Journal of Global Information Management
Volume 30 • Issue 9

WGAN-Based Image Denoising Algorithm

XiuFang Zou, South China Normal University, China
Dingju Zhu, South China Normal University, China*
Jun Huang, The First Affiliated Hospital of Jinan University, China
Wei Lu, South China Normal University, China
Xinchu Yao, South China Normal University, China
Zhaotong Lian, University of Macau, China

ABSTRACT

Traditional image denoising algorithms are generally based on spatial domains or transform domains to denoise and smooth the image. The denoised images are not exhaustive, and the depth-of-learning algorithm has better denoising effect and performs well while retaining the original image texture details such as edge characters. In order to enhance denoising capability of images by the restoration of texture details and noise reduction, this article proposes a network model based on the Wasserstein GAN. In the generator, small convolution size is used to extract image features with noise. The extracted image features are denoised, fused and reconstructed into denoised images. A new residual network is proposed to improve the noise removal effect. In the confrontation training, different loss functions are proposed in this paper.

KEYWORDS

Image Denoising, Residual Network, Wasserstein GAN

INTRODUCTION

The generation of image noise generally occurs during the process of image acquisition and transmission. Image denoising is the process of removing the noise contained in an image to reduce the influence of noise on the image. Therefore, image denoising is particularly important. Common image noises include Gaussian noise, salt and pepper noise, multiplicative noise, and other mixed noises, which are composed of multiple single noises.

Through the continuous development of deep learning, deep learning has made significant contributions to the field of computer vision. In 2014, Goodfellow et al. (Goodfellow et al., 2014) proposed a network model based on Generate Antagonistic Networks—GAN. GAN contains a generator and a discriminator. The generator can generate highly realistic images based on the input image and noise through a discriminator confrontation training process. After the GAN model was proposed, it was widely loved by researchers. Up to now, GAN optimization and derivative models emerge endlessly. GAN has made extraordinary contributions to the field of images. GAN can not only generate ultra-high pixel images, transform images, learn image migration, realize image repair, but also generate videos and predict the next trend of videos. In addition to the field of images, scholars also try to apply GAN in different fields, such as text generation (Dai, 2018; Fedus, Goodfellow, & Dai,

DOI: 10.4018/JGIM.300821 *Corresponding Author

2018; L. Yu, Zhang, Wang, & Yu, 2017; Yizhe Zhang, Gan, & Carin, 2016), cryptography (Gomez et al., 2018) and so on. Improved GAN models including Info-GAN, DC-GAN, f-GAN, Cat-GAN (D. Zhang, Dongru, Kang, & Zhang, 2019). Since the GAN method adopts a feed-forward neural network structure, it eliminates the need to sample potentially complex high-dimensional processes, and can be easily extended to systems with a large number of uncertainties. As a rookie in the field of artificial intelligence in recent years, GAN's excellent data generation ability has emerged as a new hotspot in data quality research.

The image denoising method combined with deep Convolutional Neural Networks achieves very good denoising effect (Jiang, Xiao, & Shao-hui, 2019; K. Zhang, Zuo, Chen, Meng, & Zhang, 2017; Yizhe Zhang et al., 2016). Similarly, GAN can also be used for image denoising as well, since GAN generator and discriminator can use neural networks and functions. Image denoising with GAN can preserve the texture details of image while denoising, which can not only achieve good denoising effects, but also retain texture details and content of the image. Image noise reduction using GAN has increasingly become a research hotspot as researchers become more enthusiastic about GAN.

1 RELATED WORK

The combination of deep learning has a significant impact on image denoising. The main methods include multilayer perceptron (Rosenblatt, 1959), automatic encoder (Rumelhart, Hinton, & Williams, 1985), and deep convolutional neural network (LeCun et al., 1989). Burger et al. (Harold Christopher Burger, Christian J Schuler, & Stefan Harmeling, 2012a, 2012b) used multilayer perceptron for image denoising, which had a better denoising impact than BM3D. In 2017, Zhang et al. (K. Zhang et al., 2017) proposed a deep convolutional neural Gaussian noise denoising network (DnCNN) based on residual learning. Its denoising result is better than that of other traditional denoising algorithms, and it is currently considered as one of the best denoising algorithms. Cheng Chung Ming at al. (C.-M. Chen, Zhang, & Hsu, 2020) developed a new model to achieve extraction of weak edges in speckle noisy images, the level set model called Edge Attraction Force. Kuo-Kun (Tseng et al., 2019) used Kalman filtering to estimate the state of a dynamic system based on a series of measurements that did not fully account for the noise.

DnCNN is suitable for Gaussian noise, and the denoising effect of other noises is not as good as Gaussian noise. Most of the existing denoising methods are based on known noise, and the noise type and distribution information are indeterminate. In GAN, the generator inputs the picture with known noise, the discriminator outputs the corresponding noiseless image (Yuanqi, 2018) combined with Generative Adversarial Networks and denoised the color image, transforms the noisy image into feature domain for processing through the neural network, and retained the rich texture details of the image with multi-scale features. Xie Chuan et al. (Xie, Wang, Lin, Zheng, & Zhao, 2018) used the Generative Adversarial Network to denoise Monte Carlo noise. For the unknown noise information contained in the image, that is, blind denoising of the image, GAN has strong learning ability, can learn complex distributions and can be used to construct paired noiseless-noisy image training data set. Chen et al.(J. Chen, Chen, Chao, & Yang, 2018) proposed an image blind denoising algorithm based on Generative Adversarial Networks and Convolutional Neural Networks. The basic idea of this algorithm is to first extract noise image blocks, then use GAN for noise modeling to generate noise samples, use the extracted noise to quickly construct paired training dataset, and finally achieve blind noise de-noising by using depth CNN method. When GAN is used to model noise, it can improve the diversity of noise and train CNN to carry out blind denoising, which not only has better denoising impact but also retains the image texture details effectively. Image denoising usually requires a large number of training samples, including training pairs of clean and noise-containing images, but completely clean images are often difficult to obtain. Jaakko et al.(Lehtinen et al., 2018) proposed a denoising method that does not require noiseless images as training samples, and its input and output are both noise-containing images. Yong Zhang et al.(Yong Zhang et al., 2020) proposed an enhanced

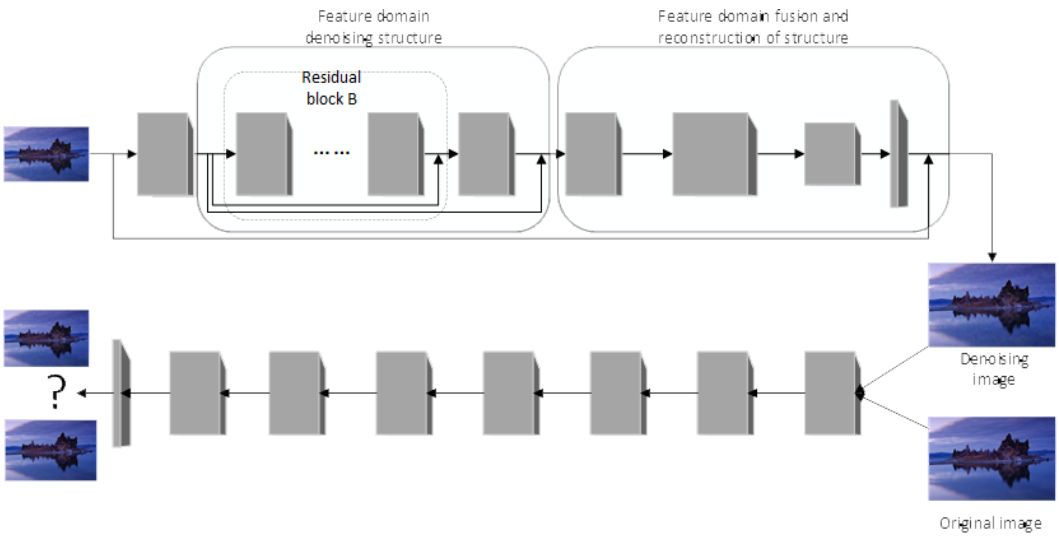
adversarial network model. The proposed model can generate sharp images and eliminate draught board artefacts by using the weight of feature channel. In addition, the mixed loss function enables the network to output high-quality images.

