Received: 27 March 2021 | Revised: 3 September 2021 | Accepted: 16 September 2021

IET Image Processing

DOI: 10.1049/ipr2.12354

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Detecting adversarial examples by additional evidence from noise domain

Song Gao[1,2] | Shui Yu[3] | Liwen Wu[4] | Shaowen Yao[2,4] | Xiaowei Zhou[3]

[1] School of Information Science and Engineering, Yunnan University, Kunming, China

[2] Engineering Research Center of Cyberspace, Yunnan University, Kunming, China

[3] School of Computer Science, University of Technology Sydney, Sydney, Australia

[4] National Pilot School of Software, Yunnan University, Kunming, China

Correspondence
Shaowen Yao, National Pilot School of Software, and Engineering Research Center of Cyberspace, Yunnan University, Kunming 650504, China.
Email: yaosw@ynu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 61863036; Project of Yunnan Provincial Science and Technology Department, Grant/Award Number: 201901BB050076; Project of Provincial Industrial Internet, Grant/Award Number: TC200H01C

Abstract

Deep neural networks are widely adopted powerful tools for perceptual tasks. However, recent research indicated that they are easily fooled by adversarial examples, which are produced by adding imperceptible adversarial perturbations to clean examples. Here the steganalysis rich model (SRM) is utilized to generate noise feature maps, and they are combined with RGB images to discover the difference between adversarial examples and clean examples. In particular, a two-stream pseudo-siamese network that fuses the subtle difference in RGB images with the noise inconsistency in noise features is proposed. The proposed method has strong detection capability and transferability, and can be combined with any model without modifying its architecture or training procedure. The extensive empirical experiments show that, compared with the state-of-the-art detection methods, the proposed approach achieves excellent performance in distinguishing adversarial samples generated by popular attack methods on different real datasets. Moreover, this method has good generalization, it trained by a specific adversary can defend against other adversaries effectively.

1 | INTRODUCTION

Deep neural networks have achieved superior performance on many perceptual tasks, such as face recognition [1], object detection [2] and image classification [3]. However, there is obvious difference between the perception systems of humans and neural networks. Szegedy et al. [4] have demonstrated that adversarial examples generated by adding tiny but elaborately designed perturbations can easily fool neural networks with high confidence. Many different methods [5–13] have been proposed to design the worst-case perturbations. For images, the perturbations are imperceptible and do not stir any doubt about the correct classification for humans. Most strikingly, adversarial examples have transferability, i.e., an adversarial example generated by a model can remain attack effective for other models. This makes adversaries can successfully attack a model without knowing its details.

The undesirable property of deep neural networks has become major problem in safety-critical applications like medicine, finance and autonomous driving. Methods to increase the robustness of neural networks have been proposed from augmenting the training data [14–17] with adversarial examples to distilling robust networks from the original networks [18, 19]. Unfortunately, no matter how robust a model is, there are always new attacks that can successfully fool it. When a trained model is being applied, the cost is huge of retraining it to deal with new attacks. Therefore, convenient and flexible defence methods are essential.

Detection only methods are flexible, and can provide protection to a model even if the model is being used. For example, KD+BU [20] and ML-LOO [21] utilize the distribution character of different categories to detect adversarial examples. [22] grafts a detection subnetwork on the targeted model. Although these methods show compelling performance results on a

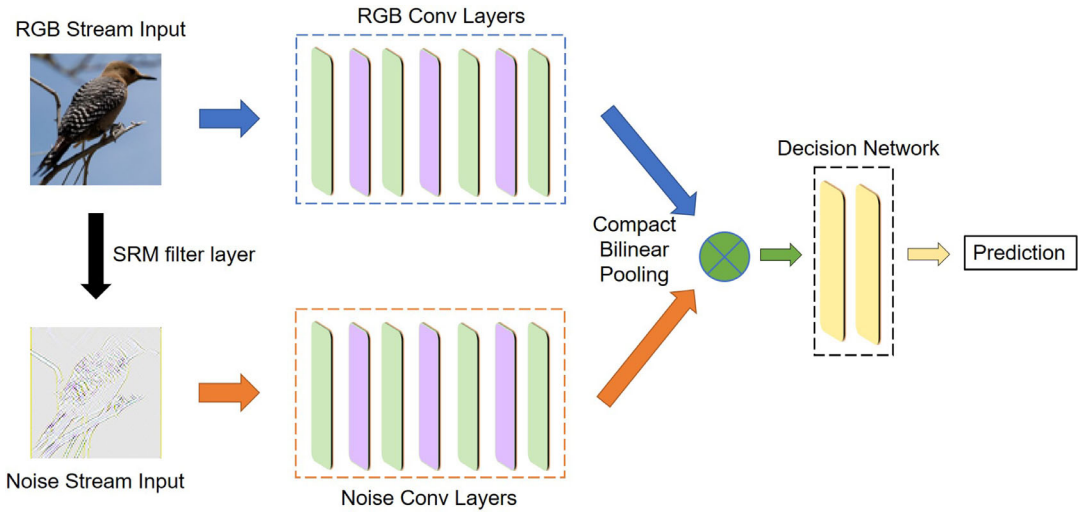

FIGURE 1 Illustration of our two-stream pseudo-Siamese network. Colour code used: light green = Conv+BN+ReLU, purple = max pooling, green = bilinear pooling, yellow = fully connected layers. The RGB stream uses original images as input, and the noise stream uses noise features produced by the SRM filter layer as input. Bilinear pooling combines the spatial co-occurrence features, and the decision network utilizes the combined features to discriminate adversarial examples

number of state-of-the-art adversarial attacks, one major drawback is that these methods depend closely on the protected model. These methods will fail when they are used to protect a model that we cannot obtain its details such as Amazon Machine Learning and BigML. If a detection method can detect adversarial examples effectively without relying on the targeted model, the problem is no longer a problem. However, adversarial perturbations are tiny, i.e. an adversarial image is very similar to its corresponding original image, hence it is extremely difficult to discriminate the adversarial of an image only from this image itself.

In this paper, we treat adversarial perturbations as a kind of noise, and extract noise features to provide additional evidence for adversarial example detection. Our model consists of a two-stream pseudo-Siamese network and a decision network (see Figure 1). The first stream discovers clues to the subtle difference like contrast difference, unnatural pixels from RGB features. The second stream is a noise stream, which utilizes noise features to capture the noise inconsistency. The intuition behind the second stream is that although adversarial perturbations are pretty special, they are still noise, the noise features between original images and adversarial images are unlikely to match. To utilize the noise features, we need to choose a suitable tool to convert RGB images into noise domain. We observe that the total variation of adversarial images is obviously larger than that of original images, which shows that the value difference between adjacent pixels is larger in adversarial images. We thus select steganalysis rich model (SRM) [23, 24] to generate noise features. SRM extracts local noise features from adjacent pixels, and amplifies local pixel difference in adversarial images. We then adopt bilinear pooling [25, 26] to combine the features produced from the two streams. Bilinear pooling is often used for fine-grained classification, it can fuse two streams while preserving spatial information. Finally, the fused features are passed into a decision network to distinguish whether an input image is adversarial or not.

We summarize our contributions as follows:

1. We propose a novel two-stream adversarial example detection framework. The proposed method can obtain rich feature information from noise features to provide additional evidence for adversarial example detection. With the rich feature information, our method gets rid of the dependence on the targeted model and achieves good transferability, it can be reused to protect different models after once training.
2. We select the steganalysis rich model (SRM) to produce noise feature maps. We notice that the total variation of adversarial images is significantly larger than that of clean images. This means the difference in the value of neighbouring pixels is larger in adversarial images. SRM amplifies the difference in noise domain, and obtains additional plentiful information to assist in detecting adversarial samples.
3. Extensive experiments show that our method achieves excellent performance in defending against both white-box attacks and black-box attacks. Moreover, our method has good generalization, it trained by an attack can defend against other attacks effectively.

The remainder of this paper is organized as follows: Section 2 briefly reviews steganalysis rich model, adversarial attacks and defences. Section 3 presents our proposed method in detail. In Section 4, we describe the experimental setting, experimental results and correlation analysis. Finally, Section 5 shows the conclusion.

2 | RELATED WORK

2.1 | Steganalysis rich model

Steganalysis rich model is mainly used in image forensics tasks. It extracts local noise features from adjacent pixels to capture

TABLE 1 The average total variation of images with different attacks in different datasets

Dataset	Clean	FGSM	BIM	PGD	MIM	SPSA
MNIST	101	217	201	213	214	217
CIFAR-10	342	483	375	424	445	454
CIFAR-100	333	463	372	414	431	454
ImageNet	13,390	16,189	14,465	15,394	15,747	17,307

$$\frac{1}{4}\begin{bmatrix}0&0&0&0&0\\0&-1&2&-1&0\\0&2&-4&2&0\\0&-1&2&-1&0\\0&0&0&0&0\end{bmatrix}\quad\frac{1}{12}\begin{bmatrix}-1&2&-2&2&-1\\2&-6&8&-6&2\\-2&8&-12&8&-2\\2&-6&8&-6&2\\-1&2&-2&2&-1\end{bmatrix}\quad\frac{1}{2}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&-2&1&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$$

FIGURE 2 The SRM filter kernels used in our work. In grayscale images, we only use the left kernel to generate SRM images

the inconsistency between authentic and tampered regions. Ref. [27] demonstrated the performance of SRM in distinguishing tampered regions from authentic regions, and combined SRM features with Convolutional Neural Networks to perform manipulation localization. Ref. [30] used an SRM filter kernel as the initialization of a Convolutional Neural Network to improve detection accuracy. Ref. [24] utilized SRM filter kernels to extract low-level noise used as input to a Faster R-CNN network, and captured tampering traces in the noise features. Ref. [42] adopted SRM to extract the steganalysis features of images, and enhanced the features by the estimated probability of modifications in images to detect adversarial examples.

