Detecting Generated Images by Real Images

Bo Liu, Fan Yang, Xiuli Bi*
Bin Xiao*, Weisheng Li and Xinbo Gao

Chongqing University of Posts and Telecommunications, China
boliu@cqupt.edu.cn S200201074@stu.cqupt.edu.cn
{bixl,xiaobin,liws,gaoxb}@cqupt.edu.cn

Abstract. The widespread of generative models have called into question the authenticity of many things on the web. In this situation, the task of image forensics is urgent. The existing methods examine generated images and claim a forgery by detecting visual artifacts or invisible patterns, resulting in generalization issues. We observed that the noise pattern of real images exhibits similar characteristics in the frequency domain, while the generated images are far different. Therefore, we can perform image authentication by checking whether an image follows the patterns of authentic images. The experiments show that a simple classifier using noise patterns can easily detect a wide range of generative models, including GAN and flow-based models. Our method achieves state-of-the-art performance on both low- and high-resolution images from a wide range of generative models and shows superior generalization ability to unseen models. The code is available at `https://github.com/Tangsenghenshou/Detecting-Generated-Images-by-Real-Images`.

Keywords: Image forensics, forgery detection, image noise, frequency domain analysis, GAN, generated images

1 Introduction

Can you find out the fake images in Fig. 1? The answer is that all the images are fake. The popularity of deep neural networks has driven the rapid development of synthesis technology. Various mind-boggling technologies have entered our lives, from image editing to composite scenes, from face attribute tampering to face-swapping. For example, in the GPU technology conference hosted by Jen-Hsun Huang at NVIDIA in 2021, the video and Jen-Hsun Huang himself were synthesized, successfully fooling most people and bringing the image forgery to the limelight. Meanwhile, the concerns about image synthesis technology are growing as making global tampering becomes very easy. In particular, impressive progress has been made on generative models such as Generative Adversarial Networks(GAN) [1] and its variants. Examples include conditional GANs such as CycleGAN [2] based on unpaired data, StarGAN [3] that uses a generator and

* Corresponding Author

Fig. 1. Which pictures are real and which are fake?

a discriminator to learn mappings between multiple domains, and GauGAN [4] that uses spatially adaptive normalization; unconditional GANs such as BigGAN [5] based on orthogonal regularization, ProGAN [6] using feature vector normalization of pixels, and StyleGAN [7] using nonlinear mapping networks and an improved version of StyleGAN2 [8]. The other generative models, such as HiSD [9] based on hierarchical style decoupling, and the flow model Glow [10] based on reversible 1×1 convolution, can also produce high-quality generated images. Currently, many generated images can deceive the human eyes. Therefore it is urgent to pay more attention to image forensics. This paper proposed a detection method to expose globally tampered images yielded by generative models.

Generated image detection methods can be divided into two main categories: artifacts detection and data-driven approaches. The former detects artifacts in the spatial domain in generated images left by the upsampling components of networks or the periodical signals in the frequency domain. They are effective for most of the generated images in low quality by checking the traces generated by conditional GANs during upsampling. However, they become ineffective to unconditional GANs with high image quality. The data-driven approaches learn a large number of real and fake images, making the classifier learn the common features in GAN-generated images. However, the classifier is susceptible to unseen models and therefore does not generalize well as it is impossible to learn the common features shared by all generative models. A generic data-driven-based approach is introduced by Wang et al. [11]. However, such methods pay attention to the characteristics of generated images, resulting in generalization issues. We perform forgery detection from the perspective of real images. Specifically, we learn the shared properties of real images so that the detection network can work across various generative models, even with unseen models.

In this paper, we rethink the relationship between real and generated images. Analysis shows that real images possess spatial and frequency domain features not presented in generated images. This discrepancy can be observed in the representations of the image noise under the high-dimensional spatial mapping of the neural network, which we call the Learned Noise Patterns (LNP). We used a network to classify real and generated images with the help of LNP. Using LNP can effectively suppress the high-frequency information of images and reduce the

influence of image semantics on classification. In order to make full use of the information in the LNP, we utilized the amplitude and phase spectrum of images along with the LNP so that the network uses the spatial and frequency domain features.

To sum up, this paper proposes a method to detect generated images. The main contributions of this paper can be summarised as follows:

• Our frequency domain analysis of the noise patterns of real images reveals its consistency in real images, while the generated images are far different.

• We discriminate the generated images by their inconsistent noise patterns to real images rather than detecting the artifacts or patterns of generated images.

• The proposed detection method achieves the SOTA performance in publicly available datasets and shows superior generalization ability to unseen models.