2 IMAGE DENOISING MODEL

In this article, we develop a novel Generate Antagonistic Denoising Network, in which both the generator and discriminator are constructed using deep Convolutional Neural Networks. A new residual network has been designed for the generate network to denoise the image and improve the accuracy of training by introducing a gradient punishment in the discriminant network. Figure 2-1 shows the structure of t Generate Antagonistic Denoising Networks.

In the generate network, we first extract features using the convolutional layer, then we get the preliminary denoised image after the denoising structure of the feature domain, and finally we get the final denoised image after the fusion and reconstruction of the structure in the feature domain (Figure 2-1). The original and the denoising images obtained from the generate network are used as inputs to the discriminant network. After the discriminant network passes through the network layer, the input is judged, and the output is fed back to the generate network to optimize the network and generate more realistic denoising images. The algorithm used in this study is shown in Table 2-1, and the generate network and discriminant network are described in next section.

Figure 1. The overall structure of Generate Antagonistic Blind Gaussian Denoising Network



2.1 Generating Network

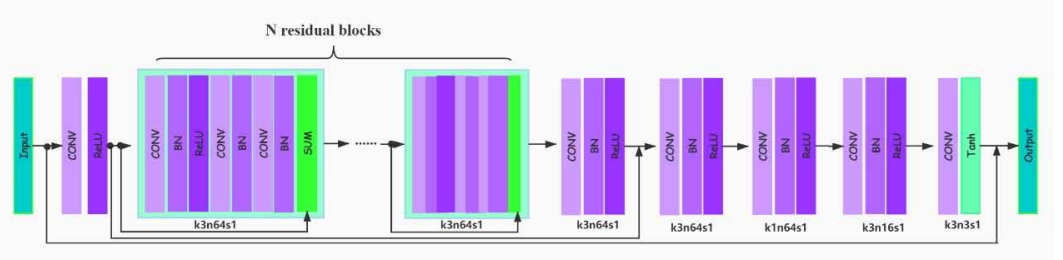
In this study, we propose a new generate network for image denoising combining Residual Network and Convolutional Neural Networks. Generator network consists of feature extraction, feature denoising, feature fusion and feature reconstruction. Feature extraction is the first layer of convolution, which includes a layer of convolutional network and activation function to extract image features from the input noisy image. The second part is image feature denoising. The Residual Network learning noise distribution is introduced to obtain the residual image for image denoising. Finally, feature

Algorithm: Ours. We use default values of $\lambda = 10, n_{critic} = 5, \alpha = 0.0001, \beta_1 = 0, \beta_2 = 0.9$.

1. While θ has not converged do
2. **For** t=1,2,..., n_{critic} **do**
3. **For** i=1,2,..,m **do**
4. Sample real data $x \sim P_r$,latent variable z $\sim p(z)$,a random number $\varepsilon \sim U[0,1]$
5. $\hat{x} \leftarrow G_\theta(z)$
6. $\tilde{x} \leftarrow \varepsilon x + (1-\varepsilon)\tilde{x}$
7. $L^{(i)} \leftarrow D_w(\hat{x}) - D_w(x) + \lambda\left(\nabla_{\tilde{x}} D_w(\tilde{x})_2 - 1\right)^2$
8. **End for**
9. $w \leftarrow Adam\left(\nabla_w \frac{1}{m}\sum_{i=1}^{m} L^{(i)}, \alpha, \beta_1, \beta_2\right)$
10. **End for**
11. Sample a batch of latent variables $\left\{z^{(i)}\right\}_{i=1}^{m} \sim p(z)$.
12. $l^{(i)} \leftarrow -logD_{\theta_D}\left(G_{\theta_G}(x)\right) + \frac{1}{2}\mathrm{x}_i - G(y_i)^2$
13. $\theta \leftarrow \mathrm{Adam}\left(\nabla_w \frac{1}{m}\sum_{1}^{m} l^{(i)}, \alpha, \beta_1, \beta_2\right)$
14. End while

fusion and reconstruction of the denoised image are carried out. The denoised image features are fused by dimension reduction, and then the obtained image is connected with the noise-containing image across layers to obtain the denoised image with richer texture details. The generate network structure is shown in Figure 2-2, where K represents the convolution kernel size, N represents the feature output channel, and S represents the step size.

Figure 2. The structure of generate network



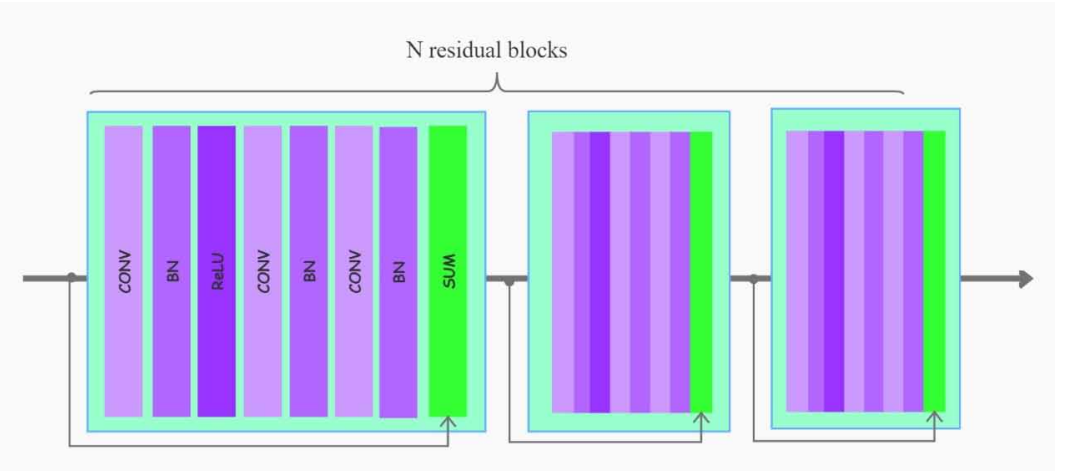
(1) Image Feature Extraction

The input of the generate network is the image with noise, so it is necessary to extract the image features and input the features into the next layer of the network for subsequent denoising. In the generate network, the first layer which is convolution layer for feature extraction, general neural network, the convolution kernel size is 1 x1, 3 x3, 5 x5, 7 x7 and 9 x9. In this paper, we choose a size of 3 x3 for the convolution kernel that accurately extracts the noise and features of the image. Since it is neither too small to extract too much redundant information, nor too big to filter out some important features. In the feature extraction layer, the ReLU activation function is selected as the activation function. In case of no standardized operation on data, the ReLU activation function can improve the convergence ability of the network.

(2) Image Feature Domain Denoising

We introduced the Residual Network (ResNet) (He, Zhang, Ren, & Sun, 2016) in feature domain denoising. The Residual Network structure designed in this paper refers to the Residual Network structure in an image super resolution algorithm (WDSR)(J. Yu et al., 2018) that improves efficiency and accuracy through the widened activation layer feature map, and a new Residual Network is constructed. The residual structure in the generate network is shown in Figure 2-3

Figure 3. The structure of Residual Network



In Figure 2-3, the residual block contains three convolutional layers. The convolution kernel size of the first convolutional layer is 1×1, the input channel is w_0, and the output channel is w_1. Then it performs the BN operation and activates the function using ReLU. The second convolutional layer has a convolution kernel size of 1×1 and an output channel of w_2. The last convolution kernel has the size of 3×3, and the output channel is w_3. Before using the activation function, in order to widen the width of the characteristic map to extract more features, we added one extension factor τ, making $w_1 = \tau \times w_0$. In the low-order convolution, the number of channels is reduced by setting $w_2 = l \times w_0$, where L is the compression factor; By setting the convolution kernel size as 3×3 and $w_3 = w_0$, the spatial orientation features were extracted. In the residual block, batch standardization is performed for each convolutional layer. In general, w_0 was set to the number of channels for feature extraction