2.2 | Adversarial attacks

Suppose $x \subset R^m$ is a clean image and y is x's label. For a trained deep neural network f with parameters θ, $f(x, \theta) = y$. What adversarial attacks do is:

$$\min ||r||_p \text{ subject to } f(x + r, \theta) \neq y, \tag{1}$$

where r is the perturbation, and $x + r$ is the adversarial example x_{adv}. $||\cdot||_p$ denotes L_p norm, including L_0, L_2 and L_∞ norm. L_0 norm counts the number of the changed pixels in x_{adv}, L_2 norm measures the Euclidean distance between x_{adv} and x, L_∞ norm denotes the maximum change of all pixels in x_{adv}.

FGSM [5] is an earlier adversarial attack. It performs a single-step gradient update along the direction of the sign of gradient at each pixel. The formula for FGSM to generate adversarial examples is as follows:

$$x_{adv} = x + \varepsilon \cdot sign\left(\nabla_x L\left(f(x, \theta), y\right)\right), \tag{2}$$

where ε is a coefficient, that control the boundary of perturbations. sign is the step function, and $L(\cdot)$ is the cost function. ∇_x denotes the gradient of the model with respect to a clean input x, and y denotes the correct label of x.

Equation (2) generates adversarial examples by increasing the cost between $f(x, \theta)$ and y. This attack is called non-targeted attack, which just let the classifier go wrong. Let us change Equation (2) to Equation (3), then non-targeted attack becomes targeted attack:

$$x_{adv} = x - \varepsilon \cdot sign\left(\nabla_x L\left(f(x, \theta), y'\right)\right), \tag{3}$$

where y' denotes a specific target class. Targeted attack is more difficult than non-targeted attack, so the attack success rate of targeted attack is lower than non-targeted attack. Since the focus of this study is defence, non-targeted attacks with higher attack success rate are used to generate adversarial examples.

BIM [6] is a straightforward extension of FGSM. It replaces the single-step with multiple small steps, extending FGSM into an iterative algorithm. Projected gradient descent (PGD) [7] is similar as BIM, the only difference between them is that PGD first randomly perturbs the original image before starting the iteration. PGD is regarded as the strongest first-order attack, which can even reduce the accuracy of the classifier to 0 in the white-box attack. JSMA [8] uses L_0 norm to restrict perturbations. For images, this means that JSMA modifies only a few pixels in images to fool the targeted classifier. The extreme case of JSMA is the one pixel attack [12], which only changes one pixel in an image to fool a classifier. DeepFool [9] aims to find the shortest distance from the clean image to the decision boundary of the adversarial image. Momentum iterative method (MIM) [10] integrates the momentum term into the iterative process for attacks to generate more transferable adversarial examples. CW_2 [13] solves the task of generating adversarial examples as an optimization problem, and converts adversarial examples into the argtanh space, making it more flexible to use optimization solvers.

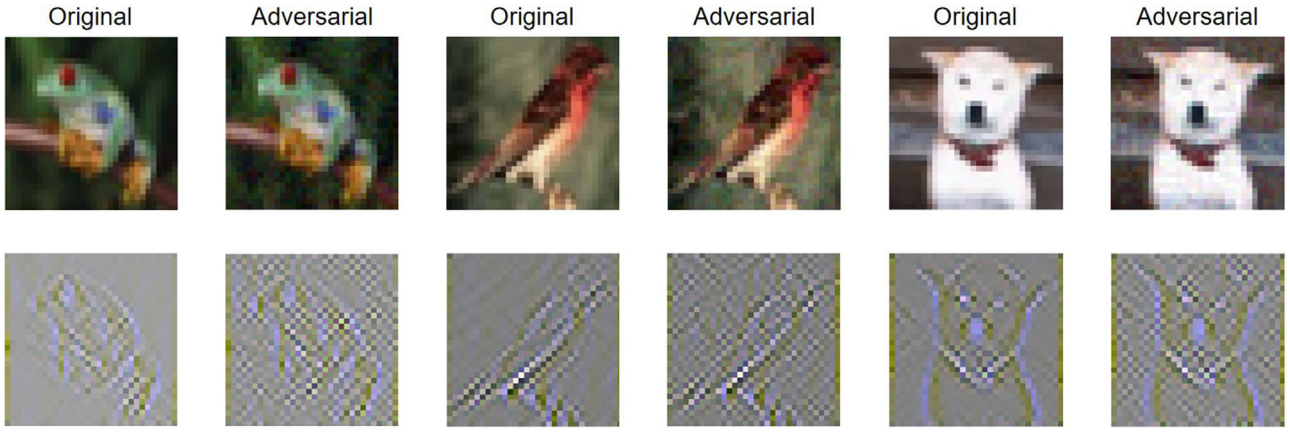

FIGURE 3 Examples of original, adversarial images and corresponding SRM images in CIFAR-10. The adversarial images are produced using PGD with maximum perturbation $\varepsilon = 8/255$. Each column shows an RGB image and its corresponding SRM image. A clean image usually consists of different smooth regions and complex texture regions. The adversarial perturbations will destroy the smooth regions and make the complex texture area more cluttered. The main focus of RGB channel is on semantic image content, thus ignoring this difference

TABLE 2 Classifier architectures for MNIST and CIFAR-10. These three classifiers are designed by the authors, so it is specifically explained here in the form of a table

MNIST Local model	MNIST Black model	CIFAR-10 Local model
Conv(32,3,1), ReLu	Conv(32,3,1), ReLU	Conv(64,3,1), ReLU
Max Pooling 2×2	Conv(32,3,1), ReLU	Conv(64,3,1), ReLU
Conv(64,3,1), ReLU	Max Pooling 2×2	Max Pooling 2×2
Max Pooling 2×2	Conv(64,3,1), ReLU	Conv(128,3,1), ReLU
Full Connected 200	Conv(64,3,1), ReLU	Conv(128,3,1), ReLU
Softmax 10	Max Pooling 2×2	Max Pooling 2×2
	Full Connected 200	Full Connected 256
	Full Connected 200	Full Connected 256
	Softmax 10	Softmax 10

Conv(d, k, s) denotes the convolutional layer with d as dimension, k as kernel size and s as stride.

TABLE 3 Classifiers for CIFAR-10, CIFAR-100 and ImageNet

	CIFAR-10	CIFAR-100	ImageNet
Local	*	ResNet101V2	VGG16
Black	VGG16	ResNet152	MobileNet
	MobileNet	DenseNet169	InceptionNetV3
	ResNet50	DenseNet201	DenseNet121

2.3 | Adversarial defences

Due to the huge potential hazard of adversarial samples, a number of defence methods have been proposed. Adversarial training is a standard brute force method that improves the robustness of a model by injecting adversarial examples into the training set [14, 15, 17]. Although adversarial training can enhance the robustness of a network, it is a non-adaptive method that requires the information like architecture, hyperparameters of the network. And we can still generate effective adversarial examples by the adversarial trained network. Gradient regularization [31] usually adds regularization terms to the cost functions. For instance, ref. [32] adds the input gradient regularization to the cost function to ensure that if any input changes slightly, the difference between the predictions and the labels will not change significantly. Obviously, these methods need to be combined with adversarial training. Defensive distillation [18, 19] transfers knowledge from a complex network to a smaller one to defend against adversarial examples. Distillation can provide a smoother cost function for the second network, and makes the network have high classification accuracy for adversarial examples. These defence methods often involve modifications in the model training process, which usually require higher computational or example complexity.

Input reconstruction [33, 34, 35] reconstructs inputs to remove adversarial perturbations before they are passed into the targeted model. Data compression [37, 38, 39, 40, 41] compresses or transforms inputs to remove the adversarial of inputs. These methods do not need to change the targeted model, and can protect the targeted model at testing phase. However, changing inputs often lead to loss of accuracy.

Complimentary to the previous defence methods, an alternative line of works focus on screening out adversarial samples. Ref. [22] grafted a discriminator on the targeted model, and used the output of the intermediate layers as input to the discriminator. Refs. [20] and [21] utilized the distribution character in hidden-layer output of different categories to identify adversarial examples. Although these defences do not modify the targeted model and input samples, they rely closely on the targeted model. These methods will fail when we cannot achieve the knowledge of the targeted model. Moreover, these methods are model-specific. The improvement of robustness of a model cannot be generalized to other models. Our work focuses on presenting an effective adversarial example detection method which can get rid of the dependence of the targeted model. This means our method is flexible, it can protect a model even if we

TABLE 4 The architecture of TSD for MNIST

For MNIST	
RGB stream	**Noise stream**
Conv(32, 3, 1), BN, ReLU	Conv(32, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(64, 3, 1), BN, ReLU	Conv(64, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(128, 3, 1), BN, ReLU	Conv(128, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Compact bilinear pooling	
Full connected 256	
Full connected 256	
Softmax 2	

BN presents batch normalization.