2 Related Work

Existing methods for detecting generated images can be classified into image artifacts detection and data-driven approaches. For those focus image artifacts, Dang et al. [12] found that the spatial information of the tampered region is important, and the tampered region is located by estimating the attention map of a particular image. Liu et al. [13] proposed GramNet, proving that CNNs consider texture as an important factor while finding that the texture statistics of real and false images differ significantly. Zhao et al. [14] used the attention mechanism to improve detection performance by extracting texture information at shallow and locating forgery at deep levels. Zhang et al. [15] introduce a generator that simulates sampling artifacts on several common GANs and demonstrates superior performance in the frequency domain by learning to classify sampling artifacts on GANs in both the spatial and frequency domains. It is argued in [16] that local information is easier to extract helpful information than global information. Frank et al. [17] demonstrate that upsampling operations in the pipeline cause artifacts in GAN-generated images, and the detection is performed using the DCT transform. Durall et al. [18] show that commonly used up-sampling operations (deconvolution or transposed convolution) make such models fail to reproduce the spectral distribution of the training data correctly.

For data-driven methods, Wang et al. [11] directly trained ResNet50 as the classifier with a large number of real images and ProGAN-generated images, which can be well generalized to the detection of different generative models using global information. On this basis, Gragnaniello et al. [19] used the modified ResNet50 network with two fewer down-sampling layers to improve the detection performance but significantly increase the training time.

Image noise is widely utilized in local tampering and source device identification. Each device will have its specific fingerprints left on the shooting process, which is also caused by imperfections in the manufacturing process of the device, and this pattern is called the PRNU noise pattern. Therefore, the equipment identification can be performed based on these fingerprints, e.g. [20]. Based on the specific properties of PRNU, Davide et al. [21] introduce a method

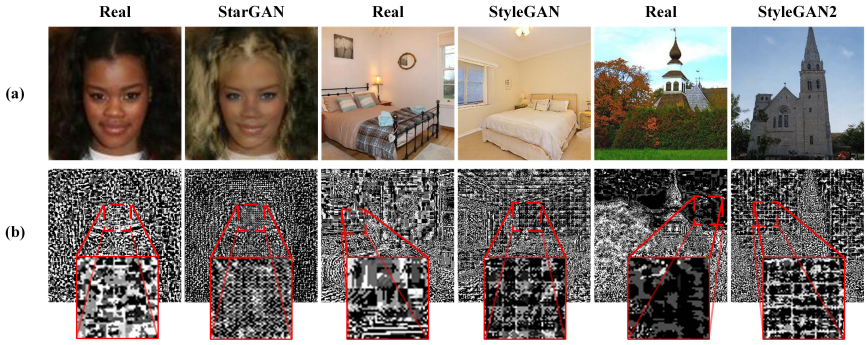

Fig. 2. (a) row represents the real and generated images, and (b) row represents their LNP. The first, third and fifth columns are real images. The second, fourth and sixth columns are images generated by StarGAN, StyleGAN and StyleGAN2, respectively. The red box indicates that the generated images show the grid effect.

that learns the camera noise by denoising the network for local forgery detection. Ghosh et al. [22] extract noise fingerprints to identify real patches and forged patches for local tampering detection.

However, the previous approaches, whether detecting artifacts or by data-driven, look for fingerprints left by generative models, resulting in lower versatility. Instead, we focus on learning the common properties of real images to avoid generalization issues.

3 Method

3.1 Learned Noise Patterns (LNP)

Although the generated images from early GAN models are easy to detect, with the development of unconditional GAN such as StyleGAN and stylegan2, the current GAN-generated images have become more and more realistic. As shown in Fig. 2(a), we can hardly distinguish between StyleGAN (column 4), StyleGAN2 (column 6), and the real images (columns 3 and 5) with naked eyes. It is necessary to extract the discriminative features of the images to amplify their differences.

In the imaging process, a camera converts photons into electrons, and then the signal goes through components such as digital-to-analog converters. As the photons enter a camera, the incident light intensity at various places in a real image will show no regularity. Therefore, the pixel values do not change periodically for most real images.

In the pipeline of GANs for generating images, papers such as [15] have introduced the artifacts in the generated images due to up-sampling operations. However, in unconditional GANs with high generation quality, the artifacts are

