Wiley
International Journal of Intelligent Systems
Volume 2025, Article ID 1546016, 22 pages
https://doi.org/10.1155/int/1546016

WILEY

Research Article

Generative Adversarial Networks With Noise Optimization and Pyramid Coordinate Attention for Robust Image Denoising

Min-Ling Zhu,[1] **Jia-Hua Yuan**,[1] **En Kong**,[1] **Liang-Liang Zhao**,[1] **Li Xiao**,[2] **and Dong-Bing Gu**[3]

[1]*Computer School, Beijing Information Science and Technology University, Beijing, China*
[2]*School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China*
[3]*School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK*

Correspondence should be addressed to Li Xiao; andrewxiao@bupt.edu.cn

Received 19 January 2024; Revised 18 November 2024; Accepted 4 February 2025

Academic Editor: Alexander Hošovský

Image denoising is a significant challenge in computer vision. While many models perform well in low-noise environments, their denoising capabilities are relatively weak under high-noise conditions. In addition, these models often overlook the robustness issues under adversarial attacks, leading to a marked decrease in denoising stability when facing malicious attacks. To address the challenges of achieving consistently high-quality denoising in both high-noise and low-noise environments, adapting to various complex scenarios with high robustness, and enhancing the model's resilience against attacks, we propose the NOP-GAN, a powerful image denoising model. This model modifies the GAN architecture by integrating a U-Net with a pyramid coordinate attention mechanism and a noise optimization algorithm into a generator of the GAN. Experimental results demonstrate that the NOP-GAN possesses superior performance in denoising tasks and robustness against adversarial attacks.

Keywords: attention mechanism; generative adversarial network; image denoising; noise optimization; robustness

1. Introduction

Image denoising is a challenging core problem in computer vision that has garnered widespread attention in recent years. Over the past two decades, traditional image processing methods have primarily relied on rule-based mathematical operations to smooth images and mitigate the effects of noise, such as median filtering [1, 2] and mean filtering [3, 4]. These methods have played a significant role in the field of image denoising. Rooted in well-defined mathematical principles, these techniques aim to enhance image quality by systematically reducing noise interference. However, they are often limited to specific types of noise, require manual parameter tuning, and lack generalization ability, making them inadequate for handling real noise characterized by complex distributions, often resulting in suboptimal denoising quality. In contrast, deep learning–based denoising methods, which learn the prior knowledge of images from large-scale data, can improve the quality of denoised images to a certain extent [5, 6]. Various deep learning methods have been developed for image denoising [7–9]. For instance, Zhang et al. proposed three classic neural network denoising models: denoising convolutional neural networks (DnCNNs) [10], fast and flexible DnCNNs (FFDNets) [11], and convolutional blind denoising networks (CBDNet) [12].

These models represent a new CNN architecture that significantly enhances image denoising performance through the clever combination of residual learning and batch normalization (BN) layers.

Chen et al. [13] were the first to leverage generative adversarial networks (GANs) for real noise modeling and build a paired training dataset. Kim et al. [14] improved the previous method by including more parameters such as clean image, International Organization for Standardization (ISO), and shutter speed as additional inputs to the generator. Yue et al. [15] designed a dual GAN to learn the joint distribution of noisy and clean pairs. In 2021, Zhang and Sakurai [16] used the conditioned explicit relativistic GANs to generate synthetic noisy images, which were mixed with real-world noisy images and used to train CNNs for denoising. Most recently, the blind image denoising method is based on the asymmetric GAN (ID-AGAN) [17], proposed by Wang et al., to optimize the high-dimensional image information denoising. Zhou et al. [18] improved the conditional GAN (cGAN) model and applied it to in-house optical coherence tomography (OCT) scanner for speckle noise suppression. Xie et al. [19] proposed a multitask GAN (MGAN) for retinal OCT image denoising.

However, although certain models can demonstrate good denoising performance in low-noise environments, their denoising capabilities remain relatively weak in high-noise environments, exhibiting a noticeable decline in performance. In addition, these models often overlook the robustness of image denoising models against adversarial attacks, leading to a sudden drop in denoising stability when facing malicious attacks.

To enhance the robustness of the denoising network, Yan et al. [20] proposed a hybrid adversarial training (HAT) strategy for deep image denoisers (DIDs) to enhance their adversarial robustness and generalization ability. Zhang et al. [21] proposed a robust deformed DnCNN (RDDCNN) for image denoising. The RDDCNN can extract more representative noise features by using a deformable learnable kernel and a stacked convolutional structure and reconstruct a clean image by using a residual block and an enhanced block. Furthermore, enhancing the robustness of image denoising models against attacks is crucial for ensuring their reliability and security in practical applications. Deep learning–based image denoising models are often highly sensitive to minor perturbations in input data [22], making them vulnerable to malicious attacks, such as adversarial attacks, which can lead to significant performance degradation or even complete failure, thereby impacting the model's reliability and security. Therefore, when designing neural networks for image denoising, it is also essential to investigate and develop effective antidisturbance algorithms to ensure the model's robustness in real-world applications. In summary, the aforementioned methods aim to recover clear and authentic images from noise-affected ones, preserving as much original detail and texture information as possible. In addition, these methods strive to handle various types and levels of noise effectively and enhance robustness across different environments to address a wide range of complex application scenarios.

To address the numerous challenges of achieving consistently high denoising quality in both high- and low-noise environments, adapting to various complex application scenarios with high robustness, and enhancing the model's resistance to adversarial attacks, we proposed a new image denoising model robust to adversarial attacks. The model is developed with a U-Net–based [23] generator in the GAN framework as baseline, the noise optimization algorithm (NOA) is adopted to the generator of the GAN during training, and a pyramid coordinate attention (PCA) module is further added to learn the global and local difference between the denoised and normal-dose images in image and gradient domains, thereby improving the model's overall robustness against attacks. With the proposed methods, the new model achieves superior performances on various tasks, especially when facing adversarial attacks, demonstrating its robustness.

The main contributions of the present paper include the following:

- We proposed a GAN-based denoising model with NOA during training, which directly inject the noises on the neuron and optimize their standard deviations in the (BP) process without involving additional network structures or computing resources. This method further improves the denoising capability of the network and enhances its robustness.
- We proposed the PCA network structure with a spatial pyramid feature fusion method, which enhances the capability of feature expression and improves the extraction of hidden information and enhances the accuracy of image restoration.
- Extensive experimental results on denoising datasets demonstrate that our new model, NOP-GAN, exhibits state-of-the-art accuracy and robustness across various denoising tasks.

2. Related Work

2.1. GAN-Based Denoising. The GAN [24], as a significant machine learning model, along with its variants, has been successfully applied in many fields, such as image synthesis [25, 26], image translation [27, 28], and image classification [29, 30]. A GAN consists of two parts: a generator and a discriminator. The generator learns the real data distribution and generates synthetic samples similar to real ones from random noise, while the discriminator attempts to classify input samples, trying to distinguish between real and generated samples. During training, the generator continuously optimizes to deceive the discriminator, and the discriminator simultaneously learns to more accurately differentiate between the samples. Through this alternating training process, a game is formed, ultimately leading the generator to produce results that are very close to real samples.

In recent years, several studies have applied GANs to image denoising, such as the CGAN-based method [31] and SWDGAN [32]. Furthermore, Huang et al. proposed a GAN based on a dual-domain U-Net (DU-GAN) [33]. This method employs two U-Net discriminator models to differentiate between global structures and local details, allowing it to learn the differences between denoised and standard images at both global and local levels. As a result, DU-GAN achieves excellent denoising performance. However, while GAN-based methods can restore noisy images and enhance visual quality, they often neglect the preservation of the image structure [31].

Unlike existing GAN-based denoising models, our approach modifies the GAN architecture by integrating a U-Net with a PCA mechanism and a NOA into the generator of the GAN. This enables our model to excel at extracting and fusing multiscale information, thereby enhancing denoising and data recovery capabilities, ultimately contributing to improved network robustness.

2.2. Noise Optimization. The noise optimization is developed based on the pathwise stochastic gradient estimation technique, which is also known as infinitesimal perturbation analysis (IPA) in the simulation literature [34]. The IPA and the likelihood ratio (LR) method are two classic unbiased stochastic gradient estimation techniques [35, 36]. Recent advances can be found in [37, 38]. Stochastic gradient estimation has been a central topic in the simulation optimization, and recently, a comprehensive review paper is written by DeepMind [39]. In Xiao et al.'s study [40], the noise optimization is developed by computing the pathwise stochastic gradient estimate with respect to the standard deviation of the Gaussian noise added to each neuron of the ANN. The gradient estimate with respect to noise levels is a byproduct of the BP algorithm, and thus negligible additional cost is involved. The noise optimization has shown to improve the accuracy, robustness, and efficiency on various tasks, including classification [41] and fault detection [43]. In this work, we adopted the new algorithm to train the generator of the GAN-based denoising model to improve its robustness.

2.3. Coordinate Attention (CA). Attention mechanisms [44, 45] and their derivatives have been proven beneficial in various computer vision tasks, such as image classification [46, 47] and image segmentation [48–50]. However, the computational cost of attention mechanisms significantly increases, especially for large-scale tasks, where the memory and computational demands of attention mechanisms are very high. Hou et al. [51] embedded positional information into channel attention (e.g., squeeze-and-excitation attention [47]), naming it CA. This approach avoids the high overhead associated with attention mechanisms while capturing cross-channel information, directional awareness, and position-sensitive information, allowing the model to more accurately locate and identify objects of interest. Unlike channel attention, which converts a feature tensor into a single-feature vector through 2D global pooling, CA divides the channel attention into two 1D feature encoding processes, aggregating features along two spatial directions separately. In this way, long-range dependencies can be captured along with one spatial direction, while precise positional information can be preserved along with the other spatial direction. The resulting feature maps are then encoded into a pair of direction-aware and position-sensitive attention maps, which can be complementarily applied to the input feature maps to enhance the representation of objects of interest. Moreover, CA performs better in downstream tasks such as object detection and semantic segmentation. Given the flexibility and powerful applicability of CA, we incorporated the CA mechanism into the generator of the GAN, alongside a feature pyramid structure, thereby innovatively applying CA to denoising tasks.

3. Methods

This paper proposes the NOP-GAN architecture, as shown in Figure 1. In the basic GAN architecture, we primarily modified the generator (upper part of Figure 1) to be a U-Net, incorporating a NOA and a PCA structure to enhance the generator's robustness. After training, the generator is used as a denoising network. To further improve the model's robustness, we added real noise from the SIDD dataset and randomly generated Gaussian noise from PyTorch to the original input image, creating a noisy image. The generator takes this noisy image as input and outputs a denoised image of the same size. The discriminator network outputs a real or fake value to determine the authenticity of the input feature map.

3.1. Network Architecture. The NOP-GAN is an innovative image denoising model that incorporates a U-Net structure within the GAN framework. Since the effectiveness of image denoising largely depends on the generator's capability, the generator is the core component of the entire model. Unlike traditional GANs, our model redesigns the generator (upper part of Figure 1) within the GAN structure by using U-Net as the overall architecture for the generator. This allows for encoding and decoding of images while preserving high-resolution details. In addition, the NOP-GAN integrates a NOA and PCA into the U-Net, enhancing the model's ability for feature extraction and denoising, and improving denoising robustness.