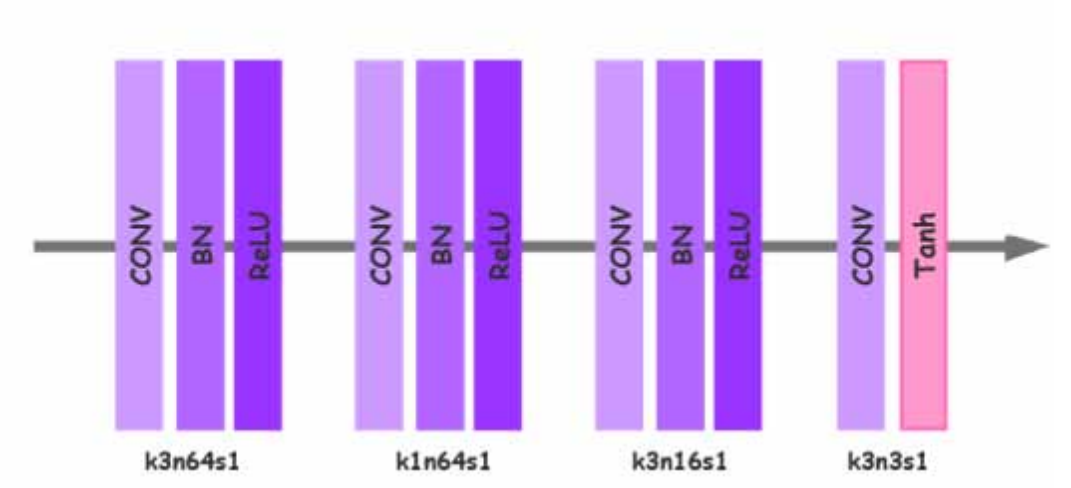
part output 64, expansion factor was 3, and compression factor L was 0.8. Experimental results show that when the residual block is 8, the image denoising effect is optimal.

After feature extraction of the image containing noise, Residual Network learning noise distribution and deepening of the network, the residual network learning fitting noise distribution can be used to obtain the residual image. Then, by cross-layer connection, the noise is removed from the noisy image distribution to obtain the preliminary denoised image. After layer by layer transmission through the network, the residual images obtained do not fully fit the noise distribution, which may not be clean enough for noise removal and may lose some image features in the denoising process. Therefore, it is necessary to further process and improve the denoising images to obtain the final denoised images.

(3) Image feature fusion and reconstruction

The Residual Network is used to denoise the noise-containing image in the feature domain to obtain the denoised image features, which also need to be screened, merged and reconstructed into denoised images. The feature fusion and reconstruction model is shown in Figure 2-4.

Figure 4. Feature fusion and reconstruction



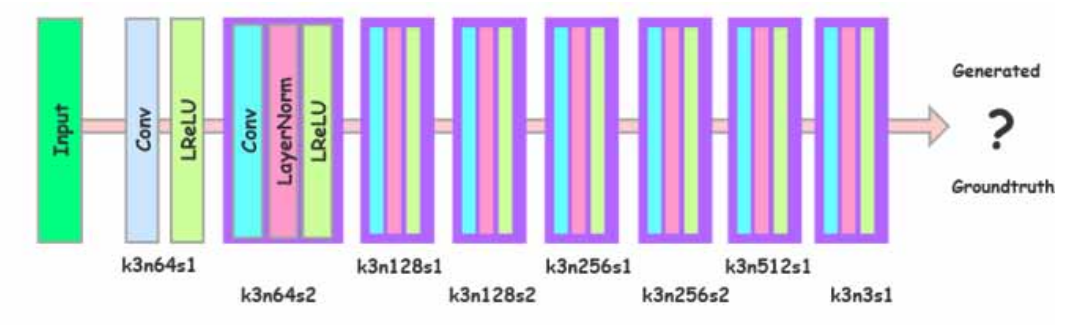
Feature fusion and reconstruction structures use 3x3 and 1x1 sized convolution kernels to extract the important features of the de-noised image and remove the redundant noise features. The output channels of the convolutional layer are reduced from 64 to 16 and then to 3 channels. The output channel is the same as the input image. In the first and second convolutional layer, 3×3 and 1×1 are respectively used to fully extract the features of the image after preliminary denoising. In the third convolutional layer, the output channel is reduced to 16 to filter the redundant noise features and retain the features of the original image. In the last layer, Tanh activation function is used to limit the output range to [-1, 1] and maintain the same pixel value as the input image. Then, through cross-layer connection and combining with the features of the original image, the feature information of the original image lost in the denoising process is supplemented to construct the final denoising image. Image feature fusion and reconstruction extracts image features and feature selection, cross connection layer with the input noise-contained image and mixes the texture detail features of the

original image, adds image loss details and builds the denoised image that is closer to the original image, retaining the color, texture details etc adequately.

2.2 Discriminant Network Structure

The discriminator is the counterpart of the generator. The discriminator is mainly used to determine whether the input image is real or generated by the generator. In this paper, a Convolutional Neural Network discriminator network is adopted. The discriminator network structure diagram refers to the discriminator structure in SRGAN(Lehtinen et al., 2018). The discriminator structure is shown in Figure 2-5.

Figure 5. The structure of discriminator network



Where, K represents the convolution kernel size, N represents the number of features, and S represents the step size of each convolutional layer. For the first six layers, the layer normalization(Ba, Kiros, & Hinton, 2016) is used in each convolution layer to process the data, and the Leaky ReLU activation function is used as the activation function, while the sigmoid activation function is removed in the last layer and the max-pooling is removed in the overall network.

Batch normalization is often used in deep neural network training and USES normalization of data to achieve consistent distribution. The data is usually taken as batch and standardized training is carried out for each batch to obtain mean variance, so that the data of each batch conforms to normal distribution and is distributed within a certain range in order to avoid gradient disappears effectively. Batch standardization depends on the batch size. When the block size is suitable, the effect is ideal. However, when the block size is too small, the mean variance obtained by the batch is different from the mean variance of the whole data, so it cannot replace the whole data. And batch standardization is not suitable for networks that input serialized data, such as Recurrent Neural Networks (RNN). This paper introduces the gradient penalty strategy instead of the weight clip method. In gradient penalty strategy, not for each batch apply gradient penalty but for each sample apply gradient penalty independently. So the discriminant model cannot use the batch of standardization, because it can cause the same batch of interdependencies between different samples, so using Layer Normalization instead of a batch of standardized operation. Layer normalization is to standardize the input of all neurons in a certain layer of the network. The mean variance of the input of neurons in the same layer is the same, and the input samples in different layers have different mean variance, so it does not depend on the size of the block.

Leaky ReLU function was selected as the activation function. Leaky ReLU function was improved upon the Leaky ReLU function. The ReLU function is calculated by formula 2-1:

$$ReLU\left(x\right)=\begin{cases}x,\ x>0\\0,\ x\le 0\end{cases}$$

The ReLU function sets the value to 0 when x<0 and to x when x>0. When the learning rate is too high, the RELU function is prone to "Dead ReLU" problem, that is, the ReLU activation function does not act on some neurons, resulting in the failure to update some relevant parameters. Unlike the RELU function, Leaky ReLU does not set the value to 0 when X <0, but assigns the value to a non-zero slope a, usually a=0.01. The RELU function is calculated by formula 2-2:

$$LReLU\left(x\right)=\begin{cases}x,\ x>0\\ax,\ x\le 0\end{cases}$$

The final layer removes the sigmoid activation function. In GAN, the task performed by the discriminator is to determine whether the input data is real or generated by the generator. It is a dichotomy task. Sigmoid activation function limits the output to the interval [0, 1], which meets the requirements of GAN discriminator task. The discriminator in WGAN is a regression task to approximate the Wasserstein distance between the real data distribution and the generated data distribution. The sigmoid activation function converges slowly at both ends of the function, and the gradient is close to 0. Therefore, the discriminator in WGAN needs to remove the sigmoid activation function.