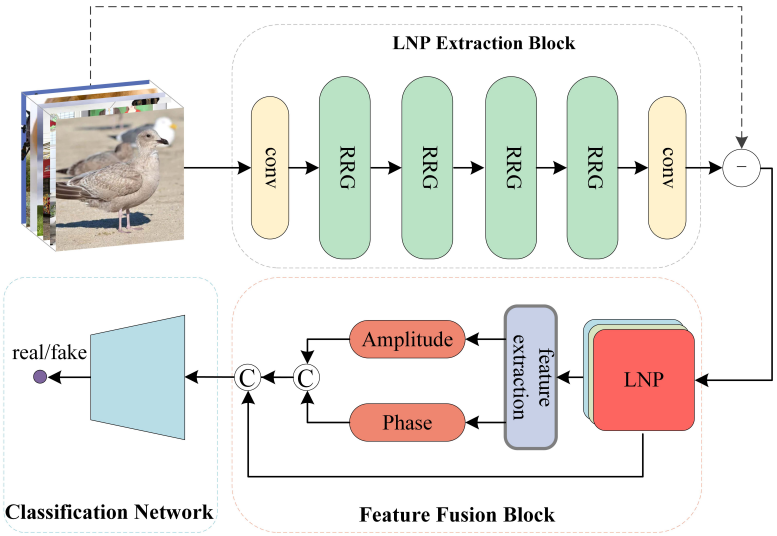

Fig. 3. The structure of the image verification network. ⓒindicates concatenation.

not apparent in the spatial domain. Moreover, a large amount of semantic information in the spatial domain interferes with the classifier's performance. Existing methods directly use images for classification. Although good results can be achieved after extensive training, their generalization performance has room to improve. For example, for the popular generative models, detection results are not satisfied (Table 3). It is because the classifier focuses too much on artifacts in fake images, but different generative models produce different artifacts. In order to discriminate generated images from real images, we should find a feature or a pattern shared only by real images.

The exclusive pattern of real images can be extracted in image noise space, and neural networks can learn this pattern. For real images, the smooth regions show different patterns depending on the light intensity, as in column 1, column 3, and column 5 in Fig. 2(b). However, in the images generated by the GANs, the smooth regions exhibit checkerboard patterns, exhibiting periodicity, as in columns 2, 4, and 6 in Fig. 2(b).

Our goal is to find common properties in real images, so we do not need semantic information of images. A denoising network takes a set of noisy images and outputs a set of clean images after denoising. Therefore, the denoising network can maintain detailed information such as the edge texture of the original image. Then we can use the original image minus the denoised image to obtain the noise pattern without semantic interference, and we name it Learned Noise Patterns (LNP). The denoising network can be described as

$$Dst = F(Src(x, y)), \tag{1}$$

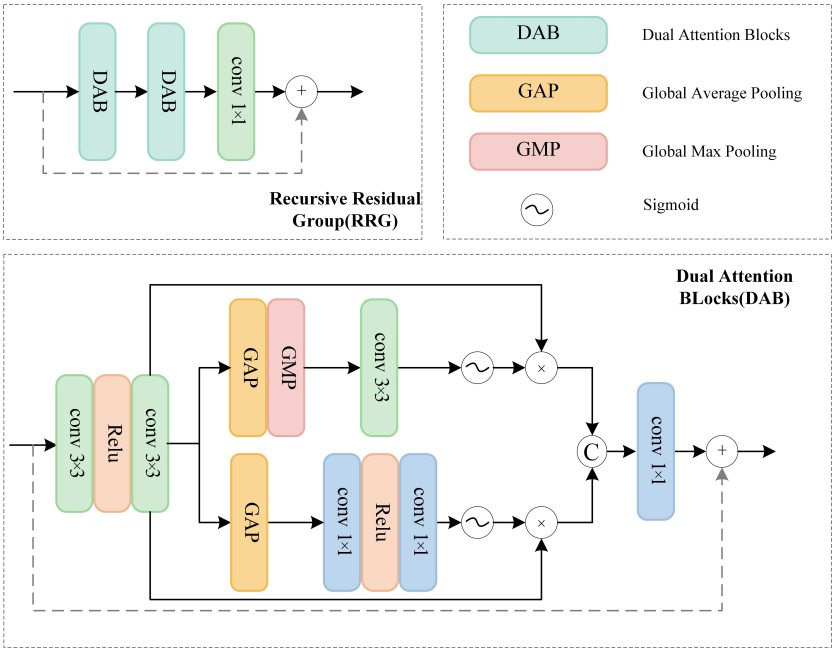

Fig. 4. The structure of the RRG module.

where $Src(x, y)$ denotes the input noisy image, and $F(\cdot)$ denotes the denoising network, and Dst denotes the final clean image. We use the result of $Src(x, y) - Dst$ as LNP.