In the U-Net–based generator, the first half consists of the Mp layers (yellow in Figure 1), which refer to the max-pooling operations after convolution. These operations downsample the input feature maps, reducing the number of parameters, enhancing the model's generalization ability, and retaining important feature information. The Ellan layers (orange in Figure 1) are obtained by combining convolutional layers with different gradients. The noise optimization layers (green in Figure 1) apply NOA to the extracted image features to improve the stability of image restoration and ensure the robustness of the network. Low-level features contain a large amount of disordered, fine-grained image details, while high-level features contain a large amount of abstract, coarse-grained information. To fully utilize image features of different granularity and decode the features into images of the same size as the original, different modules undergo five downsampling steps (such as the "Mp" and "Ellan" modules) and five upsampling steps (such as convolution + ReLU operations). The U-Net captures global image features through downsampling (encoder) and restores image details through upsampling (decoder). In addition, the U-Net uses skip connections to transfer features from the encoder to the corresponding layers of the decoder, ensuring that detail information is not lost during downsampling. After upsampling, low-resolution and high-resolution feature maps are fused,

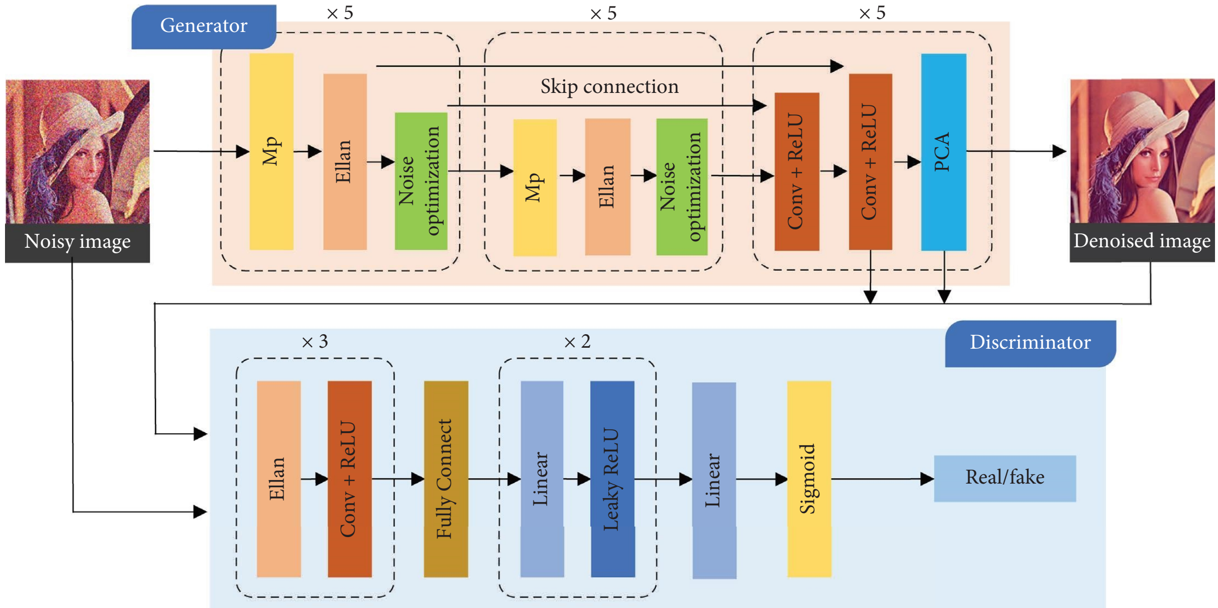


Figure 1: Total network structure of the NOP-GAN.

and the PCA module (blue in Figure 1) is used to enhance the fusion and representation of image features. This facilitates the recovery of image information and improves the denoising capability.

In the GAN of the NOP-GAN, the discriminator is used to determine the authenticity of the input feature maps. The discriminator network is based on CNNs. During training, the input consists of feature maps from different stages of the generator, which enhances the discriminator's loss expression and increases its influence on the generator, making it easier for the generator to perform image denoising.

3.2. NOA. An effective denoising neural network model needs to have robustness and some antidisturbance ability [52]. Models with antidisturbance ability can maintain accuracy and stability in diverse or corrupted datasets. Therefore, we adopted the optimization as shown in [40]. The optimization adds Gaussian noise on each neuron and optimizes both the noise distribution and model parameters during the training through a pathwise stochastic gradient estimate technique.

Let τ represent the number of layers in the neural network, m_t represent the number of neurons in the t layer, where $t \in 1, 2, \ldots, \tau$. We denoted the output of the t layer as $x^{(t)} = [x_1^{(t)} x_2^{(t)}, \ldots, x_{m_t}^{(t)}] \in R^{m_t}$, where $x^{(0)}$ is the input of the network. Assuming that there are N inputs in the network, denoted as $x^{(0)}(N)$, where $N = 1, 2, \ldots, n$. For the n-th input, the i-th output of the t-th layer is given in the following equations:

$$x_i^{(t+1)}(n) = \varphi\left(v_i^{(t)}\right), \tag{1}$$

$$v_i^{(t)} = \sum_{j=0}^{m_t} \theta_{i,j}^{(t)} x_j^{(t)}(n) + z_i^{(t)}(n), \tag{2}$$

where $x_j^{(t)}(n)$ represents the j-th input of the n-th data in the t-th layer, $\theta_{i,j}^{(t)}$ represents the weight of the i-th input in the t-th layer, $v_i^{(t)}$ represents the output of the i-th neuron in the t-th layer, φ is the activation function, and $z_i^{(t)}(n)$ represents the independent random noise added to the i-th neuron in the t-th layer for the n-th data. The bias term b is an additional learnable parameter in a neural network, which introduces a shift or constant value in the model. The bias term adjusts the activation threshold of the neurons by adding a fixed value, allowing the neural network to better adapt to the training data and improve its ability to fit the data. Figure 2 shows a visualization of the NOA.

Let L represent the loss function, and let $L(x^{(0)}(n), Y(n))$ represent the loss value of the n-th data $x^{(0)}(n)$ labeled as $Y(n)$. In our work, we attempted to optimize the size of the standard deviation $\sigma_i^{(t)}$ of the central normal random noise added to each neuron, such that $z_i^{(t)}(n) = \sigma_i^{(t)} \varepsilon_i^{(t)}(n)$, where $\varepsilon_i^{(t)}(n)$ is a standard normal random variable. We defined the residual of the n-th data passed backward through the i-th neuron in the t-th layer of the neural network, as shown in the following equation:

$$\delta_i^{(t)}(n) = \begin{cases} e_i^{(\tau)}(n)\varphi'\left(v_i^{(\tau-1)}(n)\right), & t = \tau, \\ \varphi'\left(v_i^{(t-1)}(n)\right)\left(\sum_{j=0}^{m_t} \theta_{i,j}^{(t)} \delta_j^{(t+1)}(n)\right), & t < \tau, \end{cases} \tag{3}$$

where $e_i^{(\tau)}(n)$ is defined as shown in the following equation:

$$e_i^{(\tau)}(n) = \left.\frac{\partial L(x, Y(n))}{\partial x_i}\right|_{x=x(\tau)(n)}. \tag{4}$$

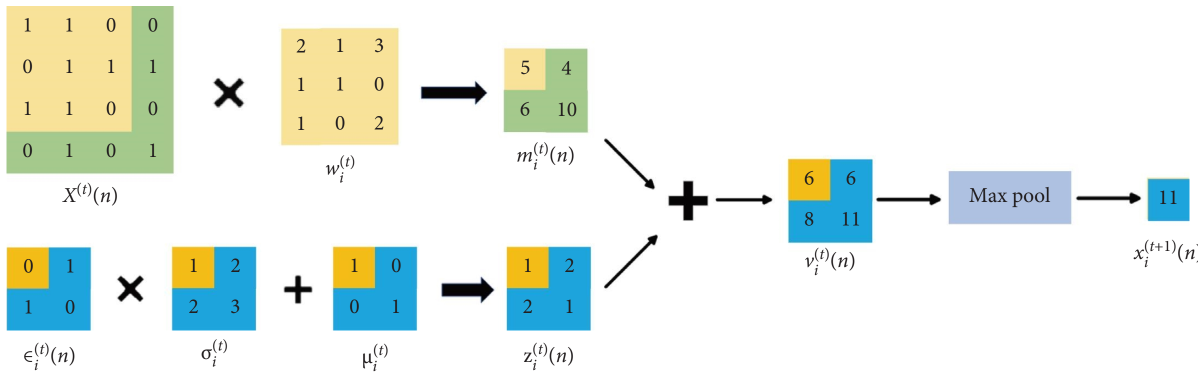


Figure 2: Visualization of the noise optimization algorithm.

BP essentially provides a stochastic gradient estimate of the path [53]. Derivatives of the loss function L with respect to all parameters $\theta_{i,j}^{(t)}\,(t = 1, 2, \ldots \tau - 1)$ can be seen in the following equation:

$$\frac{\partial L\left(x^{(\tau)}(n), Y(n)\right)}{\partial \theta_{i,j}^{(c)}} = \delta_j^{(t+1)}(n) x_j^{(t)}(n), \tag{5}$$

where $j \in \{0, 1, \ldots, m_t\}, i \in \{0, 1, \ldots m_{t+1}\}$.

The complete calculation procedure is shown in Algorithm 1, where the input is a set of training data P and the loss function L. The algorithm initializes the weights W and noise level N to random values and iteratively updates them to minimize the loss function. In each iteration, the algorithm computes the output using equations (1) and (2), calculates the loss, estimates the gradient of the loss with respect to the weights and noise level using equations (3) and (5), and then updates the weights and noise level. When the model converges, meaning that the parameters no longer change significantly, the algorithm terminates and returns the optimal weights W and noise level N.

This multiscale attention mechanism not only captures fine-grained image details but also integrates broader contextual information, thereby significantly improving the accuracy and robustness of denoising tasks.

In the PCA module, input features are extracted through three convolutional layers with different kernel sizes (k_1, k_3, and k_5) to capture multiscale information, which are then fused into a unified representation through feature concatenation and convolution. Meanwhile, the CA mechanism separates horizontal ($X_1 D$) and vertical ($Y_1 D$) feature distributions to generate channel and spatial attention weights, which are subsequently used to reweight the input features. This design effectively captures both fine-grained multiscale information and contextual semantics, significantly enhancing the model's ability to represent features in complex scenes.

The structure of the PCA is shown in Figure 3, given an input feature map $F \in R^{C*H*W}$ (F is with dimensions of $C*H*W$). For a pixel point (i, j) in channel C, its value is $x_C(i, j)$. Global pooling is divided into two steps (a pair of 1D pooling), where two pooling kernels $(H, 1)$ and $(1, W)$ are applied along two different directions X and Y of the feature map. The output of the C-th channel at X position can be formulated in the following equation:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i). \tag{6}$$

Similarly, the output of the c-th channel at Y position can be written as the following equation:

$$z_c^w(h) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w). \tag{7}$$

Two embedded information feature maps $X_1 D$ and $Y_1 D$ are obtained, and the obtained two embedded feature maps are concatenated along the spatial dimension and activated after a shared 1×1 convolutional transformation function F_1, which is shown in the following equation:

$$f = \delta\left(F_1\left(\left[z^h, z^w\right]\right)\right) \in R^{C/(r \times (H+W))}, \tag{8}$$

where $[\cdot, \cdot]$ represents the concatenation of feature maps along the spatial dimension and δ represents a nonlinear activation function. f is a feature map that captures spatial information in both the horizontal and vertical directions and r is the reduction ratio used to control the block size.

The split operation is then performed along the spatial dimension to obtain two separate feature maps $f^h \in R^{C/r \times H}$ and $f^w \in R^{C/r \times W}$. Another two 1×1 convolutional transformations F_h and F_w are utilized to separately transform f^h and f^w to tensors with the same channel number to the input; it can be summarized in the following equation:

$$\begin{aligned} g^h &= \sigma\left(F_h\left(f^h\right)\right), \\ g^w &= \sigma\left(F_w\left(f^w\right)\right), \end{aligned} \tag{9}$$

where σ is the sigmoid function. The outputs g^h and g^w are then expanded and used as attention weights, respectively. Finally, the value of pixels on the output feature map can be written as the following equation:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j), \tag{10}$$

where $x_c(i, j)$ represents the value of a pixel point on the input feature map. $g_c^h(i)$ denotes the weight of the pixel point (i, j) in the Y direction, while $g_c^w(j)$ represents the

Input: Training data $P = \left\{x^{(0)}(n), Y_{(n)}\right\}_{n=1}^{N}$, loss function L
Output: Optimal weights W and noise level N
Initialize the weights W and noise level N to random values
Repeat:
a. Calculate the output $x^{(\tau)}(n)$ using equations (11) and (12)
b. Calculate the loss $L(x^{(\tau)}(n), Y(n))$
c. Estimate the gradient of the loss with respect to the weights and noise level using equations (13) and (2)
d. Update the weights and noise level
Until Return the optimal weights W and noise level N

Algorithm 1: NOA.

