source-based optical coherence tomography systems, *Phys. Med. Biol.*, **61** (2016), 7652.

18. I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, *arXiv preprint*, (2014), arXiv:1412.6572. https://doi.org/10.48550/arXiv.1412.6572

19. A. Kurakin, I. J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in *5th International Conference on Learning Representations*, ICLR, (2017), 1–14

20. S. M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: A simple and accurate method to fool deep neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 2574–2582.

21. N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in *2017 IEEE Symposium on Security and Privacy (sp)*, IEEE, (2017), 39–57. https://doi.org/10.1109/SP.2017.49

22. Y. Luo, X. Boix, G. Roig, T. A. Poggio, Q. Zhao, Foveation-based mechanisms alleviate adversarial examples, *arXiv preprint*, (2015), arXiv:1511.06292. https://doi.org/10.48550/arXiv.1511.06292

23. B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (2017), 136–144.

24. W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, et al., Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 1874–1883.

25. J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: A large-scale hierarchical image database, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2009), 248–255. https://doi.org/10.1109/CVPR.2009.5206848

26. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2016), 2818–2826. https://doi.org/10.1109/CVPR.2016.308

27. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90

28. C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI, (2017), 4278–4284. https://doi.org/10.1609/aaai.v31i1.11231

29. S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 1765–1773.

30. N. Q. K. Le, Q. T. Ho, V. N. Nguyen, J. S. Chang, BERT-Promoter: An improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection, *Comput. Biol. Chem.*, **99** (2022), 107732. https://doi.org/10.1016/j.compbiolchem.2022.107732

31. N. Q. K. Le, T. T. Nguyen, Y. Y. Ou, Identifying the molecular functions of electron transport proteins using radial basis function networks and biochemical properties, *J. Mol. Graphics Modell.*, **73** (2017), 166–178. https://doi.org/10.1016/j.jmgm.2017.01.003

32. S. Baluja, I. Fischer, Adversarial transformation networks: Learning to generate adversarial examples, *arXiv preprint*, (2017), arXiv:1703.09387. https://doi.org/10.48550/arXiv.1703.09387

33. C. Xiao, B. Li, J. Y. Zhu, W. He, M. Liu, D. Song, Generating adversarial examples with adversarial networks, *arXiv preprint*, (2018), arXiv:1801.02610. https://doi.org/10.48550/arXiv.1801.02610

34. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, (2016), 2921–2929. https://doi.org/10.1109/CVPR.2016.319

35. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, *Int. J. Comput. Vision*, **128** (2020), 336–359. https://doi.org/10.1007/s11263-019-01228-7

36. D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, D. Song, Natural adversarial examples, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 15262–15271.

37. S. Zagoruyko, N. Komodakis, Wide residual networks, *arXiv preprint*, (2016), arXiv:1605.07146. https://doi.org/10.48550/arXiv.1605.07146

AIMS Press