TABLE 5 The architecture of TSD for CIFAR-10 and CIFAR-100

For CIFAR-10 and CIFAR-100	
RGB stream	**Noise stream**
Conv(32, 3, 1), BN, ReLU	Conv(32, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(64, 3, 1), BN, ReLU	Conv(64, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(128, 3, 1), BN, ReLU	Conv(128, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(256, 3, 1), BN, ReLU	Conv(256, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Compact Bilinear Pooling	
Full Connected 512	
Full Connected 512	
Softmax 2	

BN presents batch normalization.

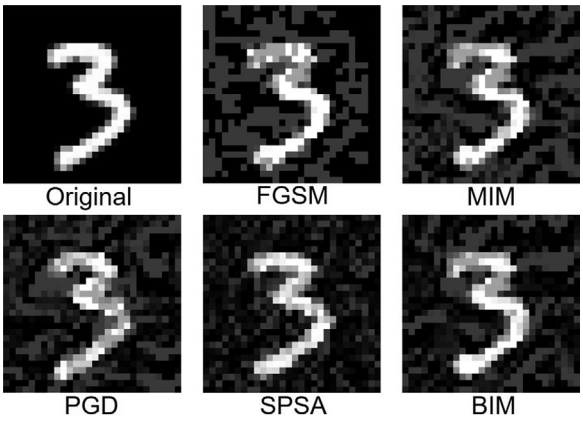

FIGURE 4 Examples of clean and adversarial images in MNIST dataset. The adversarial images are produced with maximum perturbation $\varepsilon = 0.3$ (out of 1.0)

TABLE 6 The architecture of TSD for ImageNet

For ImageNet	
RGB stream	**Noise stream**
Conv(32, 3, 1), BN, ReLU	Conv(32, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(64, 3, 1), BN, ReLU	Conv(64, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(128, 3, 1), BN, ReLU	Conv(128, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(256, 3, 1), BN, ReLU	Conv(256, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(512, 3, 1), BN, ReLU	Conv(512, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Conv(512, 3, 1), BN, ReLU	Conv(512, 3, 1), BN, ReLU
Max Pooling 2×2	Max Pooling 2×2
Compact Bilinear Pooling	
Full Connected 512	
Full Connected 512	
Softmax 2	

BN presents batch normalization.

TABLE 7 Performance of G-RGB with the assistance of different attacks in defending against adversarial examples on ImageNet. Adversarial examples are produced by local model

Assistant attack	Evaluation metric	FGSM	MIM	PGD	SPSA	BIM
PGD	TPR	**0.396**	0.562	**0.913**	**0.189**	0.877
	AUC	0.711	0.967	**0.984**	**0.628**	0.951
MIM	TPR	0.388	**0.942**	0.900	0.020	0.894
	AUC	**0.851**	**0.978**	0.949	0.608	0.979
BIM	TPR	0.006	0.342	0.898	0.047	**0.976**
	AUC	0.613	0.867	0.951	0.521	**0.998**

do not know the information of the model. Meanwhile, flexibility gives our method good transferability, our method can protect different models after once training.

3 | PROPOSED METHOD

We adopt a two-stream pseudo-Siamese architecture to detect adversarial images. As shown in Figure 1, the RGB stream uses RGB images as input and the noise stream uses SRM images produced by the SRM filters as input. We then utilize compact bilinear pooling to fuse the features before a decision network.

TABLE 8 Performance of GTS with the assistance of different attacks in defending against adversarial examples on ImageNet. Adversarial examples are produced by local model

Assistant attack	Evaluation metric	FGSM	MIM	PGD	SPSA	BIM
PGD	TPR	0.648	0.980	**0.992**	**0.244**	0.996
	AUC	0.963	0.981	**0.998**	**0.704**	0.972
MIM	TPR	**0.655**	**0.981**	0.964	0.044	0.934
	AUC	**0.972**	**0.989**	0.977	0.703	0.982
BIM	TPR	0.020	0.550	0.983	0.002	**0.998**
	AUC	0.583	0.962	0.990	0.522	**0.999**

TABLE 9 Performance of RGB with the assistance of different attacks in defending against adversarial examples on ImageNet. Adversarial examples are produced by local model

Assistant attack	Evaluation metric	FGSM	MIM	PGD	SPSA	BIM
PGD	TPR	0.931	0.916	**0.924**	**0.896**	**0.532**
	AUC	0.984	0.972	**0.979**	**0.949**	**0.783**
MIM	TPR	**0.972**	**0.936**	0.514	0.710	0.218
	AUC	**0.995**	**0.991**	0.949	0.912	0.763
BIM	TPR	0.400	0.392	0.370	0.384	0.354
	AUC	0.556	0.551	0.535	0.544	0.527

TABLE 10 Performance of Noise with the assistance of different attacks in defending against adversarial examples on ImageNet. Adversarial examples are produced by local model

Assistant attack	Evaluation metric	FGSM	MIM	PGD	SPSA	BIM
PGD	TPR	**0.922**	**0.925**	**0.998**	**0.903**	**0.912**
	AUC	**0.957**	**0.960**	**0.964**	**0.934**	**0.963**
MIM	TPR	0.850	0.836	0.414	0.511	0.642
	AUC	0.931	0.946	0.549	0.660	0.751
BIM	TPR	0.112	0.134	0.167	0.114	0.156
	AUC	0.502	0.511	0.507	0.543	0.528

TABLE 11 Performance of TSD with the assistance of different attacks in defending against adversarial examples on ImageNet. Adversarial examples are produced by local model

Assistant attack	Evaluation metric	FGSM	MIM	PGD	SPSA	BIM
PGD	TPR	**0.998**	**1.0**	**0.998**	**0.994**	**0.944**
	AUC	0.990	0.991	**0.991**	**0.967**	**0.969**
MIM	TPR	0.982	0.964	0.732	0.722	0.794
	AUC	**0.991**	**0.995**	0.926	0.874	0.910
BIM	TPR	0.695	0.688	0.697	0.642	0.699
	AUC	0.842	0.846	0.851	0.864	0.881

3.1 | SRM filters

When generating adversarial images, L_p norms, including L_0 norm, L_2 norm and L_∞ norm, are usually used to restrict the change of pixel values. Hence an adversarial image and its corresponding original image are very similar. It is extremely difficult to detect the adversarial of an image only from this image itself. We have to look for other features to aid adversarial example detection. Table 1 shows the total variation of clean images and adversarial images produced by different attacks in different datasets. We can clearly see that the total variation of adversarial images is significantly larger than that of original images. This is easy to understand, due to the addition of adversarial perturbations, the value difference of adjacent pixels in an image becomes larger. We use SRM filter kernels to extract local noise features from RGB images to amplify the difference.

Steganalysis rich model generates the noise features of one image through the residual between a pixel value and the estimation of the pixel value generated by only interpolating the adjacent pixel value. SRM uses 30 basic filters to gather the basic noise features by performing non-linear operations like maximum and minimum of the nearby output after filtering. By quantifying and truncating the output of these filters, SRM extracts the nearby co-occurrence information as final features. Ref. [24] has demonstrated that using only 3 kernels can achieve similar performance as using 30 kernels. We choose the same three kernels that have the best performance as [24], and their weights are shown in Figure 2. We copy each kernel twice to form three $5 \times 5 \times 3$ convolution kernels as the parameters of SRM filter layer with 3-channel input and 3-channel output. For grayscale images, we only use the left kernel as the parameters of SRM filter layer with 1-channel input and 1-channel output.

Figure 3 shows some noise feature maps after the SRM layer. We can clearly see that there is significant difference between original images and adversarial images in the SRM images although they are similar in the RGB images. Especially in smooth regions, SRM amplifies the insignificant difference between neighbouring pixels.