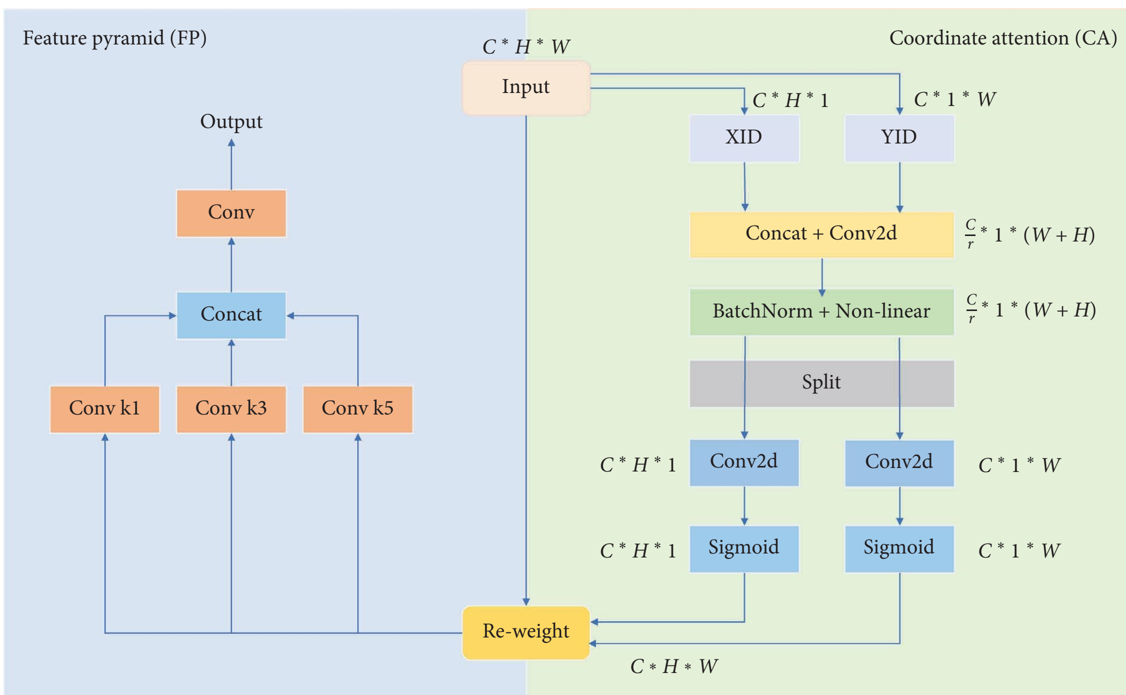


Figure 3: The structure of PCA.

weight in the X direction. Finally, these weights are multiplied to obtain the output y_c. All the individual output pixel points combine together to form the complete reweight feature map "Reweight".

After obtaining an attention weight map of the same size as the input feature map, CA applies this attention weight map to the input feature map. By doing so, the model can adjust the importance of different spatial positions in the feature map based on the assigned attention weights. This process yields the output of CA. The output feature map is fused using a pyramid feature fusion layer which uses convolutional kernels of different scales "1, 3, and 5" to increase feature diversity. Finally, the features obtained from the different convolutional kernels are concatenated, improving the expressiveness of the features and enhancing the network's ability to extract features.

By combining pyramid feature fusion and CA in our NOP-GAN, we can fully utilize the contextual information from multiscale features and spatial attention on key positions, thereby improving the denoising performance. The NOP-GAN structure can achieve a remarkable performance in image denoising tasks, effectively removing noise of various levels while preserving image clarity and details and also enhancing the model's robustness.

3.3. Loss Function. The loss function is an important part of the GAN training, which defines the optimization objectives and evaluation criteria for the generator and discriminator. The loss function for the GAN is to measure the difference between the denoised images generated by the generator and the real images, as well as the classification accuracy of the discriminator for the generated images and the real images. The goal of the loss function is to make the generator generate denoised images that are closer to the real images, while reducing the discriminator's ability to distinguish between the generated images and the real images.

3.3.1. Total Loss. The loss function is made up of two parts: the pixelwise loss L_{MSE} and the adversarial loss L_{adv}. Mean square error (MSE) loss is used to measure the pixel-level difference between the generated image and the real image, and adversarial loss is used to measure the difference in the distribution between the generated image and the real image. The MSE loss and the adversarial loss are added together to form a total loss function that is used to optimize the parameters of the generator. The total loss is shown in the following equation:

$$L_{\mathrm{total}} = \alpha L_{\mathrm{MSE}} + \beta L_{\mathrm{adv}}, \tag{11}$$

where α and β are the two hyperparameters that are used to control the weights of the L_{MSE} and the L_{adv}. The ratio of L_{MSE} to countermeasure loss affects the quality and effectiveness of the generated image. If the L_{MSE} is too large, the resulting image may be too smooth and blurry, losing detail and texture. If the proportion of L_{adv} is too large, the resulting image may be too sharp and noisy, with artifacts and distortions.

3.3.2. Pixelwise Loss. Pixelwise loss is a method for calculating the interpixel loss of the predicted image and the target image, which can be achieved with L_{MSE}. The objective of this loss function is to optimize the generator's performance, making it able to generate images that are more similar to the target image. By minimizing L_{MSE}, we can adjust the generator's stability and improve its image quality. The L_{MSE} is shown in the following equation:

$$L_{\mathrm{MSE}} = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - f\left(x_i\right)\right)^2, \tag{12}$$

where y_i denotes the pixel values in the original image, $f\left(x_i\right)$ denotes the corresponding pixel value in the denoised image, and m is the total number of pixels in the image.

3.3.3. Adversarial Loss. L_{adv} is used to train the discriminator, making it able to effectively distinguish real images and generated images. The objective of this loss function is to maximize the probability of real images and minimize the probability of generated images. By optimizing L_{adv}, we can improve the discriminator's classification accuracy and make it better at recognizing generated images. The adversarial loss is a key element of GANs, and its main role is to evaluate whether the images produced by the generator are sufficient to confuse the discriminator. It is shown in the following equation:

$$\min_G \max_D L_{\mathrm{adv}} = E_{x\sim p_{\mathrm{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log\left(1 - D(G(z))\right)]. \tag{13}$$

Thet term $E_{x\sim p_{\mathrm{data}}(x)}[\log D(x)]$ represents the discriminator's prediction on real data. x is the data sampled from the real data distribution; $p_{\mathrm{data}} \cdot D(x)$ is the prediction of the discriminator on these real data. G and D below min and max represent generator and discriminator, respectively, and min and max represent the optimization goals of generators and discriminators, respectively. The goal of this term is to maximize the output of the discriminator on real data, that is, let the discriminator identify real data as real as possible.

The term $E_{z\sim p_z(z)}[\log\ (1 - D(G(z)))$ represents the prediction of the discriminator on the generated data. z is the noise sampled from some noise distribution, $p_z \cdot G(z)$ is the fake data (the denoised picture) generated by the generator using this noise, and $D(G(z))$ is the prediction of the discriminator for these fake data. This term is designed to maximize the probability that the discriminator classifies fake data as false. In other words, it encourages the discriminator to identify fake data as accurately as possible.

In this iterative process, the parameters of both the generator and discriminator are updated through the optimization of their respective loss functions. This mutual optimization establishes a dynamic equilibrium, where the generator progressively improves its ability to produce realistic images, while the discriminator faces increasing challenges in differentiating between generated and real images.

4. Experiments

In our study, we conducted extensive evaluations of the proposed model on three image datasets. We compared our proposed model with seven advanced models, including BM3D [54], DnCNN [10], TNRD [55], FFDNet [11], RDDCNN [21], SCUNet [56], and MambaIR [57]. Notably, we also applied fast gradient sign method (FGSM) and projected gradient descent (PGD) attacks to the original images to demonstrate the robustness of our denoising method. The experiment was conducted on a hardware platform equipped with a Tesla P100 GPU featuring 16 GB of memory. The software environment was set to Ubuntu 18.04, CUDA 10.2, and Python 3.6. The deep learning framework used was PyTorch 1.8.

The next content is organized as follows. In Section 4.1, we introduced the datasets and provided the details of the experimental setup. Section 4.2 presents the experimental results and compares our model with mainstream denoising models under various Gaussian noise levels. In Section 4.3, we compared the runtime of our model with that of other models. Section 4.4 presents the ablation studies conducted to demonstrate the usefulness of each component, including the NOA and the PCA module. Finally, in Section 4.5, we applied FGSM and PGD attacks to the original images to demonstrate the robustness of our denoising method.

4.1. Datasets and Implementation Details

Description of Datasets: To evaluate the effectiveness of image denoising, we selected four widely used and well-recognized image datasets. Each dataset has unique characteristics and presents different challenges for denoising methods. The specific details of these datasets are as follows:

Dataset 1: The Berkeley segmentation dataset 500 (BSD500) [58] is a dataset consisting of 500 natural images, which have been manually annotated for image segmentation. This dataset is commonly used by researchers in the fields of computer vision and image processing for evaluation purposes. In our study, we used both grayscale and color images from the BSD500 for training and testing, cropping each image to a size of $256 \times 256 \times 3$. Subsequently, we introduced Gaussian noise levels of 25, 50, and 75, as well as real noise levels of 1, 2, and 3. This process results in various noisy-clean pairs, each exhibiting different degrees and types of noise.

Dataset 2: The smartphone image denoising dataset (SIDD) [59] is an image denoising dataset captured by five representative smartphone cameras under different lighting conditions, containing 30,000 noisy images from 10 different scenes, with ground truth images provided alongside the noisy images. We selected the SIDD-medium dataset for testing, which consists of 320 image pairs. We converted the images in the SIDD dataset to grayscale and conducted tests. The dataset images are also cropped to a uniform size of 180×180. In addition, Gaussian noise [60] added at standard deviations of $\sigma = 25$ to $\sigma = 75$, in increments of five, are added to all images to construct the corresponding noisy-clean pairs.

Dataset 3: The PolyU real-world noisy images dataset (PolyU) [61] consists of 100 real noisy images with dimensions of 2784×1856, obtained by five different cameras: a Nikon D800, Canon 5D Mark II, Sony A7 II, Canon 80D, and Canon 600D. Similar to Dataset 2, we converted the images to grayscale for testing and crop them to a consistent size of 180×180. Gaussian noise added at standard deviations of $\sigma = 25$ to $\sigma = 75$, in increments of five, are applied to the images to create the corresponding noisy-clean pairs.

Dataset 4: The Set12 dataset [62] comprises 12 classic images, including Lena and Cameraman. Among them, Images 1–7 have a resolution of 256×256, while Images 8–12 have a resolution of 512×512, making the dataset widely used for evaluating image denoising methods. We applied our proposed method to color image denoising on this dataset and achieved notable results under Gaussian noise levels of 25 and 50, with enhanced image clarity and effectively preserved color fidelity.

Implementation Details: During training, the training set contains 1200 to 1400 images. The batch size was set to 128, and the Adam algorithm was used to update the gradients. The initial learning rate was set to 0.001 and progressively reduced by a factor of 10 after a predefined number of epochs. In addition, the momentum parameter was configured to 0.9.

4.2. Denoising Results. Image denoising refers to the removal of noise from an image, distracting information that does not belong to the image itself. The evaluation methods of image denoising results can be divided into two types: objective evaluation and subjective evaluation.

In all tables, an upward arrow (↑) indicates that a higher value of the corresponding metric signifies better performance, while a downward arrow (↓) denotes that a lower value is preferable. Additionally, the best result for each metric is highlighted in bold.