The early denoising networks, such as DNCNN, add Gaussian white noise (AWGN) to images to form training data. It is superficial and very different from the real world since there are not just AWGNs in real images. To simulate the real-world scene, [23] uses an RGB image to construct its RAW image, adding noise to the RAW image and then converting the RAW image to an RGB image to simulate the process of a real camera shot. To extract more real noise patterns, we used CycleISP [23] denoising network (LNP extraction block in Fig. 3) which was trained on real image dataset and synthetic dataset. We can then use this denoising network to extract LNP from images. For an image $I_{in}(x, y)$, $1 \leq x \leq M, 1 \leq y \leq N$, where M and N are the size of that image, it will be processed as

$$M_0 = K_3(I_{in}(x, y)), \tag{2}$$

where K_3 denotes a 3×3 convolution, and M_0 contains multiple feature maps with low-level features. Then, we used the Recursive Residual Group module (RRG) (Fig. 4) to further process the features. The RRG module is composed of two Dual Attention Blocks (DABs). Each DAB calibrates the features by two types of channel attention and spatial attention. This process can be expressed

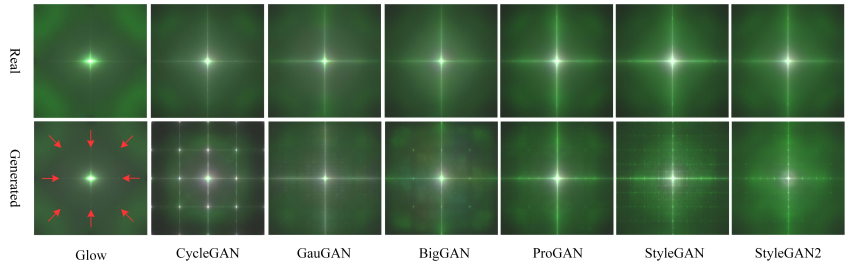

Fig. 5. The amplitude spectrum is plotted by averaging all the original or generated images for each GAN model from the dataset provided in [11]. The top indicates the average amplitude spectrum of LNP of the real images in their dataset, and the bottom shows the average amplitude spectrum of the LNP of the generated images from each model. The red arrows indicate peaks.

as

$$M_1 = RRG(RRG(RRG(RRG(M_0)))). \tag{3}$$

Finally, the three-channel feature map M_2 can be obtained by $M_2 = K_3(M_1)$. The extracted LNP is $I_{LNP} = -M_2$.

3.2 LNP Amplitude Spectrum

The LNP characteristics of real images are not fully shown in the spatial domain. For a better exploration of the LNP, we analyzed its amplitude spectrum. For an image of M × N size, its two-dimensional discrete Fourier transform can be described as

$$\mathrm{F}(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-i2\pi ux/M} e^{-i2\pi vy/N}. \tag{4}$$

where $F(u,v)$ denotes the frequency component at frequency domain (u,v) and $f(x,y)$ is the gray value at point (x,y) in the spatial domain of a channel of the input image. The high frequency corresponds to the part of the image where the pixel value changes drastically. And the low frequency corresponds to the flat area of the image. The amplitude spectrum A in the frequency domain can be expressed as

$$A(u,v) = \sqrt{R^2(u,v) + I^2(u,v)}. \tag{5}$$

where $R(u,v)$ and $I(u,v)$ denote the real and imaginary parts of $F(x,y)$, respectively.

Fig. 5 shows the averaged amplitude spectrum of LNP of all images from each generative model, including the generated and real images provided by [11]. For GauGAN [4], BigGAN [5], ProGAN [6], the networks use the nearest neighbor

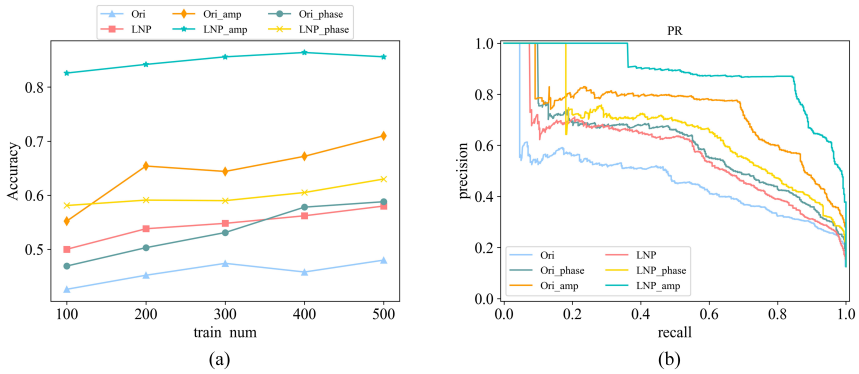

Fig. 6. Multiple classifications with SVM to discern original images, CycleGAN [2], StarGAN [3], GauGAN [4], BigGAN [5], ProGAN [6], StyleGAN [7] and StyleGAN2 [8], where the test sets include 500 images. The horizontal coordinate of plot (a) indicates the number of training images, and the vertical coordinates indicate the test accuracy. (b) represents the PR curves for 500 images in each training set.

Table 1. Accuracy in the CycleGAN and StarGAN datasets using OC-SVM in the amplitude spectrum of the original images compared to the LNP amplitude spectrum.