3.2 | Two-stream network

Siamese networks [28, 29, 36] are often used to measure the similarity between different inputs. We adopt the similar architecture of Siamese network, i.e. two-stream structure, to extract features from RGB domain and noise domain. Although RGB stream and noise stream have the same network structure, they do not share parameters. RGB stream discovers subtle difference between clean images and adversarial images like contrast difference or unnatural pixels from RGB features. Noise stream utilizes noise features to provide additional evidence for adversarial image detection. The features extracted by the two streams are fused by bilinear pooling [25]. Compared with sum or average, bilinear pooling uses second-order statistic information to fuse features from different channels to achieve fine-grained classification. However, the features combined by bilinear pool-

TABLE 12 Performance of detection methods in defending against adversarial examples on MNIST

		FGSM		MIM		PGD		SPSA		BIM	
Model	**Method**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**
Local	G-RGB	0.929	0.973	0.953	0.992	0.973	0.997	0.722	0.921	0.922	0.989
	GTS	0.945	0.988	0.981	0.991	0.983	0.993	0.746	0.952	0.932	0.990
	KD+BU	0.789	0.771	0.846	0.795	0.865	0.809	0.769	0.749	0.866	0.810
	RGB	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	Noise	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.999	**1.0**
	TSD	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Black	G-RGB	0.916	0.984	0.938	0.958	0.916	0.954	0.615	0.899	0.906	0.965
	GTS	0.924	0.990	0.943	0.966	0.927	0.937	0.626	0.903	0.831	0.934
	KD+BU	0.782	0.785	0.783	0.765	0.785	0.772	0.720	0.645	0.780	0.752
	RGB	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.999	**1.0**
	Noise	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.975	**1.0**
	TSD	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

TABLE 13 Performance of detection methods in defending against adversarial examples on CIFAR-10

		FGSM		MIM		PGD		SPSA		BIM	
Model	**Method**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**
Local	G-RGB	0.435	0.949	0.909	0.996	0.898	0.994	0.213	0.828	**0.924**	**0.994**
	GTS	0.727	0.975	0.954	0.996	0.925	0.993	0.681	0.940	0.911	0.986
	KD+BU	0.849	0.599	0.847	0.721	0.870	0.761	0.368	0.521	0.893	0.798
	RGB	0.997	0.999	**0.997**	0.999	0.994	0.999	0.976	0.993	0.753	0.954
	Noise	0.992	0.999	0.989	0.998	0.956	0.996	0.957	0.971	0.859	0.970
	TSD	**0.999**	**1.0**	0.996	**1.0**	**0.998**	**1.0**	**0.993**	**1.0**	0.894	0.983
VGG16	G-RGB	0.164	0.839	0.129	0.830	0.133	0.722	0.159	0.797	0.126	0.677
	GTS	0.579	0.919	0.569	0.917	0.504	0.882	0.516	0.933	0.389	0.769
	KD+BU	0.350	0.493	0.665	0.513	0.690	0.524	0.341	0.500	0.689	0.524
	RGB	0.951	0.972	**0.941**	0.972	0.990	0.999	0.970	0.993	0.540	0.786
	Noise	0.941	0.972	0.936	0.971	0.916	0.993	0.953	0.970	0.696	0.853
	TSD	**0.954**	**0.977**	0.933	**0.977**	**0.998**	**1.0**	**0.999**	**1.0**	**0.752**	**0.863**
MobileNet	G-RGB	0.101	0.826	0.173	0.809	0.117	0.708	0.157	0.797	0.116	0.684
	GTS	0.528	0.935	0.527	0.935	0.512	0.898	0.514	0.931	0.350	0.849
	KD+BU	0.345	0.486	0.331	0.494	0.702	0.522	0.683	0.528	0.706	0.537
	RGB	0.986	0.998	0.991	0.999	0.992	0.999	0.963	0.992	0.793	0.960
	Noise	0.843	0.987	0.865	0.989	0.873	0.989	0.923	0.965	0.642	0.960
	TSD	**0.999**	**1.0**	**0.992**	**1.0**	**0.999**	**1.0**	**0.999**	**1.0**	**0.822**	**0.973**
ResNet50	G-RGB	0.177	0.821	0.147	0.797	0.118	0.726	0.144	0.789	0.117	0.716
	GTS	0.550	0.923	0.585	0.925	0.524	0.904	0.504	0.932	0.382	0.884
	KD+BU	0.367	0.504	0.344	0.497	0.684	0.520	0.325	0.491	0.689	0.531
	RGB	0.976	0.993	0.979	0.994	0.989	0.999	0.973	0.993	0.796	0.976
	Noise	0.809	0.967	0.926	0.975	0.981	0.982	0.936	0.966	0.701	0.968
	TSD	**0.990**	**0.996**	**0.976**	**0.996**	**0.993**	**1.0**	**0.995**	**1.0**	**0.837**	**0.983**

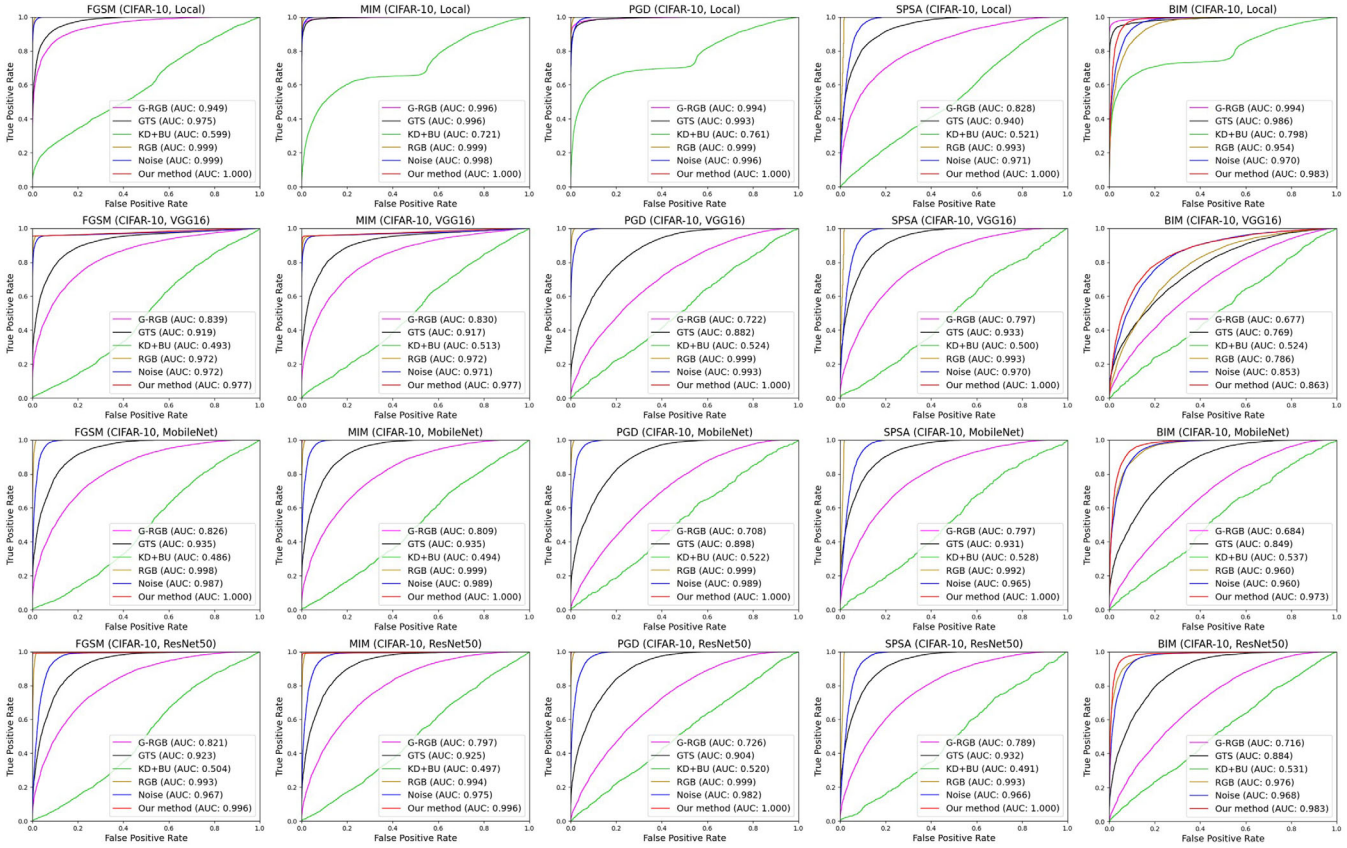

FIGURE 5 ROC curves of detection methods on CIFAR-10. The first row shows ROC curves of detection methods in distinguishing adversarial images produced by the local classifier. And the second to forth row show the six methods' ROC curves in distinguishing adversarial images generated by black classifiers. Except RGB, Noise and our method, the transferability of the other three methods is obviously not good. But GTS is noticeably better than G-RGB and KD+BU as it considers noise features

ing are high dimensional, typically on the order of hundreds of thousands to millions. To speed up training, we use compact bilinear pooling [26] to fuse the two streams. The compact bilinear pooling has the same performance as the full bilinear representation but with only thousands dimensions.

After the fused features pass through a decision network consisting of two fully connected layers and a softmax layer, the final predicted result is obtained:

$$y' = f_D\left(CBP\left(f_{RGB}(x), f_N\left(f_{SRM}(x)\right)\right)\right), \tag{4}$$

where x is an input image and y' is x's output. f_{SRM} denotes the SRM network with fixed weights. f_{RGB} and f_N are the RGB stream network and the noise stream network. CBP denotes the compact bilinear pooling. f_D is the decision network. We then use cross entropy loss and squared L_2 norm regularization that leads to the following objective function:

$$\min L = \lambda||\omega||_2 + L_{cross}\left(y', y\right), \tag{5}$$

where λ is a hyperparameter for balancing the regularization and loss. $||\cdot||_2$ denotes L_2 norm. ω are the weights of networks except for the SRM network. L_{cross} denotes cross entropy loss.

4 | EXPERIMENTS

In this section, we present the experimental setting and evaluation of our approach, and compare it with several state-of-the-art detection methods.