4.2.1. Objective Evaluation. Objective evaluation refers to the quantitative assessment of image quality using mathematical models or statistical methods. We employed four metrics to comprehensively evaluate our proposed denoising method against other existing image denoising method: peak signal-to-noise ratio (PSNR), Structural Similarity Index (SSIM) [63], learned perceptual image patch similarity (LPIPS) [64], and signal-to-noise ratio (SNR) [65, 66]. The advantage of the objective evaluation is its objectivity and quantifiability.

Dataset Settings: We conducted a comprehensive comparison of our NOP-GAN with BM3D, DnCNN, TNRD, FFDNet, RDDCNN, SCUNet, and MambaIR using the BSD500, SIDD, and PolyU datasets under different conditions. These conditions include the following: for the BSD500 dataset, we used both color and grayscale images for training and testing, adding Gaussian noise with standard deviations of 25, 50, and 75, as well as real noise levels of 1, 2, and 3; for the SIDD and PolyU datasets, we used only grayscale images for training and testing, adding Gaussian noise with standard deviations ranging from 25 to 75, in increments of five.

4.2.1.1. Analysis on BSD500 Dataset. As shown in Table 1, the PSNR values of the other methods consistently lower than our proposed NOP-GAN. Notably, the advantages of our model become more pronounced as noise levels increase. Specifically, in a high noise environment with a standard deviation (σ) of 75, the NOP-GAN exhibits an improvement of approximately 4 dB over BM3D, approximately 3 dB over DnCNN, approximately 2 dB over FFDNet, and approximately 1 dB over RDDCNN. These substantial gains in the PSNR across varied noise environments underscore the evident superiority of our model in image denoising.

Furthermore, under diverse noise conditions, the SSIM values of our proposed network consistently out perform those of the previously mentioned methods. Particularly, in a real noise environment with $L = 2$, our proposed model demonstrates improvements of approximately 0.1 over BM3D, approximately 0.09 over DnCNN, approximately 0.16 over TNRD, approximately 0.08 over FFDNet, approximately 0.02 over RDDCNN, and approximately 0.05 over SCUNet. The stability of SSIM values across different

Table 1: Image denoising results on grayscale BSD500 dataset.

Methods	Gaussian Noise						Real Noise					
	$\sigma = 25$		$\sigma = 50$		$\sigma = 75$		$L = 1$		$L = 2$		$L = 3$	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Noisy images	20.42	0.422	14.88	0.297	12.63	0.185	18.65	0.615	17.52	0.600	16.18	0.588
BM3D [54]	28.57	0.845	25.62	0.766	22.32	0.713	29.32	0.863	26.93	0.801	23.63	0.742
UM-NLF	28.64	0.803	25.62	0.676	/	/	28.14	0.848	25.73	0.703	/	/
DnCNN [10]	29.23	0.859	26.23	0.787	23.21	0.738	29.74	0.872	27.04	0.811	24.72	0.754
TNRD [55]	28.92	0.815	25.97	0.702	/	/	28.97	0.852	26.86	0.741	/	/
FFDNet [11]	29.44	0.858	27.32	0.784	24.43	0.743	30.51	0.883	28.93	0.824	25.42	0.765
RDDCNN [21]	29.27	0.863	27.39	0.783	26.22	0.779	31.60	0.902	30.19	0.881	27.67	0.864
SCUNet [56]	29.90	0.849	26.86	0.757	/	/	31.50	0.826	30.41	0.852	28.19	0.879
MambaIR [57]	30.12	0.824	28.29	0.866	/	/	/	/	/	/	/	/
NOP-GAN	**30.14**	**0.903**	**28.53**	**0.872**	**26.36**	**0.833**	**31.54**	**0.933**	**30.43**	**0.908**	**28.29**	**0.883**

noise environments highlights the robust denoising capability of our proposed model.

4.2.1.2. Analysis on SIDD and PolyU Dataset. The line charts of our various experimental results are shown in Figure 4. It should be noted that in the line chart comparing the NOP-GAN with six other models BM3D, DnCNN, TNRD, FFDNet, RDDCNN, and SCUNet, the denoising tests for DnCNN, FFDNet, RDDCNN, and SCUNet were conducted using pretrained models provided by the respective authors of these models. Since the SCUNet does not provide a pretrained model for $\sigma > 50$, our denoising tests for SCUNet were only conducted up to $\sigma = 55$. Our method outperforms others across all metrics including PSNR, SSIM, LPIPS, and SNR on the SIDD dataset and the PolyU dataset. It demonstrates exceptional denoising capabilities, particularly in high-noise image scenarios, significantly surpassing other methods in preserving image details and structures while minimizing degradation in image quality. Furthermore, our method maintains stable performance across different noise levels, showcasing robust resilience.

PSNR is a key metric for evaluating image quality in the fields of image processing and denoising. Our denoising method demonstrated strong adaptability and robustness across different datasets, especially under high-noise conditions where the noise standard deviation (σ) is greater than 50. In these cases, the PSNR values were significantly higher than those of other methods. This indicates that the denoised images are closer in overall quality to the original images, effectively removing noise while better preserving the details of the original image. By comparing the two datasets, our method excels in real noise environments (such as PolyU) while maintaining superior image quality in realistic noise environments (such as SIDD).This suggests that our denoising method has broad application potential, providing high-quality denoising results in a variety of complex environments.

SSIM is an important metric for measuring image quality, taking into account various aspects such as luminance, contrast, and structure. Our method consistently maintained high SSIM values across the entire noise range, particularly excelling under high-noise conditions ($\sigma > 55$), significantly outperforming other existing methods. Whether on the PolyU dataset or the SIDD dataset, the stable performance of SSIM demonstrates the strong robustness of our method, maintaining high image quality across various noise levels.

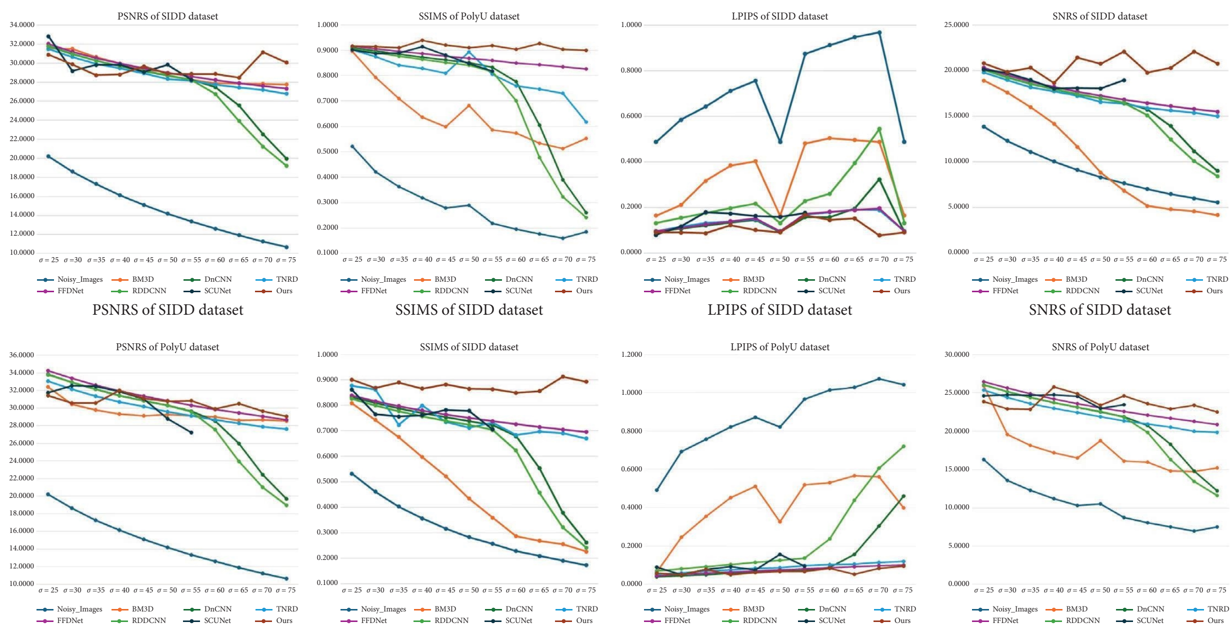
LPIPS is a metric based on deep learning models used to estimate perceptual differences between images. Our method performed exceptionally well across different datasets and noise conditions, with LPIPS values remaining the lowest at most noise levels (our results consistently stayed below 0.1, with the best reaching 0.05), particularly excelling in high-noise situations. This indicates that our method effectively preserves perceptual quality, closely aligning with human visual perception, especially when handling the real-world noise in the PolyU dataset.

SNR is used to measure the ratio of signal strength to noise level. Our method consistently achieved the highest SNR values across both datasets, particularly under high-noise conditions, demonstrating remarkable stability. This indicates that our method effectively maintains the signal strength of the image at various noise levels, showcasing strong noise suppression capabilities. Although PSNR, SSIM, LPIPS, and other metrics have been used to evaluate the performance of denoising methods, SNR still holds unique importance. SNR clearly illustrates how well a method preserves the signal and suppresses noise during the denoising process, providing a more intuitive reflection of the method's effectiveness in practical applications. This further highlights the outstanding performance of our method in directly handling signal noise in images.

Through these comprehensive evaluation results, our method not only performs excellently under various noise conditions but also significantly surpasses other existing denoising methods in terms of perceptual quality, detail preservation, and noise suppression. This demonstrates its strong potential for practical applications. The experimental results also show that our method performs better on several benchmark datasets in terms of both quantitative and qualitative metrics.

4.2.2. Subjective Evaluation. Subjective evaluation refers to judging the quality of images based on human visual perception, mainly considering factors such as clarity, detail preservation, and edge sharpness. The advantage of subjective evaluation is that it conforms to human visual habits. We performed the visualization of denoising results on grayscale images and color images separately.

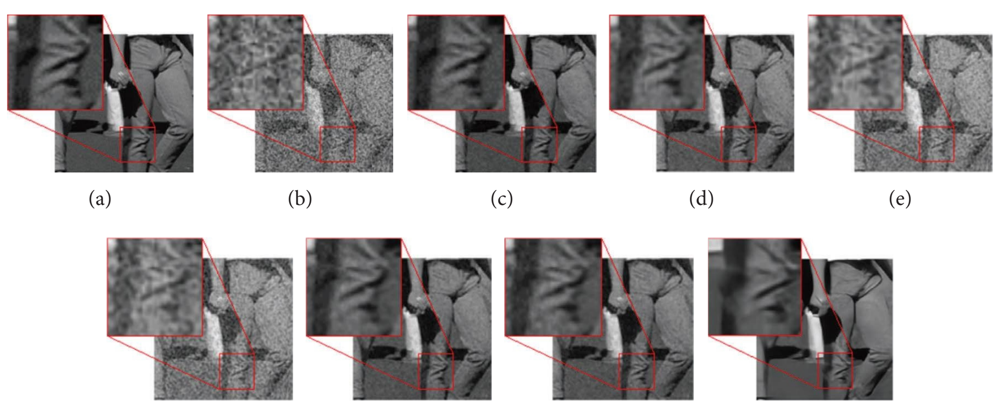
4.2.2.1. Visualization Results of Grayscale Image Denoising. To evaluate the visual perceptual differences, we selected an image from the BSD500 grayscale image dataset, conducted an experiment at a Gaussian noise level of 25, and included the corresponding PSNR metrics. We zoomed in on parts of the image to show the details more clearly, such as edges, textures, and colors. These details are essential for evaluating the image-denoising performance. Also, we can compare the advantages and disadvantages of different denoising methods more intuitively. Figure 5(a) represents the original image with no noise; Figure 5(b) is the image with added noise; Figures 5(d)–5(i) are the results after denoising with the BM3D, UM-NLF, TNRD, DnCNN, FFDNet, and RDDCNN methods, respectively. The traditional denoising method BM3D has achieved good visual results. However, it tends to produce blurring in some edge features. The UM-NLF denoising method is not thorough enough and still leaves some noise residue. On the other hand, supervised neural networks such as TNRD, DnCNN, and FFDNet can effectively remove noise while preserving texture information. However, in some areas of the image, the edges may appear smoother and slightly blurred. The RDDCNN



PSNRS of SIDD dataset SSIMS of SIDD dataset LPIPS of SIDD dataset SNRS of SIDD dataset

PSNRS of PolyU dataset SSIMS of PolyU dataset LPIPS of PolyU dataset SNRS of PolyU dataset

Figure 4: Results on SIDD and PolyU dataset.