In the convolutional layer, the convolution kernel of 3X3 is used to extract and fuse the texture features of the image. Output channel of the convolution layer is respectively 64, 128, 256, 512, and the last layer is three, step one and two respectively, when the step length is two, the input features of the network layer are doubled in length and width, under the action of convolution layer, network extracts the high-dimensional texture feature, increases output channels and characteristic number, then finally is three channel convolution layer, in order to do feature dimension reduction fusion.

2.3 Loss Function

In GAN, the generator and generator optimization process are antagonistic, and the optimization of generator depends on the training result of discriminator. Generator G is updated only once after discriminator D has been optimized K times. The optimization training process of generator G is to minimize the loss function.

A novel loss function is proposed in this study to ensure a good visual and quantitative results as well as good performance in discriminator D.

(1) Residual loss function

The noisy image is composed of noiseless original image and noise, and the expression is y=x+ v, where y represents the noisy image, x represents the original image, and V represents the noise. In some denoising algorithms, such as MLP(Ledig et al., 2017) using ordinary neural network and CSF(Ba et al., 2016) which is an effective image restoration algorithm based on contraction domain, learn to fit the distribution of the original image in the potential space, and finally a mapping function F(y)=x is obtained. However, compared with the noise distribution, the original image contains more information and features are more complex, so it is difficult to accurately calculate the distribution of the original image.

In this paper, in the generator, the Residual Network is used to learn and fit the noise distribution to get the residual image, namely $\mathrm{R}\left(\mathrm{y}\right)\approx \mathrm{v}$, where R(y) represents the residual image and v represents

the noise. After acquiring the residual image, the denoised image can be obtained simply by removing the residual image from the noise-containing image. The process is $\mathrm{x} = \mathrm{y} - \mathrm{R}\left(\mathrm{y}\right)$, where x is the original image and y is the noise-containing image. Therefore, the mean square error between the residual image and the noise estimation level of the image with noise can be used as the loss function, which is defined in formula 2-3, where N represents the training pair of the original image and the image with noise.

$$loss_{RES} = \frac{1}{2N}\sum_{i=1}^{N}\mathrm{R}\left(\mathrm{y}_i\right) - \left(\mathrm{y}_i - \mathrm{x}_i\right)^2 \tag{2.3}$$

Formula 2-3 can also be expressed as formula 2-4, which represents the mean square error between the original image and the generated denoised image, and $G\left(y_i\right)$ represents the denoised image generated by the generator when the input is y_i.

$$loss_{RES} = \frac{1}{2N}\sum_{i=1}^{N}\mathrm{x}_i - G\left(y_i\right)^2 \tag{2.4}$$

(2) Antagonistic loss function

The antagonistic loss function is used to measure the discriminant loss of the discriminator against the generator, which is described in Formula 2-5:

$$loss_{GEN} = \frac{1}{N}\sum_{i=1}^{N} - logD_{\theta_D}\left(G_{\theta_G}\left(x\right)\right) \tag{2.5}$$

To make the gradient look better, we used $-logD_{\theta_D}\left(G_{\theta_G}\left(x\right)\right)$ instead of $log\left(1 - D_{\theta_D}\left(G_{\theta_G}\left(x\right)\right)\right)$

Therefore, the generator loss function includes residual and antagonistic loss functions. Formula 2-6 defines the both losses:

$$G_{loss} = loss_{RES} + \lambda \bullet loss_{GEN} \tag{2.6}$$

Experiments show that $» = 0.5$ gives the best denoising effect for the network.

In GAN, the input of generator G is a noisy image, and the probability distribution of generated data is $P_G\left(x\right)$, The probability distribution of the real data is $\mathrm{P}_{data}\left(x\right)$. For discriminator D, the input is the probability distribution of generated data $P_G\left(x\right)$ and the probability distribution of real data $\mathrm{P}_{data}\left(x\right)$. The output is the probability of similarity between $P_G\left(x\right)$ and $\mathrm{P}_{data}\left(x\right)$. Confrontational training is used between discriminator D and generator G.

In optimized WGAN-GP (Improved Wasserstein GANs), the gradient penalty strategy was used to replace the weight pruning method. Compared with WGAN, WGAN-GP solves the problem of gradient disappearance and has a faster convergence rate.

In the gradient penalty strategy, $\mathrm{P}_{penalty}\left(x\right)$ is taken randomly from the distribution space between $\mathrm{P}_{data}\left(x\right)$ and $P_G\left(x\right)$. The gradient of the discriminator is $\nabla \mathrm{x} D\left(x\right)$ and discriminator loss function designed for this task is calculated by formula 2-7:

$$D_{loss} = E_{x\sim p_G}\left[D\left(x\right)\right] - E_{x\sim p_{data}}\left[D\left(x\right)\right] + \lambda \cdot E_{x\sim p_{penalty}}\left[\left(\nabla_x\left[D\left(x\right)\right]_2 - 1\right)^2\right] \tag{2.7}$$

According to the experimental results of WGAN-GP, the penalty coefficient is $\lambda = 10$.

3. EXPERIMENTAL ANALYSIS

3.1 Data Set

The data set USES Berkeley Segmentation Data set (BSD500). These images are extremely high resolution and rich in texture detail and edge features. Therefore, we employed BSD500 data set as the training data set. The BSD500 data set contains 500 images with a gray level of 255, including 400 images for training and 100 images for model testing. Before the training, the size of the data set was reset to 180x180, eight 64x64 image blocks were selected for each image to input into the network for training. Different levels of Gaussian noise were added to each image respectively. In the test set, Gaussian noise levels were used as σ=15 , σ=25 , σ=50 to test the model. Different levels of noise had different denoising effects.

3.2 Experimental Parameter Setting

In this paper, the size of the training batch is 128 and its power is 2. If the setting is too small, batch normalization (BN) processing will be carried out on the input, and the output will be affected by BN. On the contrary, the server memory is not large enough to process a large amount of data at one time, so the medium batch is chosen as the quantity size of each input data. The learning rate is set to 0.0001. Setting it too high will result in loss function shock, and the network will not converge easily. Similarly, the network training process will be sped up if the setting it too small. The optimizer uses the Adam optimizer. During the confrontation training, according to the experimental parameters set in WGAN-GP, the discriminator D was updated iteratively for 5 times, and the generator G was updated once.

3.3 Experimental Results and Analysis

It can be seen from the above two tables (Table 3-2, Table 3-3) that DnCNN and the algorithm presented in this paper are better than BM3D and EPLL in denoising no matter what the level of noise is. In this study, the algorithm and the PSNR exhibit higher DnCNN values (1 to 1.5dB) than BM3D, indicating a better noise reduction effect. Furthermore, the SSIM and the algorithm values for DnCNN are higher than 0.5, which allows better recovery of content information and texture details of denoised images. AS for the algorithm compared to DnCNN, the PSNR and SSIM values of de-noised images at all noise levels are higher than those of DnCNN, indicating excellent denoising effect and image texture detail recovery capability. While objective evaluation indexes such as PSNR and SSIM are used to evaluate the denoising level of each algorithm, the denoising ability of the algorithm proposed in this article is the best in PSNR and SSIM.