4.1 | Experimental setting

Datasets: We extensively evaluate our method on MNIST, CIFAR-10, CIFAR-100 and ImageNet. MNIST is a grayscale image dataset with image shape 28×28 from 10 categories, including 60,000 training images and 10,000 testing images. CIFAR-10 consists of colour images with image shape 32×32×3 from 10 categories, including 50,000 training images and 10,000 testing images. CIFAR-100 is composed of colour images with image shape 32×32×3 from 100 categories, including 50,000 training images and 10,000 testing images. For ImageNet, we choose 10 categories, i.e. ostrich, goldfish, axolotl, chameleon, violin, admiral, hummingbird, rapeseed, teapot and ice cream, from ILSVRC2012, each category contains 1300 training images and 50 test images with shape $224 \times 224 \times 3$.

TABLE 14 Performance of detection methods in defending against adversarial examples on CIFAR-100

		FGSM		MIM		PGD		SPSA		BIM	
Model	**Method**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**
ResNet101V2	G-RGB	0.506	0.864	0.875	0.968	0.925	0.981	0.413	0.624	0.925	**0.978**
	GTS	0.635	0.931	0.923	0.982	0.912	0.978	0.615	0.850	0.836	0.944
	KD+BU	0559	0.597	0.887	0.862	0.958	0.958	0.516	0.569	**0.965**	0.965
	RGB	0.992	0.948	**0.995**	0.969	0.974	0.955	0.861	0.918	0.642	0.876
	Noise	0.994	0.975	0.993	0.986	**0.986**	0.981	0.929	0.930	0.891	0.916
	TSD	**0.996**	**0.988**	**0.995**	**0.993**	**0.986**	**0.992**	**0.987**	**0.967**	0.866	0.892
ResNet152	G-RGB	0.257	0.758	0.190	0.712	0.112	0.614	0.091	0.588	0.094	0.591
	GTS	0.536	0.918	0.511	0.894	0.309	0.796	0.401	0.846	0.175	0.678
	KD+BU	0.535	0.590	0.545	0.613	0.530	0.593	0.516	0.567	0.529	0.594
	RGB	0.995	0.945	0.996	0.971	0.964	0.951	0.858	0.917	0.579	0.859
	Noise	0.992	0.959	0.995	0.976	0.990	0.971	0.925	0.928	**0.817**	**0.934**
	TSD	**0.998**	**0.979**	**0.998**	**0.989**	**1.0**	**0.988**	**0.997**	**0.965**	0.799	0.908
DenseNet169	G-RGB	0.211	0.728	0.153	0.682	0.092	0.585	0.087	0.578	0.078	0.564
	GTS	0.486	0.900	0.461	0.880	0.309	0.795	0.377	0.835	0.282	0.683
	KD+BU	0.525	0.582	0.532	0.595	0.518	0.566	0.511	0.567	0.510	0.559
	RGB	0.930	0.944	0.977	0.965	0.963	0.951	0.864	0.919	0.595	0.875
	Noise	0.943	0.922	0.969	0.948	0.979	0.962	0.925	0.929	**0.873**	**0.951**
	TSD	**0.958**	**0.961**	**0.989**	**0.975**	**0.991**	**0.982**	**0.993**	**0.966**	0.840	0.928
ResNet201	G-RGB	0.201	0.723	0.163	0.687	0.093	0.596	0.090	0.582	0.086	0.576
	GTS	0.481	0.894	0.466	0.883	0.320	0.805	0.378	0.835	0.285	0.693
	KD+BU	0.532	0.582	0.528	0.582	0.526	0.576	0.504	0.555	0.526	0.579
	RGB	0.962	0.944	0.987	0.963	0.967	0.952	0.863	0.919	0.644	0.895
	Noise	0.951	0.928	0.980	0.955	0.979	0.964	0.925	0.929	0.833	**0.960**
	TSD	**0.988**	**0.959**	**0.994**	**0.975**	**0.996**	**0.981**	**0.947**	**0.966**	**0.842**	0.946

Classifiers: All classifiers used in our study are shown in Tables 2 and 3. We train a local classifier and a black classifier for MNIST, and a local classifier and three black classifiers for CIFAR-10, CIFAR-100 and ImageNet, respectively. All classifiers are used to generate adversarial images with different attack methods for testing, but only the local model is used to assist detection methods. All classifiers are trained by Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with the batch size of 128, learning rate of 0.001, and epochs of 50.

Baseline methods: We compare our method named two-stream detector (TSD) with state-of-the-art detection methods including graft network (G-RGB) (the authors did not give their method a proper name, for convenience we named it graft network) [22], two-stream graft network (GTS), KD+BU [20], RGB-stream network (RGB) and Noise-stream network (Noise). The architectures of TSD for different datasets are shown in Tables 4, 5 and 6. Removing the noise stream in each architecture is the structure for RGB, and removing the RGB stream in each architecture is the structure for Noise.

Attack methods: We consider five attack methods, FGSM [5], BIM [6], PGD [7], MIM [10] and SPSA [11], to evaluate the discrimination power of different detection methods. To test the generalization of different methods, we just use one attack to assist detectors' training, and use all attacks to test detectors. We set $\varepsilon = 0.3$ in Equation (2) for MINST dataset, and $\varepsilon = 8/255$ for CIFAR-10, CIFAR-100 and ImageNet. To test the defence capability of different methods under different perturbation intensity, we also set $\varepsilon = 2/255$, 4/255, 6/255 and 10/255 for ImageNet.

Evaluation metrics: The true positive rate (TPR) and Area Under the receiver operating characteristic Curve (AUC) are used as the evaluation metrics for performance comparison. TPR is the proportion of adversarial images classified as adversarial.

Parameter setting: The five detectors (G-RGB, GTS, RGB, Noise and our method) are trained by Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$) with the batch size of 128, learning rate of 0.0001, and epochs of 50. We set the value of $\lambda = 0.0005$ in Equation (5). Each batch consists of 64 clean images and their corresponding adversarial images.

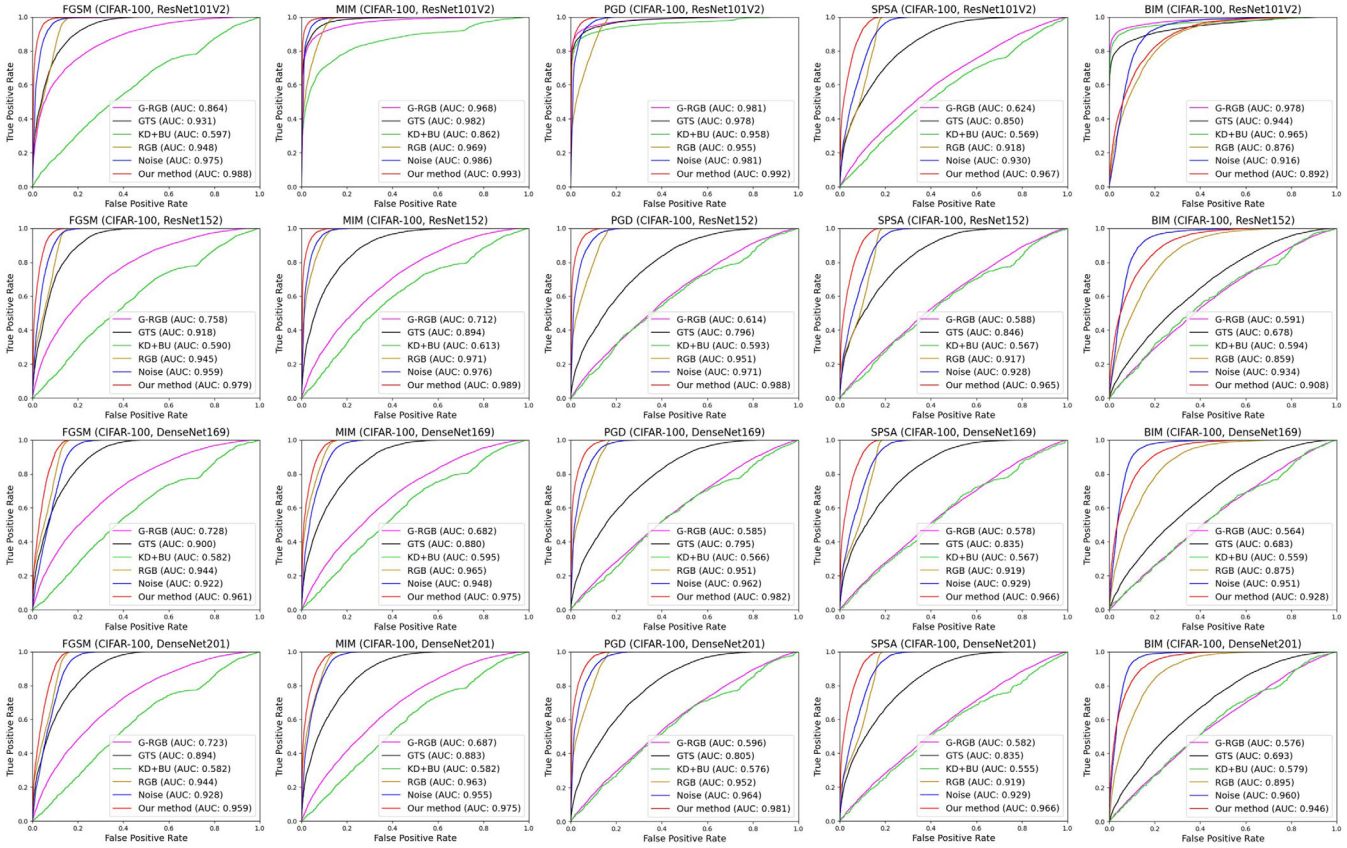

FIGURE 6 ROC curves of detection methods on CIFAR-100. The first row shows ROC curves of detection methods in distinguishing adversarial images produced by ResNet101V2, which play the role of the local model for CIFAR-100 dataset. And the second to forth row show the six methods' ROC curves in distinguishing adversarial images generated by black classifiers

4.2 | Generalization ability evaluation

A good detector should have good generalization, i.e. a detector trained by a specific adversary can generalize to other adversaries. However, the performance of a detector trained by different adversaries is different. Table 7 shows the performance of G-RGB with the assistance of different attacks in defending against adversarial examples on ImageNet. We can see that G-RGB trained by MIM has the best performance in defending against FGSM and MIM, and G-RGB trained by PGD has the best performance in defending against PGD and SPSA. On balance, G-RGB with PGD-assisted training has the best generalization ability, slightly better than MIM. Table 8 shows the detection effects of GTS trained by different attacks. Obviously, GTS is similar with G-RGB.