Dataset	CycleGAN [2]			StarGAN [3]		
Method	Ori_amp	LNP_amp	Gain	Ori_amp	LNP_amp	Gain
Accuracy	47.80%	73.80%	+26.0%	49.90%	98.80%	+48.90%

interpolation for upsampling with a period of 4. In contrast, StyleGAN [7] and StyleGAN2 [8] use bilinear interpolation for up-sampling, and we can see that the periodicity on the amplitude spectrum of their LNP is 8. For CycleGAN [2], upsampling is performed using deconvolution, and the LNP amplitude spectrum has strong vibrations with a period of 4. Glow [10] uses linear interpolation, and its LNP amplitude spectral period is also 4. Since the generated image has a prominent periodicity, the original image can easily be distinguished for lacking grid artifacts. For real images, their LNP are very similar in the frequency domain. Therefore, we can distinguish generated images by learning the special properties of real images.

To demonstrate the discriminative ability of LNP, we used eight datasets of real images, CycleGAN [2], StarGAN [3], GauGAN [4], BigGAN [5], ProGAN [6], StyleGAN [7] and StyleGAN2 [8] for multi-classification. Fig. 6(a) shows that LNP has better classification results than using the original images, and using the amplitude spectrum of LNP has a great improvement compared to using the original images. Fig. 6(b) shows better performance in PR curves.

To demonstrate the superiority of using LNP compared to the original images, we trained one-class SVM (OC-SVM) by the amplitude spectrum of the real images only. Table 1 shows that our method achieves good performance on

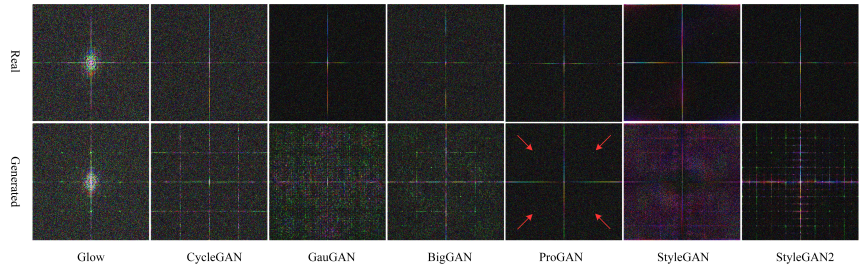

Fig. 7. The phase spectrum is plotted by averaging all the original and generated images for each GAN model from the dataset provided in [11]. The top indicates the average phase spectrum of LNP of the real images in their dataset, and the bottom shows the average phase spectrum of the LNP of the generated images in the dataset for the model. The red arrows indicate peaks.

the one-class classification task that only learns the amplitude spectrum features from real images. More experimental details are in subsection 4.3.

3.3 LNP Phase Spectrum

The phase spectrum can be described as

$$\phi(u, v) = \arctan[\frac{I(u, v)}{R(u, v)}]. \tag{6}$$

The frequency spectrum does not contain all the information in the frequency domain. Take the basic sine wave for example, the different phases determine the position of the wave. In addition to the frequency spectrum (amplitude spectrum), we also included the phase spectrum. Neural networks are more concerned with pixel information and will learn more information about the amplitude spectrum but lack the ability to learn structural information directly [24]. The phase spectrum contains more structural information in the image. Thus, we can fully use the image information by using the phase spectrum. As in Fig. 7, we can find that the LNP of real images in the phase spectrum is similar to the amplitude spectrum. Both have similar characteristics. The dataset used for the real images of the Glow model is Celeba-HQ, which used post-processing such as face alignment and cropping on the Celeba dataset. Therefore, the LNP of its real image is slightly different from the rest of the images. In general, The phase spectrum of the LNP exhibits a grid effect and is therefore also periodic as well. We have verified in Fig. 6 that the LNP phase spectrum is easier to extract useful information than the original image phase spectrum. The accuracy of using the LNP phase spectrum and PR curve is better than using the original image phase spectrum.

Table 2. The specifications of test datasets.

Generative Model Type	Generative Model	Source	Nums
Conditional Model	StarGAN [3]	CelebA	4k
	CycleGAN [2]	Style/object transfer	2.6k
	GauGAN [4]	COCO	10k
	HiSD [9]	ClelebA	4k
Unconditional Model	BigGAN [5]	ImageNet	8k
	ProGAN [6]	LSUN	8k
	Glow [10]	CelebA-HQ	2k
	StyleGAN [7](LR)	LSUN	12k
	StyleGAN [7](HR)	FFHQ	5k
	StyleGAN2 [8](LR)	LSUN	16k
	StyleGAN2 [8](HR)	FFHQ	2k

3.4 LNP Network

The above analysis shows that LNP has a good discriminative ability. Compared to solely using real images for training, the amplitude spectrum information can improve the classification performance of the network, and the phase spectrum provides more contour information in the frequency domain. Therefore, we started from the perspective of real images and found the commonality that real images have. We built the network architecture in Fig. 3 by making the LNP blend with its amplitude spectrum and phase spectrum.