(a) (b) (c) (d) (e)

(f) (g) (h) (i)

Figure 5: Denoising results of different methods on one image from BSD500 when $\sigma = 25$. The PSNR values (in dB) for each method are listed as follows: (a) original image; (b) noisy image/18.82 dB; (c) ours/30.77 dB; (d) BM3D/26.97 dB; (e) UM-NLF/19.11 dB; (f) TNRD/22.10 dB; (g) DnCNN/28.90 dB; (h) FFDNet/28.27 dB; (i) RDDCNN/30.07 dB.

method produces overly smoothed results at object boundaries, leading to insufficient restoration of the image's finer structural details. Figure 5(c) is the result of denoising using the NOP-GAN proposed in this paper, which is more visually intuitive. It can restore the complete structure of the image, retain the texture details of the original undistorted image, and enhance the visibility of clear lines and textures in the image. The denoised image is nearly indistinguishable from the original high-definition image in terms of clarity.

4.2.2.2. Visualization Results of Color Image Denoising. In this section, we applied our proposed method to color image denoising.

The visual results of Set12 dataset on Gaussian noise level of 25 and 50 are shown in Figure 6. The denoised color images exhibit enhanced clarity, with sharper contours and fine details and textures. Our method demonstrates an ability to maintain color fidelity, ensuring that the original color characteristics are accurately retained. The images display a natural appearance, free from artificial artifacts or disturbances.

4.3. Running Time Comparison. As shown in Table 2, we compared the average running time of the NOP-GAN with other denoising methods, including BM3D, UM-NLF, DnCNN, TNRD, and FFDNet.

Traditional denoising methods, such as BM3D and UM-NLF, were significantly slower than deep learning–based denoising methods [67]. Our proposed model had the shortest running time compared to other methods. This is because the U-Net architecture was utilized to incorporate cross-layer connections, which ensured the extraction of

Figure 6: Denoising results of color Gaussian noisy image from Set12 dataset.

Table 2: Average run time for different methods (s).

Methods	Gaussian noise		Real noise	
	$\sigma = 25$	$\sigma = 50$	$L = 1$	$L = 2$
BM3D	0.17	0.23	0.16	0.21
UM-NLF	1.23	1.54	1.15	1.43
DnCNN	0.018	0.022	0.018	0.021
TNRD	0.019	0.024	0.018	0.022
FFDNet	0.023	0.030	0.021	0.027
NOP-GAN	0.016	0.018	0.014	0.017

high-level features while also integrating low-level feature information. As a result, the efficiency of the model was improved.

4.4. Ablation Experiments. To further verify the denoising capability of the NOP-GAN on images, we conducted ablation experiments. These experiments involved removing the NOA and the PCA modules from the network to test their performance. The performance of the modified algorithm was then compared to the complete NOP-GAN model. The experiments were conducted on the grayscale BSD500 dataset, with Gaussian noise added at standard deviations of 25, 50, and 75, as well as real noise levels of 1, 2, and 3. The results were compared using PSNR and SSIM metrics.

As shown in Table 3, when the NOA was removed from the generator (w/o NOA) and only the PCA module was used, the PSNR increased by 1.7 dB under the Gaussian noise environment with standard deviation $\sigma = 25$. In addition, the SSIM increased by 0.01. Under the real noise condition of $L = 2$, the PSNR was increased by 0.68 dB and the SSIM increased by 0.048. It indicates that the PCA can effectively enhance the model's capability for image restoration. When the PCA module was removed from the generator (w/o PCA) and only the NOA was used, the PSNR increased by 2.93 dB under the Gaussian noise environment with a standard deviation $\sigma = 75$, and the SSIM increased by 0.091. Under the real noise condition of $L = 3$, the PSNR increased by 2.45 dB and the SSIM increased by 0.052. It indicates that the NOA can effectively remove noise from the image. When both the NOA and PCA modules were used, the model's denoising ability on the noisy image reached its optimal performance. Under the Gaussian noise environment with a standard deviation of $\sigma = 50$, the PSNR increased by 4.08 dB and the SSIM increased by 0.05. Under the real noise condition of $L = 3$, the PSNR increased by 4.68 dB and the SSIM increased by 0.076. In addition, as shown in Table 4, to further verify the role of the PCA module, we conducted extended ablation experiments based on the SIDD dataset to evaluate the impact of removing different structures within the PCA module on denoising performance. The ablation experiments were conducted at different noise levels ($\sigma = 25$, 50, and 75). The experiments compared the effects of removing the CA structure (w/o CA) on the right side of the PCA module (Figure 3) and removing the feature pyramid structure (w/o FP) on the left side of the PCA module on the model's performance. These results were compared with the

Table 3: Ablation results.

Modules	Gaussian noise						Real noise					
	$\sigma = 25$		$\sigma = 50$		$\sigma = 75$		$L = 1$		$L = 2$		$L = 3$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
w/o NOA + PCA	27.16	0.854	24.45	0.822	21.12	0.724	28.34	0.892	27.44	0.833	23.61	0.807
w/o PCA	29.32	0.872	26.63	0.852	24.05	0.815	29.92	0.914	28.96	0.889	26.06	0.859
w/o NOA	28.86	0.864	24.96	0.841	22.85	0.807	29.54	0.907	28.12	0.881	24.99	0.842
NOP-GAN	30.14	0.903	28.53	0.872	26.36	0.833	31.54	0.933	30.43	0.908	28.29	0.883

Table 4: Additional ablation experiments for the PCA module.

Modules	Gaussian noise											
	$\sigma = 25$				$\sigma = 50$				$\sigma = 75$			
	PSNR ↑	SSIM ↑	LPIPS ↓	SNR ↑	PSNR ↑	SSIM ↑	LPIPS ↓	SNR ↑	PSNR ↑	SSIM ↑	LPIPS ↓	SNR ↑
w/o CA	30.5363	0.8702	0.0973	20.6901	28.8545	0.8482	0.1249	19.3814	29.6951	0.8709	0.0998	20.6527
w/o FP	30.7162	**0.9040**	**0.0712**	**22.0628**	27.8954	0.8273	0.0950	20.6287	27.5994	0.8605	0.1442	18.8250
NOP-GAN	**30.8903**	0.9006	0.0914	20.7859	**28.9106**	**0.8650**	**0.0914**	**20.7502**	**30.0622**	**0.8926**	**0.0914**	**20.7665**

performance of the complete model, using PSNR, SSIM, LPIPS, and SNR metrics for evaluation.

Specifically, after removing the CA module, the PSNR, SSIM, and SNR metrics of the model significantly decreased across all noise levels. Notably, at a noise level of $\sigma = 75$, the PSNR dropped from 30.0622 to 29.6951 dB, SSIM decreased from 0.8926 to 0.8709, and the LPIPS metric increased from 0.0914 to 0.0998, indicating a deterioration in perceptual quality. After removing the feature pyramid module, the model could still maintain a relatively good SSIM value (0.9040) at a low noise level ($\sigma = 25$), but PSNR and SNR metrics worsened across all noise levels. At a noise level of $\sigma = 75$, the PSNR decreased from 30.0622 to 27.5994 dB, demonstrating that the feature pyramid module significantly impacts the denoising performance for high-noise images. The LPIPS metric increased significantly at high-noise levels, indicating a noticeable decline in perceptual quality after removing the feature pyramid module.

The complete model exhibited the optimal denoising performance under high-noise levels. Metrics such as PSNR, SSIM, and SNR achieved the highest values, while LPIPS reached the lowest values, indicating a significant advantage of this model in preserving image details and overall denoising effectiveness. For instance, at a noise level of $\sigma = 75$, the complete model achieved a PSNR of 30.0622 dB and an SSIM of 0.8926, both superior to the models with any structure removed. Overall, both the CA module and the feature pyramid module significantly contribute to the performance of the denoising model. Removing any module leads to a performance decline, whereas the complete model consistently shows the optimal denoising performance across all metrics.

4.5. Attack Experiments. The purpose of the attack experiment is to use adversarial sample generation methods to attack the deep neural model and then observe the results and evaluation metrics to assess its robustness and security. FGSM attack and PGD attack are two standard adversarial attack methods for deep learning models. Their goal is to trick the model by adding tiny perturbations to the input data, leading to a misclassification of the model or an inaccurate output. Firstly, such tests allow for a thorough evaluation of the model's robustness in the face of various disturbances, particularly in real-world scenarios with image noise or interference. These assessments not only contribute to refining the model's design and implementation and enhancing its performance under noise or adversarial inputs but also bolstering the model's security and credibility in practical applications.

The NOP-GAN is designed with a NOA module to enhance the model's robustness against attacks. The noise and weight parameters are jointly optimized during training, enabling the model to adapt to various levels of noise perturbation and thereby improving robustness. A residual-based BP method is employed to update the noise level layer by layer, suppressing the impact of adversarial perturbations on the output. Ultimately, the iterative optimization process enhances the model's antidisturbance performance and stability on complex datasets, making it more reliable in practical applications. In the following sections, we introduced and experimentally validated the attack resistance of the NOP-GAN.

FGSM Attacks: The FGSM is an algorithm used to generate adversarial samples based on gradients. Its objective is to maximize the loss function in order to obtain adversarial samples. The equation for the FGSM algorithm is shown as follows:

$$x' = x + \varepsilon \cdot \mathrm{sign}\left(\nabla_x J(\theta, x, y)\right), \tag{14}$$

where x represents the input, y represents the targets of x, θ, x, θ represents the parameters of the model, $J(\cdot)$ represents the loss function, ∇_x represents the gradient of x, and $\mathrm{sign}(\cdot)$ is the sign function, which ensures that the attack direction is consistent with the gradient direction. The parameter ϵ denotes the level of perturbation. This ensures that the generated images cannot be easily recognized by human eyes. By adding the calculated gradient direction into the input image, the resulting modified image will have a higher loss value when passed through the network than the original image, which is used to evaluate the robustness of the network against attacks.

Figures 7 and 8 show the results of comparing the SSIM and PSNR values of the original image with the generated image with added Gaussian noise under different degrees of attacks. Under FGSM attacks, increasing perturbation levels slightly reduce SSIM and PSNR differences, but the impact remains minimal. This is sufficient to demonstrate that our algorithm can improve the network's resilience against attacks, demonstrating the superiority of our algorithm, and making the model robust.

PGD Attacks: The PGD attack is an iterative attack that can be considered a variation of FGSM. It can also be called K-FGSM (K represents the number of iterations). The general idea is that the FGSM only does one iteration and takes one giant step, while PGD does multiple iterations and takes one small step each time. Each iteration will limit the perturbation to a certain range, and the PGD algorithm can be expressed as follows:

$$x_{t+1} = \prod_{x+s} \left(x_t + \alpha \cdot \text{sign}\ \left(\nabla_x J\left(x_t, y\right)\right)\right), \tag{15}$$

where x_t is the input, y is the label of x_t, $J(\cdot)$, x_t, $J(\cdot)$ is the loss function, ∇_x is the gradient of x_t, and sign$(\cdot)$ is the sign function, ensuring that the attack changes in the same direction as the gradient. α is the step size for each attack iteration. We used 10-step PGD adversarial training with a step size of 0.01 to validate the stability of the model under different levels of attacks. The results are shown in Table 5.