Table 9 displays the generalization ability of RGB trained by different attacks. RGB with MIM-assisted training has the best performance in defending against FGSM and MIM, and RGB with PGD-assisted training has the best performance in defending against PGD, SPSA and BIM. However, RGB trained by PGD can also discriminate the adversarial samples generated by FGSM and MIM effectively, but RGB trained by MIM cannot effectively discriminate the adversarial samples generated by PGD and SPSA. Noise (see Table 10) with PGD-assisted training has the best detection ability. Our method (TSD) (see Table 11) is similar with Noise, it trained by PGD has the best generalization ability. Overall, these five methods with PGD-assisted training have the best generalization capability. Therefore, in the following experiments, the five methods are trained with the assistance of PGD.

4.3 | Detection capability evaluation

4.3.1 | Verification on MNIST

Table 12 shows the TPR and AUC scores of different detection methods on MNIST dataset. We can see that the performance of KD+BU is inferior to other methods, GTS performs better than G-RGB, and RGB, Noise and our method (TSD) have excellent effects in defending against both white-box attacks and black-box attacks. Due to the images in MNIST are simple (the images in MNIST are grayscale images of handwritten numeral), ε is usually set to 0.3 to produce adversarial images. There is some distinct distinction between MNIST's images and their corresponding adversarial images. Figure 4 shows an image in MNIST and its corresponding adversarial images produced by FGSM, MIM, PGD, SPSA and BIM, respectively. We

TABLE 15 Performance of detection methods in defending against adversarial examples on ImageNet

		FGSM		MIM		PGD		SPSA		BIM	
Model	**Method**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**
VGG16	G-RGB	0.396	0.711	0.562	0.967	0.913	0.984	0.189	0.628	0.877	0.951
	GTS	0.648	0.963	0.980	0.981	0.992	**0.998**	0.244	0.704	**0.996**	**0.972**
	KD+BU	0.486	0.534	0.509	0.672	0.512	0.748	0.456	0.509	0.549	0.773
	RGB	0.931	0.984	0.916	0.972	0.924	0.979	0.896	0.949	0.532	0.783
	Noise	0.922	0.957	0.925	0.960	0.997	0.964	0.903	0.934	0.912	0.963
	TSD	**0.998**	**0.990**	**1.0**	**0.991**	**0.998**	0.991	**0.994**	**0.967**	0.944	0.969
MobileNet	G-RGB	0.152	0.762	0.148	0.769	0.118	0.591	0.118	0.621	0.114	0.538
	GTS	0.290	0.834	0.280	0.836	0.234	0.651	0.248	0.703	0.228	0.556
	KD+BU	0.567	0.523	0.537	0.500	0.451	0.517	0.567	0.510	0.460	0.521
	RGB	0.835	**0.910**	0.854	0.912	0.877	0.922	0.891	0.917	0.357	0.632
	Noise	0.872	0.878	0.884	0.883	0.932	0.947	0.944	0.935	**0.792**	0.750
	TSD	**0.912**	**0.910**	**0.897**	**0.913**	**0.984**	**0.979**	**0.990**	**0.966**	0.790	**0.824**
InceptionNetV3	G-RGB	0.134	0.654	0.140	0.661	0.116	0.581	0.118	0.620	0.108	0.514
	GTS	0.200	0.689	0.210	0.695	0.232	0.635	0.246	0.702	0.224	0.519
	KD+BU	0.574	0.517	0.553	0.510	0.574	0.502	0.560	0.498	0.460	0.520
	RGB	0.534	0.726	0.548	0.733	0.935	0.972	0.902	0.924	0.202	0.555
	Noise	0.664	0.707	0.670	0.711	0.972	0.943	**0.988**	0.933	0.460	0.575
	TSD	**0.707**	**0.750**	**0.699**	**0.756**	**0.984**	**0.978**	**0.988**	**0.966**	**0.507**	**0.581**
DenseNet121	G-RGB	0.142	0.762	0.14	0.758	0.112	0.592	0.118	0.620	0.112	0.544
	GTS	0.220	0.849	0.232	0.846	0.238	0.650	0.248	0.702	0.224	0.565
	KD+BU	0.458	0.487	0.563	0.517	0.563	0.501	0.576	0.503	0.446	0.509
	RGB	0.986	0.982	0.966	0.981	0.938	0.972	0.879	0.923	0.226	0.680
	Noise	0.945	0.942	0.933	0.944	0.927	0.949	0.935	0.933	**0.818**	0.808
	TSD	**0.994**	**0.983**	**0.985**	**0.984**	**0.987**	**0.981**	**0.959**	**0.965**	0.776	**0.822**

can clearly see the difference between the original image and its corresponding adversarial images. Therefore, both RGB features and noise features can significantly reflect the difference between adversarial images and clean images. Although GRB and Noise only start from unilateral features, they can still distinguish adversarial samples well. From the results in MNIST dataset, it is difficult to judge which is better, RGB, Noise or TSD. Nevertheless, we can see that the performance of GTS is better than G-RGB. In this perspective, combining RGB characteristics with noise characteristics can indeed help the detection of adversarial samples.

4.3.2 | Verification on CIFAR-10

On CIFAR-10 dataset, see Table 13, in the face of white-box attacks (adversarial examples generated by the local model), our method (TSD) is superior to other methods when defending against FGSM, MIM, PGD and SPSA. RGB ranks behind our approach, and Noise ranks third. Although G-RGB and GTS have good AUC scores, their TPR scores are obviously not good enough, especially when defending against FGSM and SPSA, G-RGB and GTS categorize a goodly part of adversarial images with low confidences as clean. G-RGB has the best performance in resisting BIM. GTS ranks second, following by TSD. KD+BU shows the worst performance in resisting all attacks. The first row in Figure 5 shows an intuitive display of different methods when defending against adversarial examples produced by the local classifier. We can see that RGB, Noise and our approach have good generalization, they are trained by PGD can defend against other attack methods effectively. G-RGB and GTS trained by PGD have good effects against MIM, PGD and BIM, but the effects are poor in face of SPSA and FGSM. When defending against adversarial images generated by VGG16, our method's performance is still outstanding. The effect of RGB and Noise is similar to that of defending against adversarial examples produced by the local classifier. The three methods don't depend on the protected model, so they have good transferability, and can be reused to protect different models. The performance of G-RGB, GTS and KD+BU is significantly worse than that of distinguishing adversarial samples generated by the local model. These three detection methods are model-specific, they rely closely on the targeted model. It is hard for them to transfer the security