4 Experiments

4.1 Datasets

We used 20 classes of images provided in [11], which contain 362K real images with 362K images generated by ProGAN [6] as the training set, 4k images generated by ProGAN [6] with 4k real images as the validation set. For a fair comparison, we evaluated the publicly available dataset in [11], the GAN-generated image dataset, and the face dataset. These include conditional generative models (StarGAN [3], CycleGAN [2], GauGAN [4]), and unconditional generative models (BigGAN [5], ProGAN [6], StyleGAN [7], StyleGAN2 [8]). In order to fully validate the effectiveness of our method, we added non-GAN generative models to our test set: HiSD [9] and Glow [10]. The details of each generative model are shown in Table 2. The StyleGAN [7], StyleGAN2 [8] dataset contains low resolution (LR) images (256×256 resolution) and high resolution (HR) images (1024×1024 resolution), where the HR images are selected from the FFHQ face dataset. The real images for the Glow [10] model were selected from the CelebA-HQ dataset. For the HiSD [9] model we used the officially published pre-trained model without any post-processing.

http://www.grip.unina.it/download/DoGANs/.
http://www.seeprettyface.com/information.html.
https://github.com/imlixinyang/HiSD.

Table 3. The comparison of the accuracy with the state-of-the-art methods. We used only ProGAN on both the training and validation sets.

Method	LR									HR		AVG
	Cycle GAN	Star GAN	Gau GAN	Big GAN	Pro GAN	HiSD	Glow	Style GAN	Style GAN2	Style GAN	Style GAN2	
AutoGAN-Spec(19')	75.3	81.2	73.4	74.9	76.9	69.3	49.5	59.7	53.3	84.9	84.2	71.2
DCT-CNN(20')	67.8	49.7	51.7	42.6	57.4	56.0	47.6	60.1	55.9	53.2	52.1	54.0
Wang(20')	83.9	90.9	77.0	75.7	99.9	80.2	27.0	91.6	90.9	89.8	88.5	81.4
Gragnaniello(21')	71.9	**100**	56.5	68.9	**100**	**98.4**	40.9	88.7	**98.9**	95.6	96.6	83.3
Ours	**91.6**	**100**	**79.7**	**88.1**	99.1	95.9	**80.0**	**96.0**	92.3	**99.1**	**98.8**	**92.8**

Table 4. The comparison of AP with the state-of-the-art methods.

Method	LR									HR		AVG
	Cycle GAN	Star GAN	Gau GAN	Big GAN	Pro GAN	HiSD	Glow	Style GAN	Style GAN2	Style GAN	Style GAN2	
AutoGAN-Spec(19')	83.3	81.4	78.7	71.6	85.9	73.6	40.1	60.7	55.1	92.4	91.5	74.0
DCT-CNN(20')	50.5	38.8	48.3	42.6	47.3	31.6	53.6	43.0	42.5	55.4	39.9	44.9
Wang(20')	91.5	98.1	79.1	77.3	**100**	89.8	33.2	98.5	99.1	96.2	99.6	87.5
Gragnaniello(21')	79.1	**100**	60.0	67.4	**100**	**100**	33.7	98.9	**100**	99.9	**99.9**	85.4
Ours	**98.1**	**100**	**83.3**	**95.2**	**100**	**100**	**68.6**	**99.6**	98.9	**100**	99.5	**94.8**

4.2 Setup

In our experiments, ResNet50 pre-trained in ImageNet was used. Training was performed using the Adam training optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with a batch size of 256 and an initial learning rate of 1e-4. It is worth noting that if the validation set accuracy does not rise within five epochs, the learning rate decays by a factor of ten, with a minimum learning rate of 1e-6. The validation set was indirectly involved in the training, so only ProGAN [6] was used for the validation set. Our model did not see images from other generative models during the training period except for ProGAN. All training processes were implemented on an NVIDIA Tesla V100 (32G) GPU.

4.3 Comparisons

We utilized OC-SVM to discriminate generated images using the amplitude spectrum of original images and the LNP amplitude spectrum (Table 1). We used 1000 real images from the StarGAN dataset as the training set, while 1000 fake images and 1000 real images as the test set. In the CycleGAN dataset, 500 real images were used as the training set, and 500 fake and 500 real images were used

Table 5. The comparison of F1-Score with the state-of-the-art methods.