As can be seen from the tables (Table 3-4, Table 3-5), DnCNN which used Neural convolutional networks and the proposed algorithm are still better than BM3D in the denoising effect of color images. The PSNR and SSIM values of the proposed algorithm at noise level 50 are slightly lower than those of the DnCNN algorithm. In other noise levels of different degrees, the PSNR and SSIM values of

Table 2. Average PSNR of grayscale image denoising algorithm (unit: dB)

Denoising method Noise intensity	BM3D	EPLL	DnCNN	Ours
σ=15	31.52	30.70	32.72	**33.20**
σ=25	28.50	28.30	30.11	**30.45**
σ=50	24.94	24.60	26.49	**26.61**

Table 3. Mean value of SSIM for grayscale image denoising algorithm

Denoising method Noise intensity	BM3D	EPLL	DnCNN	Ours
σ=15	0.8690	0.8819	0.9282	**0.9336**
σ=25	0.7874	0.8067	0.8744	**0.8841**
σ=50	0.6612	0.6545	0.7355	**0.7514**

Table 4. Mean value of color image denoising algorithm PSNR (unit: dB)

Denoising method Noise intensity	BM3D	DnCNN	Ours
$\sigma = 15$	30.64	32.95	**33.01**
$\sigma = 25$	27.92	30.20	**30.25**
$\sigma = 35$	26.19	28.44	**28.45**

Table 5. Average value of color image denoising algorithm SSIM

Denoising method Noise intensity	BM3D	DnCNN	Ours
$\sigma = 15$	0.8854	0.9312	**0.9318**
$\sigma = 25$	0.8039	0.8831	**0.8833**
$\sigma = 35$	0.7340	0.8281	**0.8302**

the proposed algorithm are higher than those of other denoising algorithms. The proposed algorithm has better denoising effect at low and medium noise level compared to other denoising algorithms. Due to the excellent performance of human eyes, PSNR and SSIM values cannot truly represent the denoising effect of the final algorithm, so it is necessary to conduct further comprehensive evaluation combined with human eyes' perception of the denoising image.

Figure 3-6 shows the effect of different denoising algorithms on the image with the Gaussian noise level of 15.

Figure 6. (a-f) The denoising algorithm effects of grayscale images with noise level of 15

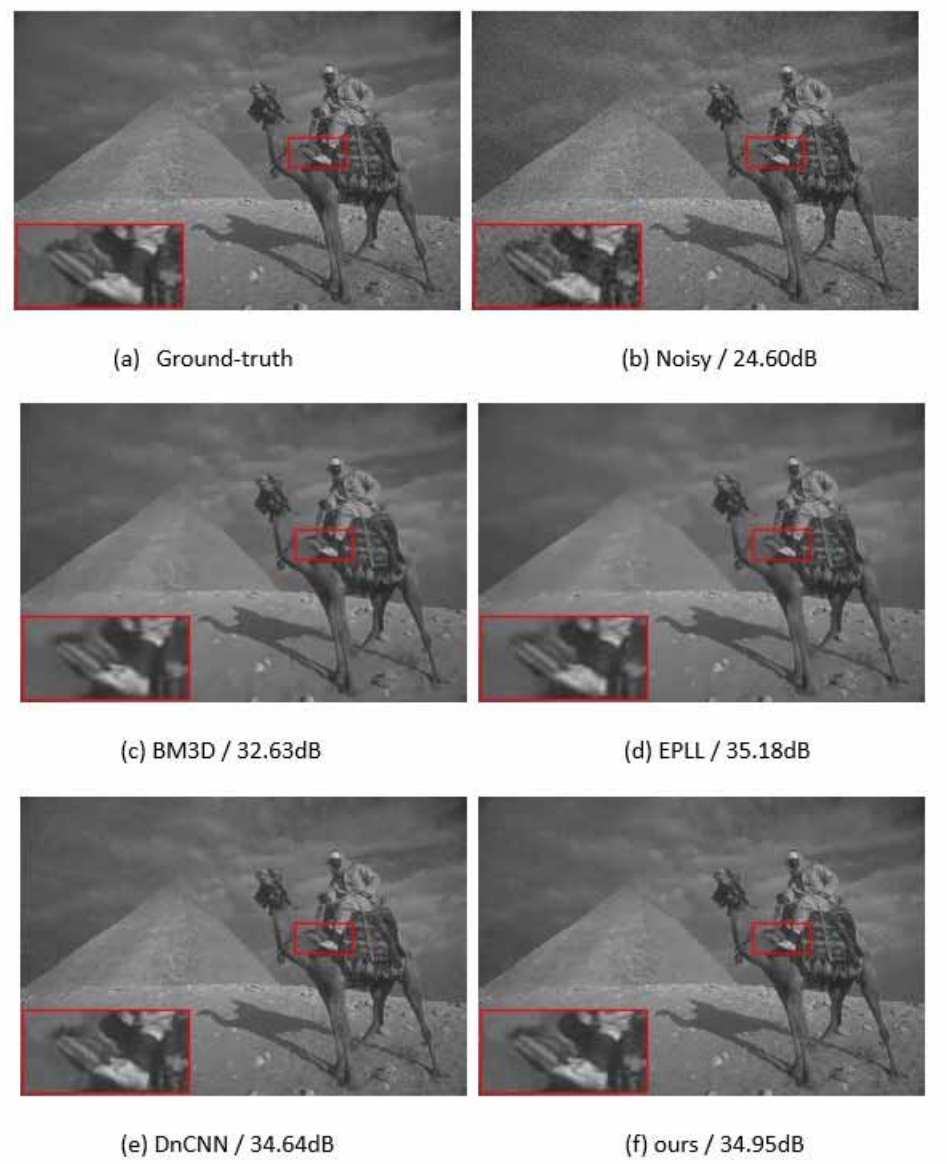


Figure 7. (a- f) Shows the effect of different denoising algorithms on the image with noise level of 25.

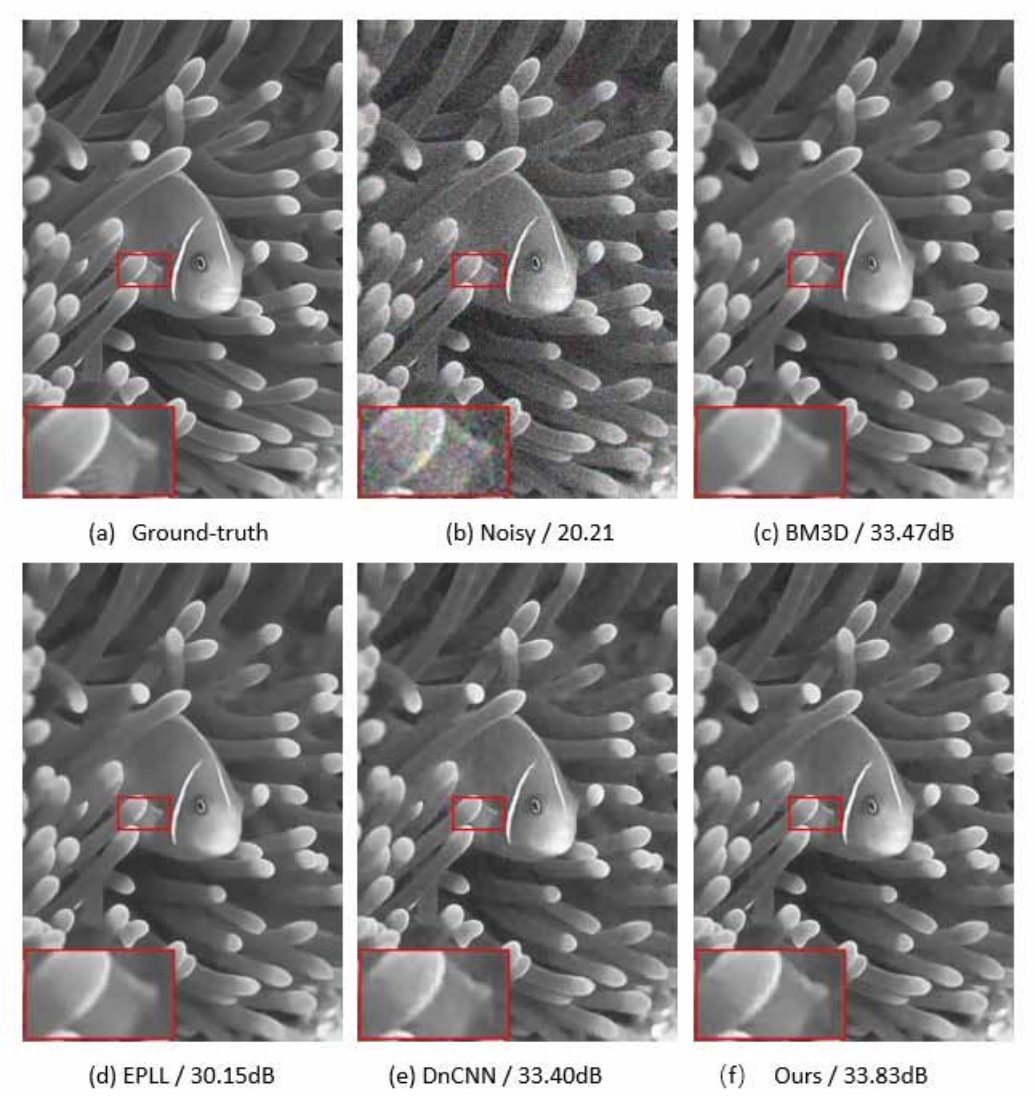


Figure 3-7 The denoising algorithm effects of grayscale images with noise level of 25

Figure 8. Shows the effect of different denoising algorithms on the image with noise level of 50.