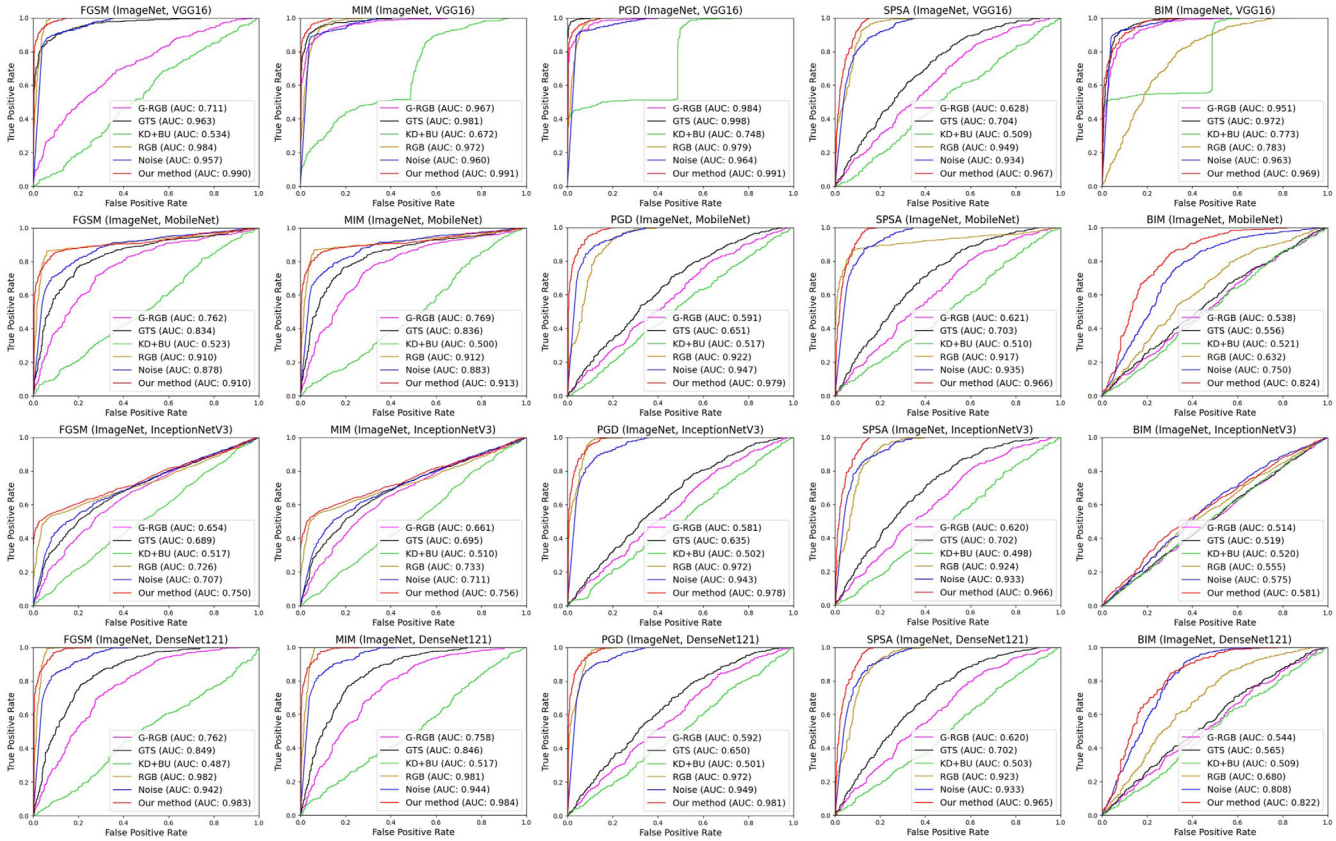

FIGURE 7 ROC curves of detection methods on ImageNet. The first row shows ROC curves of detection methods in distinguishing adversarial images produced by VGG16, i.e. the local classifier. And the second to fourth row show the six methods' ROC curves in distinguishing adversarial images generated by black classifiers

enhancement of a model to other models. The performance of the six methods in defending against adversarial images generated by MobileNet and ResNet50 is very similar to that of defending against adversarial images generated by VGG16. The second to forth row in Figure 5 show ROC curves of the six methods in defending against black-box attacks. We can intuitively see that although the performance of G-RGB, GTS and KD+BU is not good, GTS is obviously better than the other two methods due to it considers the noise characteristics.

4.3.3 | Verification on CIFAR-100

The images in CIFAR-100 are very similar to the images in CIFAR-10, but CIFAR-100 has 100 categories, so the classifiers for CIFAR-100 are more complex than the classifiers for CIFAR-10. As shown in Table 14, in defending against white-box attacks, our method is ahead of other methods in resisting FGSM, MIM, PGD and SPSA. Noise ranks second, and RGB ranks third. Although we rank RGB in the third place, other methods are not always worse than RGB. When defending against PGD, the AUC scores of G-RGB, GTS and KD+BU are all better than RGB, but their TPR scores are not as good as RGB. In resisting BIM, G-RGB has the best AUC score and KD+BU has the best TPR score. We consider TPR score is more important than AUC score, so we rank KD+BU in first and G-RGB in second. From the first row in Figure 6, we can see that the six methods are good when resisting MIM, PGD and BIM, G-RGB and KD+BU have poor performance in resisting FGSM and SPSA. On the whole, our approach has the best performance and generalization, following by Noise, RGB ranks third. The performance of KD+BU, GTS and G-RGB is not stable. In resisting black-box attacks, in addition to resisting BIM, our method is markedly superior to other methods in all other cases. Noise ranks second, following by RGB. However, in resisting BIM, Noise has the best performance, and TSD behinds Noise. The second row to forth row in Figure 6 shows ROC curves of different detection methods in resisting adversarial images produced by ResNet152, DenseNet169 and DenseNet201. We can intuitively see that KD+BU and G-RGB are not stable in defending against white-box attacks, and have poor performance in resisting black-box attacks, i.e. they do not have good generalization and transferability. GTS is better than KD+BU and G-RGB, but it is not effective enough to defend against black-box attacks. Our approach performs well in all cases, even for resisting BIM.

TABLE 16 Performance of detection methods in defending against adversarial examples with different perturbation intensity on ImageNet

		FGSM		MIM		PGD		SPSA		BIM	
Method	**Perturbationintensity**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**	**TPR**	**AUC**
G-RGB	2/255	0.030	0.618	0.062	0.722	0.063	0.734	0.006	0.513	0.064	0.735
	4/255	0.122	0.631	0.270	0.937	0.422	0.963	0.080	0.545	0.436	0.927
	6/255	0.260	0.663	0.548	0.946	0.860	0.977	0.180	0.585	0.864	0.930
	8/255	0.396	0.711	0.562	0.967	0.913	0.984	0.189	0.628	0.877	0.951
	10/255	**0.508**	**0.737**	**0.710**	**0.988**	**0.974**	**0.999**	**0.257**	**0.665**	**0.952**	**0.979**
GTS	2/255	0.144	0.623	0.294	0.662	0.400	0.714	0.020	0.522	0.404	0.716
	4/255	0.358	0.828	0.754	0.928	0.710	0.955	0.126	0.574	0.726	0.963
	6/255	0.504	0.935	0.928	0.964	0.962	0.995	0.234	0.638	0.969	0.966
	8/255	0.648	0.963	0.980	0.981	0.992	0.998	0.244	0.704	**0.996**	0.972
	10/255	**0.786**	**0.984**	**0.993**	**0.998**	**1.0**	**1.0**	**0.370**	**0.757**	0.994	**1.0**
RGB	2/255	0.316	0.673	0.266	0.654	0.310	0.641	0.606	0.755	0.212	0.606
	4/255	0.648	0.902	0.569	0.836	0.634	0.803	0.654	0.835	0.480	0.744
	6/255	0.820	0.956	0.854	0.926	0.872	0.899	0.824	0.936	0.578	0.807
	8/255	0.931	0.984	0.916	0.972	0.924	0.979	0.896	0.949	0.532	0.783
	10/255	**0.996**	**0.998**	**0.932**	**0.984**	**0.991**	**0.987**	**0.935**	**0.973**	**0.644**	**0.832**
Noise	2/255	0.800	0.819	0.808	0.835	0.888	0.826	0.832	0.829	0.880	0.822
	4/255	0.874	0.913	0.898	0.944	0.977	0.951	0.898	0.901	0.906	0.945
	6/255	0.903	0.927	0.918	0.958	**0.998**	0.962	0.904	0.922	0.908	0.960
	8/255	0.922	0.957	0.925	0.960	**0.998**	0.964	0.903	**0.934**	0.912	0.963
	10/255	**0.934**	**0.991**	**0.958**	**0.961**	**0.998**	**0.965**	**0.997**	0.926	**0.987**	**0.965**
TSD	2/255	0.752	0.788	0.746	0.776	0.702	0.756	0.828	0.900	0.782	0.803
	4/255	0.960	0.976	0.894	0.946	0.904	0.938	0.982	0.946	0.902	0.928
	6/255	**0.998**	0.990	0.982	0.983	0.984	0.986	0.988	0.946	0.934	0.968
	8/255	**0.998**	0.990	**1.0**	0.991	0.998	0.991	0.994	0.967	0.944	0.969
	10/255	**0.998**	**0.999**	0.998	**0.993**	**0.999**	**0.995**	**0.996**	**0.976**	**0.980**	**0.983**

4.3.4 | Verification on ImageNet

Although the images in CIFAR-10 and CIFAR-100 are colour images, their sizes are small. Therefore, we also validate our method on a large-scale dataset, i.e. the ImageNet. Table 15 shows the detection results of the six approaches on ImageNet. When defending against the adversarial images generated by VGG16, TSD has outstanding performance in resisting FGSM, MIM, and SPSA. RGB behinds TSD, and Noise ranks third. In resisting PGD, GTS has the highest AUC score, and TSD has the highest TPR score. As we mentioned above, TPR is more important than AUC, we thus rank GTS in first. GTS has the best performance in defending against BIM, following by TSD. The first row in Figure 7 shows the ROC curves of the six methods in defending against white-box attacks, we can see that RGB, Noise and our method have good defence capability in resisting all attacks. GTS performs poorly in resisting SPSA, and G-RGB performs poorly in resisting FGSM and SPSA. When distinguishing adversarial images produced by black models, the performance of our method is similar as the performance on CIFAR-10. The difference is that the TPR scores of Noise in detecting adversarial samples generated by BIM with MobileNet and DenseNet121 are higher than the TPR scores of TSD. However, the difference between their TPR scores is small, and their TPR scores are much higher than that of other methods. The rankings of RGB and Noise are similar to that on CIFAR-100, Noise ranks second and RGB ranks third. The second row to forth row in Figure 7 shows the ROC curves of different approach in resisting black-box attacks. We can see that G-RGB, GTS and KD+BU have poor performance in almost all cases when defending against black-box attacks. RGB, Noise and our method are not stable enough in distinguishing the adversarial images generated with InceptionNetV3. When defending against BIM, Noise and TSD are significantly better than RGB.