Although multiple rounds of PGD iterative attacks were conducted, the SSIM and PSNR values of the generated image remained within a controllable range under different noise conditions. The PSNR value dropped by about 2.6 dB, and the SSIM value decreased by about 0.02 when the perturbation was $\epsilon = 0.05$, and the Gaussian noise was $\sigma = 25$. Similarly, the PSNR value decreased by about 2 dB, and the SSIM value decreased by about 0.03 when the perturbation was $\epsilon = 0.05$ and the real noise was $L = 3$. This shows that our NOA has a strong antidisturbance ability, demonstrating its practicality.

Subsequently, a comparative experiment was conducted without applying the NOA to quantitatively evaluate its impact on improving the network defense performance.

By comparing Tables 5 and 6, it is evident that the network's defense capability against PGD attacks significantly diminishes in the absence of the NOA. Specifically, when the attack level is set to $\epsilon = 0.05$ and the actual noise level is $L = 3$, the degradation in PSNR and SSIM values is reduced to half and one-tenth, respectively, compared to the impact observed without the optimization algorithm.

In addition, we conducted comparative experiments on the attack resistance of the proposed model, NOP-GAN, against several mainstream models, including DnCNN, TNRD, FFDNet, RDDCNN, and SCUNet. Specifically, we applied PGSM and PGD attacks with an intensity of $\epsilon = 0.05$ to each model and comprehensively evaluated their robustness at noise levels $\sigma = 25$, 50, and 75. The evaluation metrics included PSNR, SSIM, LPIPS, SNR, and the comprehensive resistance rate (CRR, calculated as shown in equation (19), which reflects the model's resilience against attacks). As shown in Tables 7 and 8, our model demonstrates strong robustness under various attack scenarios, with minimal degradation in PSNR, SSIM, LPIPS, and SNR, particularly outperforming other models in high-noise conditions. In terms of the resistance rate (RR), the NOP-GAN consistently achieves the highest levels across different noise and attack conditions, surpassing the next-best model, DnCNN, by 3%–5%. The experimental results

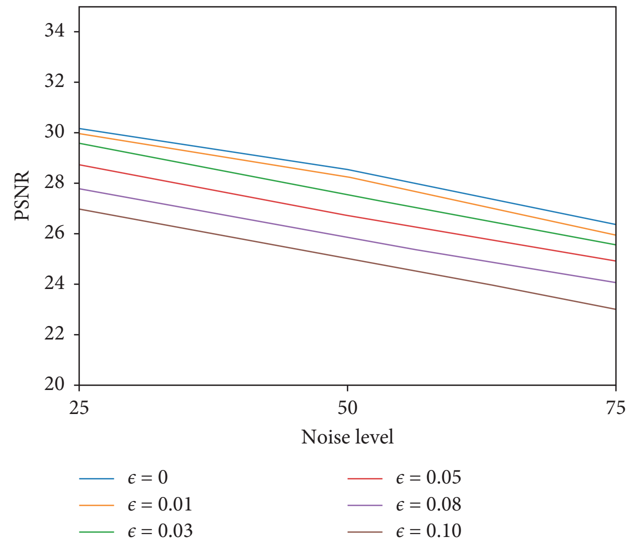


Figure 7: Comparison of PSNR values under different perturbations of FGSM attacks.

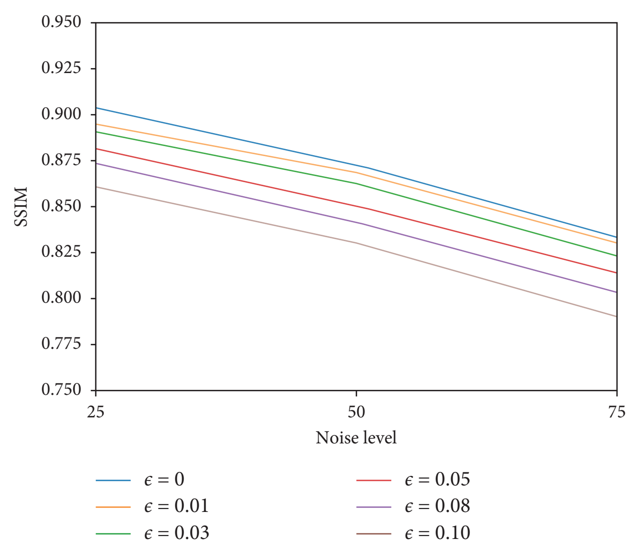


Figure 8: Comparison of SSIM values under different perturbations of FGSM attacks.

Table 5: Comparison of PSNR and SSIM values under different degree PGD attacks.

Degree of perturbation	Gaussian noise						Real noise					
	$\sigma = 25$		$\sigma = 50$		$\sigma = 75$		$L = 1$		$L = 2$		$L = 3$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$\epsilon = 0$	30.14	0.903	28.53	0.872	26.36	0.833	31.54	0.933	30.43	0.908	28.29	0.883
$\epsilon = 0.01$	29.07	0.898	27.62	0.865	25.62	0.817	31.13	0.930	29.95	0.897	27.68	0.873
$\epsilon = 0.02$	28.63	0.893	27.03	0.861	24.91	0.803	30.52	0.924	29.45	0.890	27.01	0.866
$\epsilon = 0.05$	27.74	0.884	26.41	0.852	23.07	0.796	29.72	0.913	28.55	0.881	26.22	0.853

Table 6: Comparison of PSNR and SSIM values under different degree PGD attacks for the network without the noise optimization algorithm.

Degree of perturbation	Gaussian noise						Real noise					
	$\sigma = 25$		$\sigma = 50$		$\sigma = 75$		$L = 1$		$L = 2$		$L = 3$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$\epsilon = 0$	27.16	0.854	24.25	0.822	21.12	0.724	28.34	0.892	27.44	0.833	23.61	0.807
$\epsilon = 0.01$	25.57	0.753	22.43	0.672	18.00	0.627	27.02	0.830	25.95	0.783	21.72	0.727
$\epsilon = 0.02$	23.72	0.682	20.02	0.581	16.61	0.539	25.64	0.796	23.70	0.702	18.69	0.651
$\epsilon = 0.05$	21.12	0.532	18.03	0.506	15.03	0.481	22.58	0.702	20.48	0.578	16.84	0.587

indicate that the NOP-GAN effectively resists attacks across different attack scenarios, exhibiting superior robustness compared to other mainstream models.

4.5.1. Additional Notes: RR. This paper introduces the concept of the "RR" to effectively evaluate a model's resilience against attacks and intuitively capture the stability of multiple vital metrics (e.g., PSNR, SSIM, LPIPS, and SNR) under adversarial conditions. The RR is defined as the degree to which a model retains its original performance when subjected to attacks. It quantifies the model's robustness against attacks: The closer the RR is to 1, the stronger the model's resilience, indicating minimal performance degradation; a lower value signifies significant degradation postattack. By directly translating the magnitude of changes into a stability metric, a high RR reflects the model's robustness postattack. In addition, by applying weighted adjustments to the change rates of different metrics, we can more equitably assess the model's overall performance. This metric offers a more intuitive evaluation of the model's robustness, allowing the differences in the performance under various attack scenarios to be quantified, thereby providing a quantitative basis for improving model robustness.

The formula for defining RR is as follows:

$$\text{Resistance Rate} = 1 - \text{Change Rate}, \tag{16}$$

where the Change Rate is used to measure the relative variation of each metric under the impact of the attack and is calculated as follows:

$$\text{Change Rate} = \left|\frac{\text{Original Value} - \text{Post-Attack Value}}{\text{Original Value}}\right|. \tag{17}$$

Therefore, for each specific metric (such as PSNR, SSIM, LPIPS, and SNR), the RR formula can be expanded as follows:

$$\begin{aligned}
\text{Resistance Rate}_{\text{PSNR}} &= 1 - \left|\frac{\text{PSNR}_{\text{Original}} - \text{PSNR}_{\text{Post-Attack}}}{\text{PSNR}_{\text{Original}}}\right|, \\
\text{Resistance Rate}_{\text{SSIM}} &= 1 - \left|\frac{\text{SSIM}_{\text{Original}} - \text{SSIM}_{\text{Post-Attack}}}{\text{SSIM}_{\text{Original}}}\right|, \\
\text{Resistance Rate}_{\text{LPIPS}} &= 1 - \left|\frac{\text{LPIPS}_{\text{Original}} - \text{LPIPS}_{\text{Post-Attack}}}{\text{LPIPS}_{\text{Original}}}\right|, \\
\text{Resistance Rate}_{\text{SNR}} &= 1 - \left|\frac{\text{SNR}_{\text{Original}} - \text{SNR}_{\text{Post-Attack}}}{\text{SNR}_{\text{Original}}}\right|.
\end{aligned} \tag{18}$$

In the equation (18), the RR is defined in the form of (1—Change Rate), converting "change" into "retention," meaning a higher RR indicates a greater ability of the model to retain its original performance under attack. The absolute value of the Change Rate ensures that the measure reflects only the magnitude of change, independent of its direction. To obtain an overall resistance capability of the model across multiple key metrics, we use a CRR, defined as the average of the RRs of each metric:

$$\text{Comprehensive Resistance Rate} = \frac{\text{Resistance Rate}_{\text{PSNR}} + \text{Resistance Rate}_{\text{SSIM}}}{4} + \frac{\text{Resistance Rate}_{\text{LPIPS}} + \text{Resistance Rate}_{\text{SNR}}}{4}. \tag{19}$$

Table 7: Performance comparison of various models under FGSM attacks at different Gaussian noise levels, with an attack intensity of $\epsilon = 0.05$.

Method	$\sigma = 25$					$\sigma = 50$					$\sigma = 75$				
	PSNR ↑	**SSIM ↑**	**LPIPS ↓**	**SNR ↑**	**CRR ↑**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**SNR ↑**	**CRR ↑**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**SNR ↑**	**CRR ↑**
DnCNN	**29.29**	0.74	0.18	**17.80**	87.20%	25.83	0.66	0.22	14.86	89.79%	/	/	/	/	/
TNRD	20.89	0.43	0.30	8.56	55.88%	22.58	0.50	0.31	9.56	69.37%	23.68	0.61	0.32	11.75	84.40%
FFDNet	29.72	0.75	0.18	18.12	88.35%	26.15	0.67	0.24	14.99	89.23%	23.08	0.60	0.27	12.59	86.28%
RDDCNN	23.34	0.47	0.39	12.12	63.49%	21.17	0.38	0.51	10.55	66.05%	/	/	/	/	/
SCUNet	24.41	0.45	0.45	13.10	49.01%	24.42	0.55	0.30	13.51	76.61%	/	/	/	/	/
NOP-GAN	28.23	**0.87**	**0.07**	17.77	**92.05%**	**26.39**	**0.82**	**0.17**	**16.90**	**92.23%**	**25.59**	**0.81**	**0.20**	**17.84**	**89.06%**

Table 8: Performance comparison of various models under PGD attacks at different Gaussian noise levels, with an attack intensity of $\epsilon = 0.05$.

Method	$\sigma = 25$					$\sigma = 50$					$\sigma = 75$				
	PSNR ↑	SSIM ↑	LPIPS ↓	SNR ↑	CRR ↑	PSNR ↑	SSIM ↑	LPIPS ↓	SNR ↑	CRR ↑	PSNR ↑	SSIM ↑	LPIPS ↓	SNR ↑	CRR ↑
DnCNN	**29.29**	0.75	0.18	17.82	88.05%	25.99	0.68	0.21	15.05	91.04%	/	/	/	/	/
TNRD	21.08	0.44	0.31	8.98	56.67%	22.85	0.46	0.30	11.26	71.36%	23.52	0.63	0.29	11.84	86.36%
FFDNet	29.07	0.75	0.18	17.58	87.19%	26.77	0.69	0.21	15.65	92.23%	23.36	0.61	0.26	12.91	87.43%
RDDCNN	23.25	0.46	0.48	12.17	59.15%	20.01	0.33	0.64	9.79	37.17%	/	/	/	/	/
SCUNet	26.77	0.69	0.22	15.21	74.76%	25.85	0.69	0.20	14.97	87.17%	/	/	/	/	/
NOP-GAN	28.10	**0.87**	**0.08**	**17.90**	**92.88%**	**27.99**	**0.83**	**0.15**	**18.71**	**97.74%**	**26.14**	**0.82**	**0.18**	**17.86**	**89.97%**

5. Additional Verification

This section presents additional experiments on the NOP-GAN to validate its classification accuracy and robustness under combined noise and adversarial interference. In Section 5.1, we presented the image classification method along with the experimental methodology and dataset. In Section 5.2, we validated the effectiveness of the NOP-GAN image denoising method in the context of image classification tasks, assessing its impact on classification accuracy. This serves to prove that our model not only exhibits robust denoising capabilities but also maintains strong robustness and accuracy when applied to image classification. In Section 5.3, we further verified the adversarial robustness of the NOP-GAN based on the classification task. The experiment validates the effectiveness of our noise optimization in enhancing the model's robustness.