Figure 3-9 Effect of color image denoising algorithms with noise level of 15

Figure 3-10 Effect of color image denoising algorithms with noise level of 25

Figure 3-11 Effect of color image denoising algorithms with noise level of 35

As shown in the comprehensive tables and figures, the denoising effect of the proposed algorithm is better than that of BM3D and DnCNN at low and medium noise levels. While the denoising effect is not much different from DnCNN at high noise level. The algorithm in this paper uses antagonistic training, which makes it easier to learn details than other algorithms. In addition to maintaining good denoising effect, the texture details of the image are better preserved, and the denoised image is closer to the original image.

Figure 8. The denoising algorithm effects of grayscale images with noise level of 50

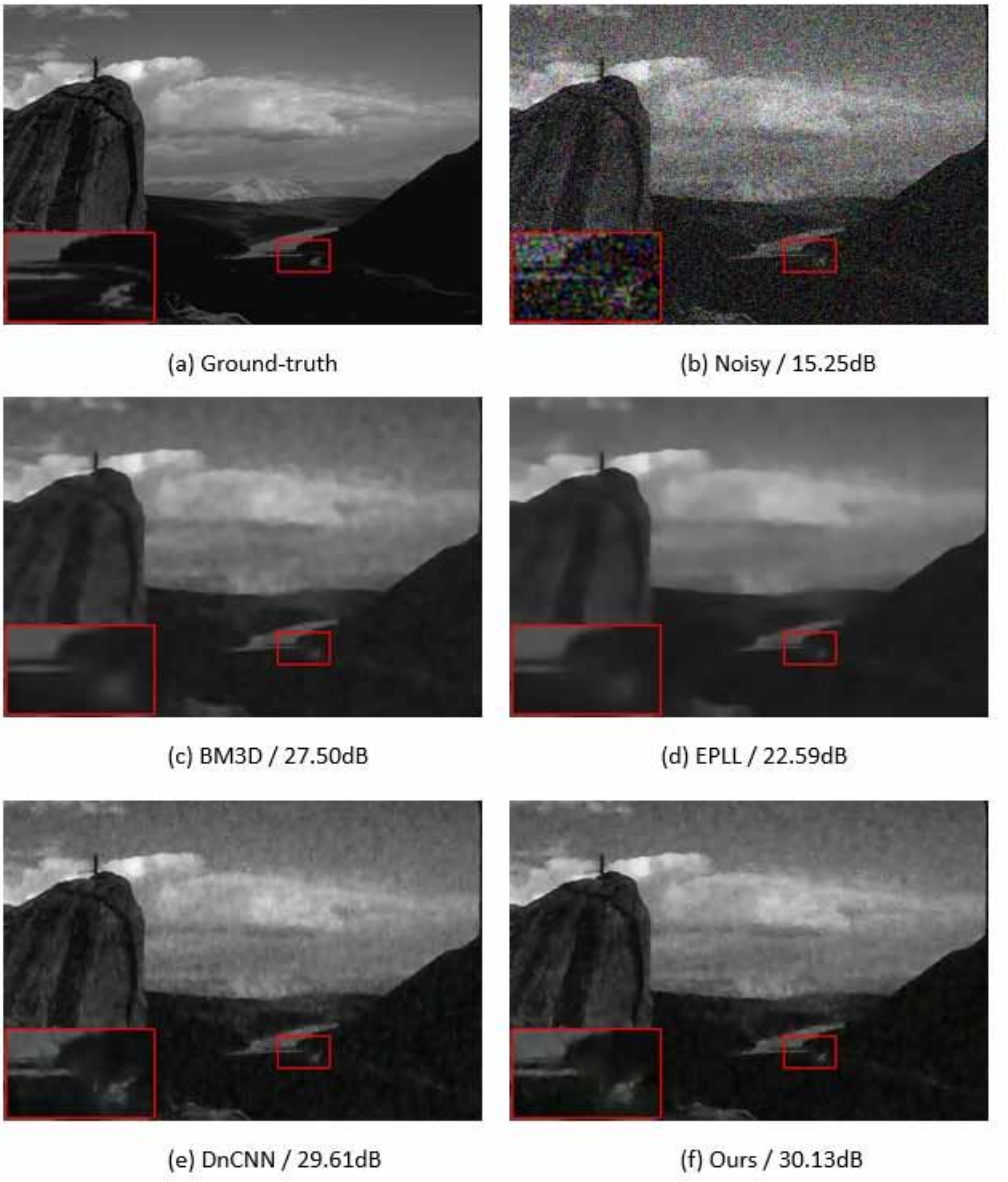


Figure 9. shows the effect of different denoising algorithms on the image with noise level of 15.

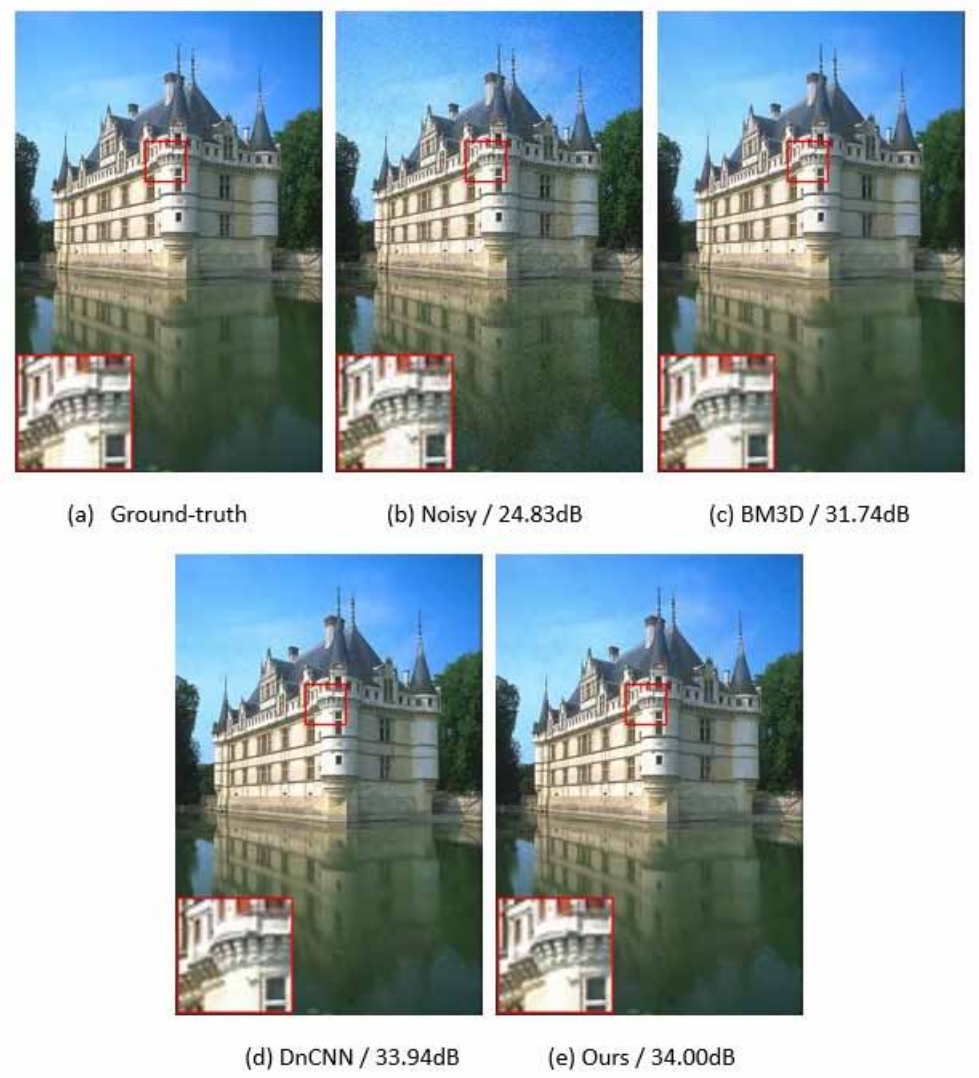


Figure 10. (a-e) Shows the effect of different denoising algorithms on the image with noise level of 25.