Overall, the generalization ability of G-RGB, GTS and KD+BU are poor, they cannot keep strong defensive ability to other attacks. Meanwhile, they rely too much on the knowledge of the targeted model and do not have good transferability. Nevertheless, GTS is obviously better than G-RGB and KD+BU. GTS considers noise features, which provide a wealth of additional information for adversarial sample detection. RGB, Noise

and our method do not rely on the targeted model, they do not care which model the adversarial samples are generated from, therefore, they have good transferability. Through the verification on the four datasets, we can see that the AUC and TPR scores of our approach are both very high, which indicates that whether adversarial examples or clean examples, our method can classify them correctly.

4.4 | Detection ability under different perturbation intensity

To test the detection ability of different methods under different perturbation intensity, we set $\varepsilon = 2/255, 4/255, 6/255, \varepsilon = 8/255$ and $10/255$ in Formula (2) for ImageNet. All adversarial training images are generated with $\varepsilon = 8/255$. Since KD+BU is not outstanding in all cases, we will not compare it here. Table 16 shows the defence ability of the five defence methods under different perturbation intensity. Except for a few exceptions, all methods follow the pattern that the greater the perturbation intensity, the better the defence effect. It is worth noting that when the perturbation intensity is greater than 2/255, Noise and TSD can effectively defending against all attacks. When the perturbation intensity is 2/255, their defence effect decreases significantly, but their recall rates can still exceed 0.7, and the recall rate of Noise can even exceed 0.8. However, the defensive effect of RGB drops off a cliff as the perturbation intensity decreases. This shows that the noise stream can discover more subtle difference between clean images and adversarial images, which can provide strong additional evidence for adversarial example detection.

5 | CONCLUSION

In this paper, we propose a novel method using an RGB stream and a noise stream to learn rich features for adversarial example detection. We extract the local noise features by SRM, which amplifies the inconsistency between original images and adversarial images. Rely on the additional evidence, our method can be independent of the protected model. That is, our method has strong transferability, and it can be reused to protect different model after once training. Experiments on standard datasets show that our method has excellent performance on both white-box and black-box attacks. In our future work, we aim to remove full connection layers from our model architecture. Full connection layers specify the size of input images. If our method after once training can detect adversarial examples with different sizes, the application scope of our method will greatly improve.

ACKNOWLEDGEMENTS

This study was supported by the National Natural Science Foundation of China (No. 61863036). The authors thank for the support of the Project of Yunnan Provincial Science and Technology Department (No. 201901BB050076) and the Project of Provincial Industrial Internet (No. TC200H01C).

CONFLICT OF INTEREST

The authors have declared no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Song Gao https://orcid.org/0000-0002-7169-6370

REFERENCES

1. Jiang, L., et al.: Robust RGB-D face recognition using attribute-aware loss. IEEE Trans. Pattern Anal. Mach. Intell. 42(10), 2552–2566 (2020)
2. Li, L., et al.: Moving object detection in video via hierarchical modeling and alternating optimization. IEEE Trans. Image Process. 28(4), 2021–2036 (2019)
3. Hu, J., et al.: Squeeze-and-excitation networks. IEEE Trans. Pattern Anal. Mach. Intell. 42(8), 2011–2023 (2020)
4. Szegedy, C., et al.: Intriguing properties of neural networks. In: Proceedings of the International Conference on Learning Representations (ICLR). Banff (2014)
5. Goodfellow, I., et al.: Explaining and harnessing adversarial examples. In: Proceedings of the International Conference on Learning Representations (ICLR). San Diego (2015)
6. Kurakin, A., et al.: Adversarial examples in the physical world. In: Proceedings of the International Conference on Learning Representations (ICLR). Toulon (2017)
7. Madry, A., et al.: Towards deep learning models resistant to adversarial attacks. In: Proceedings of the International Conference on Learning Representations (ICLR). Vancouver (2018)
8. Papernot, N., et al.: The limitations of deep learning in adversarial setting. In: Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P). Saarbrucken, pp. 372–387 (2016)
9. Moosavi-Dezfooli, S.-M., et al.: DeepFool: A simple and accurate method to fool deep neural networks. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, pp. 2574–2582 (2016)
10. Dong, Y., et al.: Boosting adversarial attacks with momentum. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, pp. 9185–9193 (2018)
11. Uesato, J., et al.: Adversarial risk and the dangers of evaluating against weak attacks. In: Proceeding of the International Conference on Machine Learning (ICML). Stockholm (2018)
12. Su, J., et al.: One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. 23(5), 828–841 (2019)
13. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: Proceedings of the IEEE Symposium on Security and Privacy (S&P). San Jose, pp. 39–57 (2017)
14. Huang, R., et al.: Learning with a strong adversary. arXiv preprint arXiv: 1511.03034 (2015)
15. Kurakin, A., et al.: Adversarial machine learning at scale. In: Proceedings of the International Conference on Learning Representations (ICLR). Toulon (2017)
16. Xie, C., et al.: Feature denoising for improving adversarial robustness. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, pp. 501–509 (2019)
17. Song, C., et al.: Robust local features for improving the generalization of adversarial training. In: Proceedings of the International Conference on Learning Representations (ICLR). Addis Ababa, Ethiopia (2020)
18. Papernot, N., et al.: Distillation as a defense to adversarial perturbations against deep neural networks. In: Proceedings of the IEEE Symposium on Security and Privacy (S&P). San Jose, pp. 582–597 (2016)
19. Papernot, N., McDaniel, P.: Extending defensive distillation. arXiv preprint arXiv: 1705.05264 (2017)

20. Feinman, R., et al.: Detecting adversarial samples from artifacts. arXiv preprint arXiv: 1703.00410 (2017)
21. Yang, P., et al.: Ml-loo: Detecting adversarial examples with feature attribution. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). New York, pp. 6639–6647 (2020)
22. Metzen, J., et al.: On detecting adversarial perturbations. In: Proceedings of the International Conference on Learning Representations (ICLR). Toulon (2017)
23. Fridrich, J., et al.: Rich models for steganalysis of digital images. IEEE Trans. Inf. Forensics Secur. 7(3), 868–882 (2012)
24. Zhou, P., et al.: Learning rich features for image manipulation detection. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, pp. 1053–1061 (2018)
25. Lin, T., et al.: Bilinear CNN models for fine-grained visual recognition. In: Proceeding of the IEEE International Conference on Computer Vision (ICCV). Santiago, pp. 1449–1457 (2015)
26. Fukui, A., et al.: Multimodal compact bilinear pooling for visual question answering and visual grounding. In: Proceeding of the Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL. Austin (2016)
27. Cozzolino, D., et al.: Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection. In: Proceeding of the ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec). Philadelphia, pp. 159–164 (2017)
28. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, pp. 4353–4361 (2015)
29. Guo, D., et al.: End-to-end feature fusion Siamese network for adaptive visual tracking. IET Image Process. 15(1), 91–100 (2020)
30. Rao, Y., Ni, J.Q.: A deep learning approach to detection of splicing and copy-move forgeries in images. In: Proceeding of the IEEE International Workshop on Information Forensics and Security (WIFS). Abu Dhabi, pp. 1–6 (2017)
31. Lyu, C., et al.: A unified gradient regularization family for adversarial examples. In: Proceeding of the IEEE International Conference on Data Mining (ICDM). Atlantic City, pp. 301–309 (2015)
32. Ross, A., Doshi-Velez, F.: Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). New Orleans, pp. 1660–1669 (2018)
33. Gu, S., Rigazio, L.: Towards deep neural network architectures robust to adversarial examples. In: Proceedings of the International Conference on Learning Representations (ICLR). San Diego (2015)
34. Liao, F., et al.: Defense against adversarial attacks using high-level representation guided denoiser. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, pp. 1778–1787 (2018)
35. Jia, X., et al.: ComDefend: An efficient image compression model to defend adversarial examples. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, pp. 6077–6085 (2019)
36. Mazumdar, A., Bora, P.: Siamese convolutional neural network-based approach towards universal image forensics. IET Image Process. 14(13), 3105–3116 (2020)
37. Xie, C., et al.: Mitigating adversarial effects through randomization. In: Proceedings of the International Conference on Learning Representations (ICLR). Vancouver (2018)
38. Dziugaite, G., et al.: A study of the effect of JPG compression on adversarial images. arXiv preprint arXiv: 1608.00853 (2016)
39. Guo, C., et al.: Countering adversarial images using input transformations. In: Proceedings of the International Conference on Learning Representations (ICLR). Vancouver (2018)
40. Das, N., et al.: Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression. arXiv preprint arXiv: 1705.02900 (2017)
41. Bhagoji, A., et al.: Enhancing robustness of machine learning systems via data transformations. In: Proceedings of the Annual Conference on Information Sciences and Systems (CISS). Princeton, pp. 1–5 (2018)
42. Liu, J., et al.: Detection based defense against adversarial examples from the steganalysis point of view. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, pp. 4820–4829 (2019)

How to cite this article: Gao, S., Yu, S., Wu, L., Yao, S., Zhou, X.: Detecting adversarial examples by additional evidence from noise domain. IET Image Process. 16, 378–392 (2022). https://doi.org/10.1049/ipr2.12354