5.1. Implementation. Image classification is a fundamental and widely used task in computer vision, which aims to assign a label to an image based on its content. However, image classification performance can be severely degraded by the presence of noise in the image, which can obscure the discriminative features and mislead the classifier. Therefore, it is important to evaluate the effectiveness of image-denoising methods on image classification accuracy [68, 69]. We applied our model to the MNIST and CIFAR-10 datasets, comparing the accuracy of the datasets before and after denoising. We also compared the classification performance of the MNIST dataset under different degrees of adversarial attacks, proving that our model not only possesses robust denoising effects but also exhibits good robustness and accuracy when applied to image classification.

> Datasets: The MNIST dataset is a classic dataset consisting of 60,000 training samples and 10,000 test samples. Each sample is a grayscale image of a handwritten digit, with a size of 28×28 pixels. The CIFAR-10 dataset contains 50,000 training images and 10,000 test images. All of the images are three-channel RGB color images with a size of 32×32 pixels. The dataset consists of 10 classes. Both datasets are used to test the model's recognition accuracy under various noise conditions.

5.2. Classifications Without Resistance Attack. We conducted experiments on the MNIST and CIFAR-10 datasets by adding Gaussian noise with standard deviations of 25, 50, and 75, as well as real noise with levels of 1, 2, and 3. We used it to represent the standard deviation of Gaussian noise and to represent the perturbation of real noise. First, we obtained classification accuracies for the noisy images using only the CNN model at different noise levels. Then, we used our proposed NOP-GAN denoising model to remove noise from the noisy images. We then employed our classification network to classify the denoised images and determine the classification accuracies after denoising with different types and levels of noise on the dataset. The experimental results on the two datasets are shown in Table 9.

The results show that the classification network achieves an accuracy of over 99% on the MNIST dataset without any added noise. However, when Gaussian or real noise is added, the classification accuracy decreases as the noise level increases if a robust denoising technique is not used. By combining our proposed denoising network and classification network, we can maintain an accuracy of over 99% under different noise environments. Furthermore, the experimental error is less than ±0.1%. The impact on the final classification accuracy is minimal. This proves that the NOP-GAN is feasible for image denoising, and it has excellent denoising performance that satisfies the classification problem in different noise environments, avoiding the influence of noise on the final recognition effect.

CIFAR-10 is a more challenging dataset for image recognition than the MNIST dataset due to its RGB images. The classification network achieves an accuracy of 90% without any added noise. However, with Gaussian or real noise added, the classification accuracy decreases without any denoising technique, indicating poor robustness of the classification network. By combining our proposed denoising network with the classification network, we were able to achieve an accuracy of 90% in various noisy environments. Furthermore, the experimental error remained stable within a range of ±1%. This demonstrates that the network we proposed exhibits significant denoising performance for grayscale images and strong denoising capabilities for RGB color images, enabling accurate classification of color images.

5.3. Classification Under Resistance Attack. FGSM and PGD are two primary adversarial attack strategies on image classification. Their objective is to mislead the predictions of deep learning models by introducing nearly imperceptible minute perturbations to the input data. When these attack strategies are applied to the MNIST dataset, there is a significant decline in the predictive accuracy of the models. In other words, even minute perturbations can potentially lead a model which could correctly classify otherwise to misclassify digits. This phenomenon reveals the potential vulnerability of deep learning models when faced with adversarial attacks [70].

According to the data in Table 10, we observed a clear trend: As the intensity of FGSM and PGD attacks increases, the accuracy of traditional CNN models continuously declines. This suggests that while these models may perform well without noise interference, their resistance to noise interference is relatively weak. This finding emphasizes the need to consider robustness when designing and training deep learning models and to take appropriate measures to enhance the model's resistance to various potential attacks.

Upon the introduction of the NOA, the robustness of the model significantly improves. As shown in Table 11, we first used our NOP-GAN model to denoise the noisy images and then performed classification under the same model. We observe that even with an increase in the intensity of FGSM and PGD attacks, the model's classification accuracy does not significantly decline, consistently remaining above 99%.

Table 9: The classify algorithm compares the classification accuracy on the MNIST and CIFAR-10 datasets.

Methods	No noise	Gaussian noise			Real noise		
		$\sigma = 25$	$\sigma = 50$	$\sigma = 75$	$L = 1$	$L = 2$	$L = 3$
Classification accuracy of the MNIST dataset (%)							
CNN (with noise)	99.43	97.45	96.22	93.86	98.22	97.41	94.05
NOP-GAN (denoise)	99.43	99.24	99.27	99.14	99.30	99.30	99.28
Classification accuracy of the CIFAR-10 dataset (%)							
CNN (with noise)	90.54	84.42	80.52	75.96	86.34	83.45	77.45
NOP-GAN (denoise)	90.54	90.02	89.63	89.12	90.09	90.10	89.52

Table 10: Classification accuracy on the MNIST under different degree FGSM and PGD attacks with NOP-GAN (%).

Degree of perturbation	No noise	Gaussian noise and FGSM attack			Gaussian noise and PGD attack		
		$\sigma = 25$	$\sigma = 50$	$\sigma = 75$	$\sigma = 25$	$\sigma = 50$	$\sigma = 75$
$\epsilon = 0$	99.43	99.39	99.27	99.07	99.31	99.18	99.02
$\epsilon = 0.01$	99.33	99.30	99.17	99.01	99.25	99.14	99.01
$\epsilon = 0.02$	99.29	99.25	99.04	98.87	99.13	99.12	98.87
$\epsilon = 0.05$	99.13	99.21	98.92	98.78	99.23	99.03	98.73

Table 11: Classification accuracy on the MNIST under different degree FGSM and PGD attacks only with CNN (%).

Degree of perturbation	No noise	Gaussian noise and FGSM attack			Gaussian noise and PGD attack		
		$\sigma = 25$	$\sigma = 50$	$\sigma = 75$	$\sigma = 25$	$\sigma = 50$	$\sigma = 75$
$\epsilon = 0$	99.43	98.85	98.70	97.86	98.99	98.58	97.85
$\epsilon = 0.01$	98.97	98.40	97.50	97.45	98.63	97.71	97.37
$\epsilon = 0.02$	98.83	97.61	97.32	96.91	97.78	97.20	96.85
$\epsilon = 0.05$	97.35	96.81	96.23	95.47	97.19	96.64	96.11

This indicates that under the dual interference of added noise and attacks, our model, through the introduction of the NOA, can effectively resist adversarial attacks and noise, maintaining high predictive accuracy. This result validates the robustness of the model.

6. Conclusions

In this paper, we introduced the NOP-GAN, a powerful image denoising model. The NOP-GAN modifies the generator within the GAN framework by using a U-Net architecture and the NOA which operates directly during BP, as well as the PCA structure which combines spatial pyramid feature fusion and attention mechanisms. These modifications significantly enhance the network's denoising capabilities and improve the model's robustness. Objective and subjective evaluations demonstrated that the NOP-GAN possesses excellent denoising capabilities. Furthermore, attack experiments based on FGSM and PGD validated the antidisturbance ability of the NOP-GAN. In addition, classification task experiments further confirmed the model's denoising performance and the robustness of the network. In summary, our proposed method demonstrates superior performance, establishing itself as a promising solution with significant implications for denoising and subsequent image analysis tasks.

Nomenclature

PCA Pyramid coordinate attention
NOA Noise optimization algorithm.

Data Availability Statement

The data supporting the findings of this study are available from the first author upon reasonable request. Requests should be directed to Min-Ling Zhu at zhuminling@bistu.edu.cn.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This research was supported by the National Natural Science Foundation of China (No. 31900979, No. 92354307), the Fundamental Research Funds for the Central Universities (No. 2023RC09), and the Qiyuan Innovation Foundation (S20210201067) and subthemes (No. 9072323404).

References

[1] J. Wu, "Wavelet Domain Denoising Method Based on Multistage Median Filtering," *The Journal of China Universities of Posts and Telecommunications* 20, no. 2 (2013): 113–119, https://doi.org/10.1016/s1005-8885(13)60037-0.

[2] A. Kumar and S. S. Sodhi, "Comparative Analysis of Gaussian Filter, Median Filter and Denoise Autoenocoder," in *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)* (IEEE, 2020), 45–51.

[3] U. Erkan, D. N. H. Thanh, L. M. Hieu, and S. Enginoglu, "An Iterative Mean Filter for Image Denoising," *IEEE Access* 7 (2019): 167847–167859, https://doi.org/10.1109/access.2019.2953924.

[4] C. T. Lu, M. Y. Chen, J. H. Shen, L. L. Wang, N. Y. Yen, and C. H. Liu, "X-Ray Bio-Image Denoising Using Directional-Weighted Mean Filtering and Block Matching Approach," *Journal of Ambient Intelligence and Humanized Computing* 1 (2018): 1–18, https://doi.org/10.1007/s12652-018-0692-8.

[5] C. Tian, Y. Yuan, S. Zhang, C. W. Lin, W. Zuo, and D. Zhang, "Image Super-Resolution With an Enhanced Group Convolutional Neural Network," *Neural Networks* 153 (2022): 373–385, https://doi.org/10.1016/j.neunet.2022.06.009.

[6] C. Tian, Y. Zhang, W. Zuo, C. W. Lin, D. Zhang, and Y. Yuan, "A Heterogeneous Group Cnn for Image Super-Resolution," *IEEE Transactions on Neural Networks and Learning Systems* 35, no. 5 (2024): 6507–6519, https://doi.org/10.1109/tnnls.2022.3210433.

[7] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu, "Neighbor2neighbor: Selfsupervised Denoising From Single Noisy Images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 14781–14790.

[8] J. Li, Z. Zhang, X. Liu, et al., "Spatially Adaptive Self-Supervised Learning for Real-World Image Denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 9914–9924, https://doi.org/10.1109/cvpr52729.2023.00956.

[9] Z. Zhang, Y. Jiang, W. Shao, et al., "Realtime Controllable Denoising for Image and Video," in *Proceedings of theIEEE/CVFConferenceon ComputerVision andPatternRecognition* (2023), 14028–14038, https://doi.org/10.1109/cvpr52729.2023.01348.

[10] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep Cnn for Image Denoising," *IEEE Transactions on Image Processing* 26, no. 7 (2017): 3142–3155, https://doi.org/10.1109/tip.2017.2662206.

[11] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a Fast and Flexible Solution for Cnn-Based Image Denoising," *IEEE Transactions on Image Processing* 27, no. 9 (2018): 4608–4622, https://doi.org/10.1109/tip.2018.2839891.

[12] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward Convolutional Blind Denoising of Real Photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 1712–1722, https://doi.org/10.1109/cvpr.2019.00181.

[13] J. Chen, J. Chen, H. Chao, and M. Yang, "Image Blind Denoising with Generative Adversarial Network Based Noise Modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 3155–3164, https://doi.org/10.1109/cvpr.2018.00333.

[14] D. W. Kim, J. R. Chung, and S. W. Jung, "Grdn: Grouped Residual Dense Network for Real Image Denoising and Gan-Based Realworld Noise Modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2019), 2086–2094, https://doi.org/10.1109/cvprw.2019.00261.