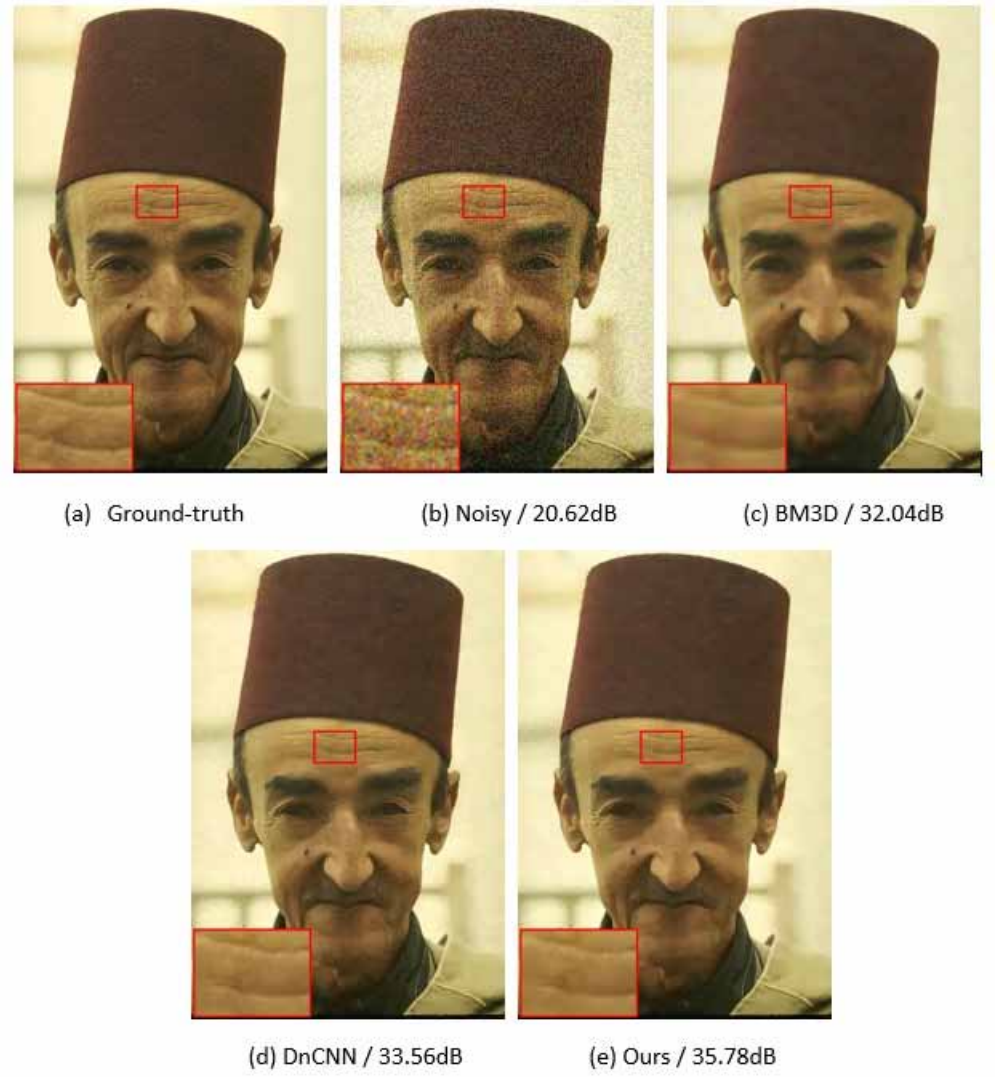
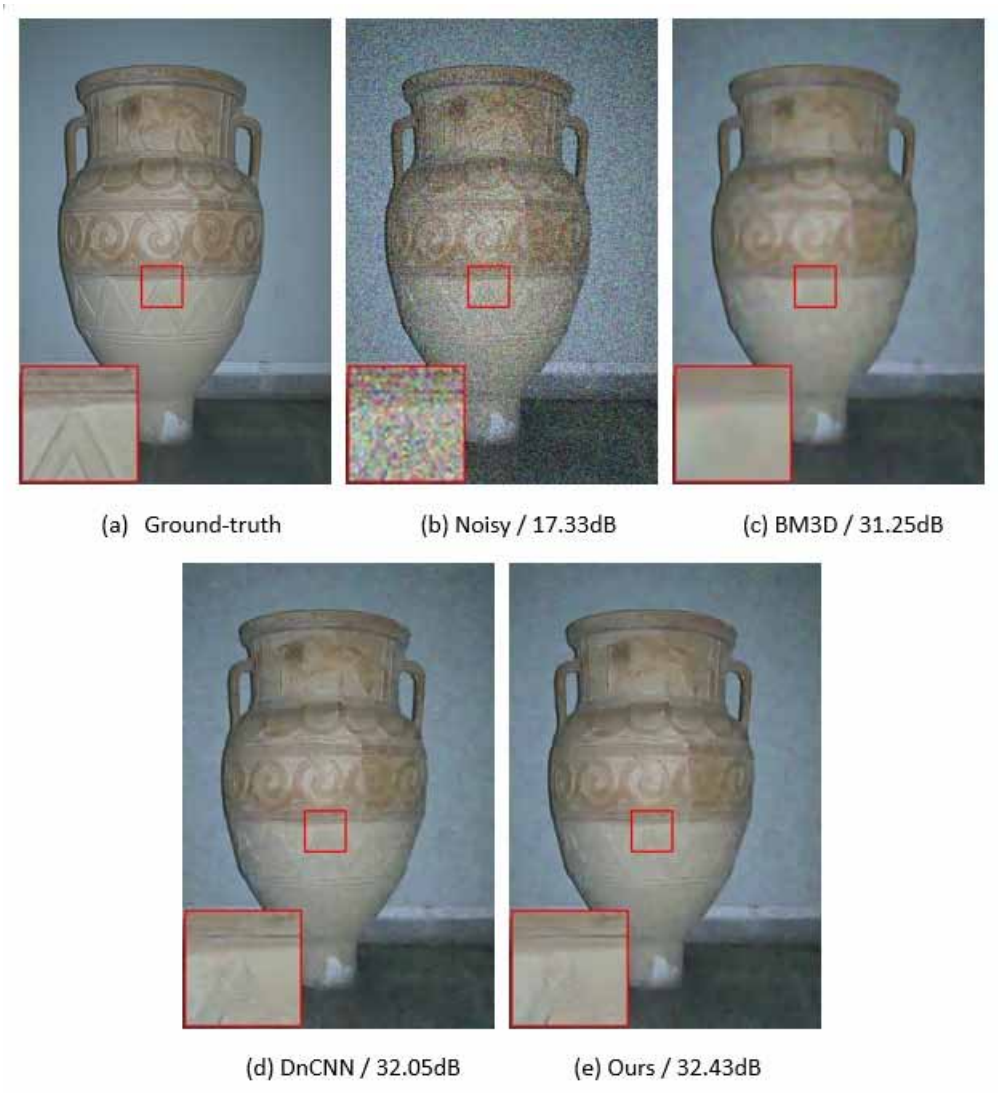


Figure 11. *(a-e)* **Shows the effect of different denoising algorithms on the image with noise level of 35.**



The denoising algorithm in this paper realizes the denoising of blind Gaussian noise. The denoising effect of this algorithm is better than that of other denoising algorithms such as BM3D and DnCNN. However, there are many kinds of noise in the image, and the model is complex (Harold C Burger, Christian J Schuler, & Stefan Harmeling, 2012), including raindrops, smoke and other noises, without specific models (Schmidt & Roth, 2014).