Method	LR									HR		AVG
	Cycle GAN	Star GAN	Gau GAN	Big GAN	Pro GAN	HiSD	Glow	Style GAN	Style GAN2	Style GAN	Style GAN2	
AutoGAN-Spec(19')	0.750	0.815	0.733	0.752	0.763	0.694	0.326	0.598	0.492	0.848	0.850	0.693
DCT-CNN(20')	0.708	0.660	0.606	0.553	0.606	0.689	0.542	0.644	0.617	0.180	0.173	0.544
Wang(20')	0.830	0.905	0.758	0.762	0.999	0.814	0.412	0.922	0.911	0.889	0.869	0.825
Gragnaniello(21')	0.623	1.0	0.561	0.574	**1.0**	**0.981**	0.147	0.897	**0.987**	0.954	0.965	0.790
Ours	**0.914**	**1.0**	**0.765**	**0.877**	0.991	0.961	**0.795**	**0.961**	0.928	**0.991**	**0.988**	**0.925**

as the test set. The experimental results show that our method can extract more useful information than the original images. Moreover, our method can effectively distinguish real images from fake images based on the common attributes of real images.

We compared our method with three state-of-the-art deep learning methods for generated image detection: Zhang et al. [15], Frank et al. [17], Wang et al. [11], and Gragnaniello et al. [19]. Table 3, Table 4 and Table 5 report the accuracy, AP and $F1$ values with the threshold of 0. For the others methods, we chose the best result in three experiments. [19] removed two down-sampling layers in ResNet50, so the training and testing time is ten times larger than our method. Our method achieves excellent performance on LR images and good generalization performance. Our experiments show that we have good performance not only on the GAN-generated images but also on other generative models, such as the flow-based models (Glow [10]). Our average accuracy across all models is over 90%. In the HR test set, we used real images different from the LR ones to avoid reusing data. On HR images, an average accuracy of over 98% was achieved, with an increase of around 10% compared to the rest of the methods.

4.4 Ablation Study

LNP Extraction Block In order to provide a more suitable LNP Extraction Block, five different models were compared, including CycleISP [23], DNCNN [25], CBDNet [26], DeamNet [27] and InvDN [28]. We compared the accuracy of the five models on the test set, trained in line with section 4.3. The results are shown in Table 6. We found that the information extracted by CycleISP is more favorable and can significantly improve the experimental results.

Feature Fusion Block To evaluate the necessity of the individual components of our model, we used accuracy and mAP on both LR images and HR images. Detection results are presented in Table 7. We first evaluated the performance of using the LNP alone, which performs well. We then used the amplitude spectrum for single and three channels and the phase spectrum for single and three

Table 6. The performance of different denoising network in the LNP Extraction Block.

Method	LR			HR		
	ACC	mAP	F1	ACC	mAP	F1
DNCNN [25]	78.2	90.3	82.7	75.9	87.2	0.808
CBDNet [26]	80.5	85.4	80.8	93.3	97.3	0.927
DeamNet [27]	81.2	88.7	84.8	91.5	98.3	0.926
InvDNInvDN [28]	76.6	83.7	81.2	54.6	74.8	0.682
CycleISP [23]	**91.4**	**93.7**	**91.0**	**98.9**	**99.7**	**0.989**

Table 7. The ablation study of the Feature Fusion Block.

LNP	Amp (1 channel)	Amp (3 channels)	Phase (1 channel)	Phase (3 channels)	LR		HR	
					ACC	mAP	ACC	mAP
✓					87.2	92.9	97.6	**99.7**
✓	✓				84.3	91.8	89.8	99.4
✓		✓			88.1	93.3	97.8	**99.7**
✓			✓		85.3	91.8	97.3	**99.7**
✓				✓	89.3	92.2	95.9	99.2
✓		✓		✓	84.6	90.1	76.4	94.7
✓	✓		✓		**92.8**	**94.8**	**98.9**	**99.7**

channels for the input to the classification network. The experimental results show an improvement in the results using LNP and three-channel amplitude spectra and LNP and three-channel phase spectra. Using LNP, three-channel amplitude spectrum, three-channel phase spectrum combining the results into nine channels has a significant degradation. This is because the number of channels in the network does not change in ResNet50, but the proportion of LNP is reduced. Using the LNP, single-channel amplitude spectrum, and single-channel phase spectrum combined into a 5-channel feature ensures the dominance of the LNP. It allows the network to learn useful information about the amplitude and phase information.

Classification Backbone To verify the effectiveness of the different classification networks, we conducted experiments using VGG16, VGG19, ResNet18, ResNet34, and ResNet50. Fig. 8 shows our results. The VGG model has more parameters than the ResNet model and therefore is more accurate on LR images than ResNet18 and ResNet34. Since the VGG model has a fully connected layer, we cropped to 224^2 on HR images. As the number of layers on the network deepens, the results are optimal on ResNet50.