[15] Z. Yue, Q. Zhao, L. Zhang, and D. Meng, "Dual Adversarial Network: Toward Real-World Noise Removal and Noise Generation," in *Computer Vision—ECCV 2020* (Springer, 2020), 41–58.

[16] H. Zhang and K. Sakurai, "Conditional Generative Adversarial Network-Based Image Denoising for Defending Against Adversarial Attack," *IEEE Access* 9 (2021): 169031–169043, https://doi.org/10.1109/access.2021.3137637.

[17] Y. Wang, D. Chang, and Y. Zhao, "A New Blind Image Denoising Method Based on Asymmetric Generative Adversarial Network," *IET Image Processing* 15, no. 6 (2021): 1260–1272, https://doi.org/10.1049/ipr2.12102.

[18] Y. Zhou, K. Yu, M. Wang, et al., "Speckle Noise Reduction for Oct Images Based on Image Style Transfer and Conditional gan," *IEEE Journal of Biomedical and Health Informatics* 26, no. 1 (2022): 139–150, https://doi.org/10.1109/jbhi.2021.3074852.

[19] Q. Xie, Z. Ma, L. Zhu, et al., "Multi-Task Generative Adversarial Network for Retinal Optical Coherence Tomography Image Denoising," *Physics in Medicine and Biology* 68, no. 4 (2023): 045002, https://doi.org/10.1088/1361-6560/ac944a.

[20] H. Yan, J. Zhang, J. Feng, M. Sugiyama, and V. Y. F. Tan, "Towards Adversarially Robust Deep Image Denoising," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence* (IJCAI-22, 2022), 1516–1522.

[21] Q. Zhang, J. Xiao, C. Tian, J. Chun-Wei Lin, and S. Zhang, "A Robust Deformed Convolutional Neural Network (Cnn) for Image Denoising," *CAAI Transactions on Intelligence Technology* 8, no. 2 (2023): 331–342, https://doi.org/10.1049/cit2.12110.

[22] C. Szegedy, W. Zaremba, I. Sutskever, et al., "Intriguing Properties of Neural Networks," (2014), https://arxiv.org/abs/1312.6199.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (Springer, 2015), 234–241.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generativeadversarial Nets," *Advances in Neural Information Processing Systems* 27 (2014).

[25] X. Pan, A. Tewari, T. Leimkühler, L. Liu, A. Meka, and C. Theobalt, "Drag Your Gan: Interactive Point-Based Manipulation on the Generative Image Manifold," in *ACM SIGGRAPH 2023 Conference Proceedings* (2023), 1–11, https://doi.org/10.1145/3588432.3591500.

[26] X. Su, W. Qiang, J. Hu, C. Zheng, F. Wu, and F. Sun, "Intriguing Property and Counterfactual Explanation of Gan for Remote Sensing Image Generation," *International Journal of Computer Vision* 132, no. 11 (2024): 5192–5216, https://doi.org/10.1007/s11263-024-02125-4.

[27] S. M. Bafti, C. S. Ang, G. Marcelli, M. M. Hossain, S. Maxamhud, and A. D. Tsaousis, "Biogan: An Unpaired Gan-Based Image to Image Translation Model for Microbiological Images," (2023), https://arxiv.org/abs/2306.06217.

[28] S. Yang, L. Jiang, Z. Liu, and C. C. Loy, "Gp-Unit: Generative Prior for Versatile Unsupervised Image-to-Image Translation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, no. 10 (2023): 11869–11883, https://doi.org/10.1109/tpami.2023.3284003.

[29] O. Fedoruk, K. Klimaszewski, A. Ogonowski, and R. Możdżonek, "Performance of Gan-Based Augmentation for Deep Learning Covid-19 Image Classification," *AIP Conference Proceedings* 3061, no. 1 (2024): 030001.

[30] X. Zhang, Z. Wang, M. Ji, Y. Li, Q. Pan, and K. Lu, "A Sea–Land Clutter Classification Framework for Over-the-Horizon Radar Based on Weighted Loss Semi-Supervised Generative Adversarial Network," *Engineering Applications of Artificial Intelligence* 133 (2024): 108526, https://doi.org/10.1016/j.engappai.2024.108526.

[31] Y. Li, K. Zhang, W. Shi, Y. Miao, and Z. Jiang, "A Novel Medical Image Denoising Method Based on Conditional Generative Adversarial Network," *Computational and Mathematical Methods in Medicine* 2021, no. 1 (2021): 1–11, https://doi.org/10.1155/2021/9974017.

[32] Y. Tian, X. Chai, Z. Gan, Y. Lu, Y. Zhang, and S. Song, "Swdgan: Gan-Based Sampling and Whole Image Denoising Network for Compressed Sensing Image Reconstruction,"

Journal of Electronic Imaging 30, no. 06 (2021): 063017, https://doi.org/10.1117/1.jei.30.6.063017.

[33] Z. Huang, J. Zhang, Y. Zhang, and H. Shan, "Du-gan: Generative Adversarial Networks With Dual-Domain U-Net-Based Discriminators for Low-Dose Ct Denoising," *IEEE Transactions on Instrumentation and Measurement* 71 (2022): 1–12, https://doi.org/10.1109/tim.2021.3128703.

[34] S. Asmussen and P. W. Glynn, *Stochastic Simulation: Algorithms and Analysis*, 57 (Springer, 2007).

[35] Y. C. Ho and X. R. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems* (New York, NY: Springer, 1991).

[36] R. Y. Rubinstein and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*, 13 (Wiley, 1993).

[37] B. Heidergott and H. Leahu, "Weak Differentiability of Product Measures," *Mathematics of Operations Research* 35, no. 1 (2010): 27–51, https://doi.org/10.1287/moor.1090.0422.

[38] Y. Peng, M. C. Fu, J. Q. Hu, and B. Heidergott, "A New Unbiased Stochastic Derivative Estimator for Discontinuous Sample Performances With Structural Parameters," *Operations Research* 66, no. 2 (2018): 487–499, https://doi.org/10.1287/opre.2017.1674.

[39] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte Carlo Gradient Estimation in Machine Learning," *Journal of Machine Learning Research* 21, no. 132 (2020): 1–62.

[40] L. Xiao, Z. Zhang, and Y. Peng, "Noise Optimization for Artificial Neural Networks," (2021), https://arxiv.org/abs/2102.04450.

[41] R. Archana and P. S. E. Jeevaraj, "Deep Learning Models for Digital Image Processing: A Review," *Artificial Intelligence Review* 57, no. 1 (2024): 11, https://doi.org/10.1007/s10462-023-10631-z.

[42] M. Towers, A. Kwiatkowski, J. Terry, et al., "Gymnasium: A Standard Interface for Reinforcement Learning Environments," (2024), https://arxiv.org/abs/2407.17032.

[43] A. Abid, M. T. Khan, and J. Iqbal, "A Review on Fault Detection and Diagnosis Techniques: Basics and beyond," *Artificial Intelligence Review* 54, no. 5 (2021): 3639–3664, https://doi.org/10.1007/s10462-020-09934-2.

[44] J. K. Tsotsos, "Analyzing Vision at the Complexity Level," *Behavioral and Brain Sciences* 13, no. 3 (1990): 423–445, https://doi.org/10.1017/s0140525x00079577.

[45] J. K. Tsotsos, *A Computational Perspective on Visual Attention* (MIT Press, 2021).

[46] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gatherexcite: Exploiting Feature Context in Convolutional Neural Networks," *Advances in Neural Information Processing Systems* 31 (2018).

[47] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 7132–7141.

[48] Q. Hou, L. Zhang, M. M. Cheng, and J. Feng, "Strip Pooling: Rethinking Spatial Pooling for Scene Parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 4003–4012.

[49] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross Attention for Semantic Segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 603–612, https://doi.org/10.1109/iccv.2019.00069.

[50] T. Sun, B. Guo, T. Ruan, X. Zhou, and D. Bai, "Raileinet: A Novel Scene Segmentation Network for Automatic Train Operation Based on Feature Alignment," *Engineering Applications of Artificial Intelligence* 138 (2024): 109295, https://doi.org/10.1016/j.engappai.2024.109295.

[51] Q. Hou, D. Zhou, and J. Feng, "Coordinate Attention for Efficient Mobile Network Design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 13713–13722.

[52] M. L. Zhu, L. L. Zhao, and L. Xiao, "Image Denoising Based on Gan With Optimization Algorithm," *Electronics* 11, no. 15 (2022): 2445, https://doi.org/10.3390/electronics11152445.

[53] L. Xiao, Z. Zhang, J. Jiang, and Y. Peng, "Noise Optimization in Artificial Neural Networks," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)* (2022), 1595–1600, https://doi.org/10.1109/case49997.2022.9926712.

[54] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising With Block-Matching and 3d Filtering," in *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, 6064 (2006), 354–365.

[55] Y. Chen and T. Pock, "Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 6 (2017): 1256–1272, https://doi.org/10.1109/tpami.2016.2596743.

[56] K. Zhang, Y. Li, J. Liang, et al., "Practical Blind Image Denoising via Swin-Conv-unet and Data Synthesis," *Machine Intelligence Research* 20, no. 6 (2023): 822–836, https://doi.org/10.1007/s11633-023-1466-0.

[57] H. Guo, J. Li, T. Dai, Z. Ouyang, X. Ren, and S. T. M. Xia, "A Simple Baseline for Image Restoration With State-Space Model," in *Computer Vision—ECCV 2024* (Springer, 2024), 222–241.

[58] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, no. 5 (2011): 898–916, https://doi.org/10.1109/tpami.2010.161.

[59] A. Abdelhamed, S. Lin, and M. S. Brown, "A High-Quality Denoising Dataset for Smartphone Cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 1692–1700, https://doi.org/10.1109/cvpr.2018.00182.

[60] C. Tian, M. Zheng, C. W. Lin, Z. Li, and D. Zhang, "Heterogeneous Window Transformer for Image Denoising," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 54, no. 11 (2024): 6621–6632, https://arxiv.org/abs/2407.05709, https://doi.org/10.1109/tsmc.2024.3429345.

[61] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, "Real-World Noisy Image Denoising: A New Benchmark," (2018), https://arxiv.org/abs/1804.02603.

[62] H. Li, S. Xia, B. Zhou, and J. Peng, "The Growth Mechanism of Grain Boundary Carbide in Alloy 690," *Materials Characterization* 81 (2013): 1–6, https://doi.org/10.1016/j.matchar.2013.04.005.

[63] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: from Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing* 13, no. 4 (2004): 600–612, https://doi.org/10.1109/tip.2003.819861.

[64] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 586–595.

[65] D. M. Chandler and S. S. Hemami, "Vsnr: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images," *IEEE Transactions on Image Processing* 16, no. 9 (2007): 2284–2298, https://doi.org/10.1109/tip.2007.901820.

[66] S. Yao, W. Lin, E. Ong, and Z. Lu, "Contrast Signal-to-Noise Ratio for Imagequalityassessment," in *IEEEInternationalConferenceon-Image Processing 2005*, 1 (IEEE, 2005), I–397.
[67] P. Gong, J. Liu, and S. Lv, "Image Denoising With Gan Based Model," *Journal of Information Hiding and Privacy Protection* 2, no. 4 (2020): 155–163, https://doi.org/10.32604/jihpp.2020.010453.
[68] F. Liu, Q. Song, and G. Jin, "The Classification and Denoising of Image Noise Based on Deep Neural Networks," *Applied Intelligence* 50, no. 7 (2020): 2194–2207, https://doi.org/10.1007/s10489-019-01623-0.
[69] D. Sil, A. Dutta, and A. Chandra, "Convolutional Neural Networks for Noise Classification and Denoising of Images," in *TENCON 20192019 IEEE Region 10 Conference (TENCON)* (IEEE, 2019), 447–451.
[70] S. Hu, E. Nalisnick, and M. Welling, "Adversarial Defense via Image Denoising With Chaotic Encryption," (2022), https://arxiv.org/abs/2203.10290.