In the recent two years, more and more scholars have been doing research on image removal of raindrops and smoke, medical image denoising, as well as using Mars camera with low image distortion to contribute to Mars missions. What is more, the research of ultrasonic image denoising is more and more hot In the recent years(Chen et al., 2020).

4. CONCLUSION

This article proposes a network model based on the Wasserstein GAN, which can enhance denoising capability of images by the restoration of texture details and noise reductiom. A new residual network is proposed to improve the noise removal effect. The proposed algorithm has better denoising effect at low and medium noise level compared to other denoising algorithms. While the denoising effect is not much different from DnCNN at high noise level. The algorithm in this paper uses antagonistic training, which makes it easier to learn details than other algorithms. In addition to maintaining good denoising effect, the texture details of the image are better preserved, and the denoised image is closer to the original image. In the following work, the model will be trained to denoise other different noises. Image denoising can be used in many applications such as 360-degree panoramic image (Lee et al., 2020), Mars camera(Hsu, Shao, Tseng, & Huang, 2021) and moon image segmentation(Du et al., 2017).

ACKNOWLEDGMENT

This work was supported by a Special project of "Research on teaching reform and practice based on first-class curriculum construction" of the China Society of Higher Education (2020JXD01), a Special Project in the key field of "artificial intelligence" in Colleges and universities in Guangdong Province (2019KZDZX1027), Provincial Key platforms and major scientific research projects of Guangdong Universities (major scientific research projects - Characteristic Innovation) (2017KTSCX048), Scientific research project of Guangdong Bureau of traditional Chinese medicine (20191411), the University of Macau (File no. MYRG2019-00031-FBA), the University of Macau (File no. MYRG2019-00031-FBA), and Guangdong Provincial Industry College Construction Project (Artificial Intelligence Robot Education Industry College).

REFERENCES

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). *Layer normalization.* arXiv preprint arXiv:1607.06450.

Burger, H. C., Schuler, C. J., & Harmeling, S. (2012a). *Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds.* arXiv preprint arXiv:1211.1544.

Burger, H. C., Schuler, C. J., & Harmeling, S. (2012b). *Image denoising with multi-layer perceptrons, part 2: training trade-offs and analysis of their mechanisms.* arXiv preprint arXiv:1211.1552.

Burger, H. C., Schuler, C. J., & Harmeling, S. (2012). *Image denoising: Can plain neural networks compete with BM3D?* Paper presented at the 2012 IEEE conference on computer vision and pattern recognition. doi:10.1109/CVPR.2012.6247952

Chen, C.-J., Pai, T.-W., Hsu, H.-H., Lee, C.-H., Chen, K.-S., & Chen, Y.-C. (2020). Prediction of chronic kidney disease stages by renal ultrasound imaging. *Enterprise Information Systems*, *14*(2), 178–195. doi:10.1080/17517575.2019.1597386

Chen, C.-M., Zhang, S.-W., & Hsu, C.-Y. (2020). A sonography image processing system for tumour segmentation. *Enterprise Information Systems*, *14*(2), 159–177. doi:10.1080/17517575.2019.1575985

Chen, J., Chen, J., Chao, H., & Yang, M. (2018). Image blind denoising with generative adversarial network based noise modeling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. doi:10.1109/CVPR.2018.00333

Dai, W. (2018). *Research on text generation based on Generative Adversarial Networks* (Doctoral dissertation). Jilin University.

Du, X., Wang, H., Du, Y., Xu, L. D., Chaudhry, S., Bi, Z., Guo, R., Huang, Y., & Li, J. (2017). An industrial information integration approach to in-orbit spacecraft. *Enterprise Information Systems*, *11*(1), 86–104. doi:10.1080/17517575.2016.1173728

Fedus, W., Goodfellow, I., & Dai, A. M. (2018). *Maskgan: better text generation via filling in the_.* arXiv preprint arXiv:1801.07736.

Gomez, A. N., Huang, S., Zhang, I., Li, B. M., Osama, M., & Kaiser, L. (2018). *Unsupervised cipher cracking using discrete gans.* arXiv preprint arXiv:1801.04883.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. et al. (2014). Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, *3*, 2672–2680.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hsu, C.-Y., Shao, L.-J., Tseng, K.-K., & Huang, W.-T. (2021). Moon image segmentation with a new mixture histogram model. *Enterprise Information Systems*, *15*(8), 1046–1069. doi:10.1080/17517575.2019.1641627

Jiang, W., Xiao, L., & Shao-hui, M. (2019). Remote sensing image denoising based on convolutional neural network. *Microelectronics & Computer, 36*(8), 59-62, 67.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551. doi:10.1162/neco.1989.1.4.541

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., & Wang, Z. et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings of the IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2017.19

Lee, H., Lee, S., & Choi, O. (2020). Improved method on image stitching based on optical flow algorithm. *International Journal of Engineering Business Management*, *12*, 1847979020980928. doi:10.1177/1847979020980928

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., & Aila, T. (2018). *Noise2noise: Learning image restoration without clean data.* arXiv preprint arXiv:1803.04189.

Rosenblatt, F. (1959). A probabilistic model for visual perception. *Acta Psychologica*, *15*, 296–297. doi:10.1016/S0001-6918(59)80143-8

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. Academic Press.

Schmidt, U., & Roth, S. (2014). Shrinkage fields for effective image restoration. *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tseng, K.-K., Li, J., Chang, Y., Yung, K., Chan, C., & Hsu, C.-Y. (2019). A new architecture for simultaneous localization and mapping: An application of a planetary rover. *Enterprise Information Systems*, 1–17.

Xie, C., Wang, Y., Lin, Z., Zheng, Q., & Zhao, L. (2018). Monte Carlo Noise Removal Algorithm Based on Adversarial Generative Network. *Moshi Shibie yu Rengong Zhineng/Pattern Recognition and Artificial Intelligence, 31*(11), 1047-1060.

Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., & Huang, T. (2018). *Wide activation for efficient and accurate image super-resolution.* arXiv preprint arXiv:1808.08718.

Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. *Proceedings of the AAAI conference on artificial intelligence*.

Yuanqi, Z. (2018). *Color image denoising method based on Generative Adversarial Networks.* Dalian University of Technology.

Zhang, D., Dongru, H., Kang, L., & Zhang, W. (2019). The generative adversarial networks and its application in machine vision. *Enterprise Information Systems*, 1–21.

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, *26*(7), 3142–3155. doi:10.1109/TIP.2017.2662206 PMID:28166495

Zhang, Y., Gan, Z., & Carin, L. (2016). *Generating text via adversarial training.* Paper presented at the NIPS workshop on Adversarial Training.

Zhang, Y., Ma, S. Y., Zhang, X., Li, L., Ip, W. H., & Yung, K. L. (2020). EDGAN: Motion deblurring algorithm based on enhanced generative adversarial networks. *The Journal of Supercomputing*, *76*(11), 8922–8937. doi:10.1007/s11227-020-03189-y

Zhang, Y. Q. (2018). *Color image denoising method based on generative confrontation network* (MS thesis). Dalian University of Technology.

Xiufang Zou is a graduate student at South China Normal University. Her research interest includes artificial intelligence.

Dingju Zhu is a Professor at South China Normal University. His research interest includes artificial intelligence. He is the corresponding author of this article.*

Jun Huang is a graduate student at South China Normal University. Her research interest includes image processing.

Wei Lu is a graduate student at South China Normal University. His research interest includes artificial intelligence.

Xinchu Yao is a student at South China Normal University. Her research interest includes artificial intelligence.

Zhaotong Lian is a full professor at the Faculty of Business Administration, University of Macau. His research interests are Stochastic Models, Data Analytics, Supply Chain Management.