4.5 Robustness

In real-world scenes, images are subjected to various post-processing processes, such as blurring and cropping. We test our method against post-processing, in-

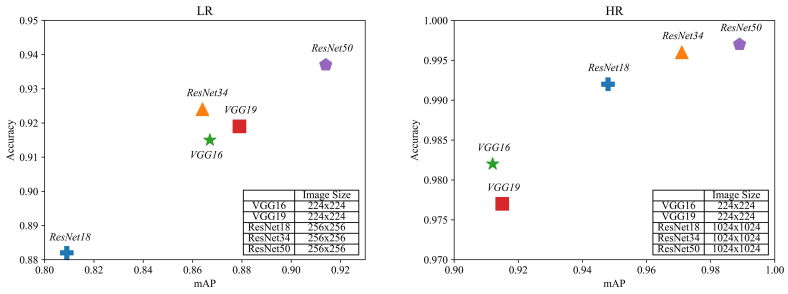

Fig. 8. The performance of different backbones in LR and HR.

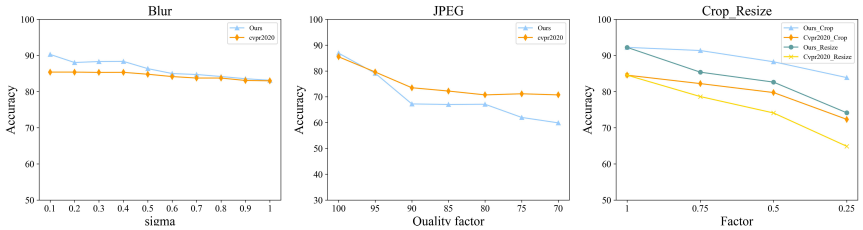

Fig. 9. The robustness of our model compared to Wang(CVPR20') [11].

cluding Gaussian blurring (sigma: 0.1∼1), JPEG quality factors (70∼100), image cropping, and resizing (cropping/scaling factor: 0.25∼1). We randomly selected 1000 images in each generative model dataset for robustness experiments. Figure 9 shows the robustness results of our comparison with [11]. Our model is better when blurring, cropping, and resizing. However, the results are lower in the JEPG compression case. The JPEG scheme generates multiple peaks in the frequency domain, similar to the periodicity in generated images. Therefore our method does not work well in the JPEG case. In the following work, we will solve this problem.

5 Conclusions

In this paper, we detect generated images using LNP of real images. We demonstrated that the LNP of real images are very similar in amplitude and phase spectrum, while the LNP of generated images is far different. Experimental results show that the method outperforms existing methods in image authentication. The superior generalization ability of the proposed method allows use in realistic scenes, even with future unseen models.

References

1. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
2. Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
3. Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.
4. Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.
5. Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.
6. Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
7. Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.
8. Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.
9. Xinyang Li, Shengchuan Zhang, Jie Hu, Liujuan Cao, Xiaopeng Hong, Xudong Mao, Feiyue Huang, Yongjian Wu, and Rongrong Ji. Image-to-image translation via hierarchical style disentanglement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8639–8648, 2021.
10. Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.
11. Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8695–8704, 2020.
12. Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 5781–5790, 2020.
13. Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8060–8069, 2020.
14. Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2185–2194, 2021.
15. Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2019.

16. Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European Conference on Computer Vision*, pages 103–120. Springer, 2020.
17. Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International Conference on Machine Learning*, pages 3247–3258. PMLR, 2020.
18. Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7890–7899, 2020.
19. Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are gan generated images easy to detect? a critical analysis of the state-of-the-art. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.
20. Davide Cozzolino, Francesco Marra, Diego Gragnaniello, Giovanni Poggi, and Luisa Verdoliva. Combining prnu and noiseprint for robust and efficient device source identification. *EURASIP Journal on Information Security*, 2020(1):1–12, 2020.
21. Davide Cozzolino and Luisa Verdoliva. Noiseprint: a cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2019.
22. Aurobrata Ghosh, Zheng Zhong, Steve Cruz, Subbu Veeravasarapu, Maneesh Singh, and Terrance E Boult. Infoprint: Information theoretic digital image forensics. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 638–642. IEEE, 2020.
23. Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2696–2705, 2020.
24. Guangyao Chen, Peixi Peng, Li Ma, Jia Li, Lin Du, and Yonghong Tian. Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 458–467, 2021.
25. Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.
26. Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1712–1722, 2019.
27. Chao Ren, Xiaohai He, Chuncheng Wang, and Zhibo Zhao. Adaptive consistency prior based deep network for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8596–8606, 2021.
28. Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon. Invertible denoising network: A light solution for real noise removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13365–13374, 2021.