ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Design of clear recognition algorithm after network image noise removal in computer vision

Ke Dong[1,*]

[1] School of Computer and Artifical Intelligence, Xihang University, Xi'an City 710077, Shaanxi Province, China

SUMMARY: *In order to solve the problems of low accuracy and poor performance in the field of image recognition in massive image recognition, this paper proposes an image denoising algorithm based on improved CGAN after optimizing and improving the traditional conditional generative adversarial network (CGAN) algorithm on the basis of the neural network model by using the least-squares loss function and adopting techniques such as attention pyramid network. After that, the loss entropy of convolutional neural network is optimized, and the model network and shared feature parameters are constructed by inductive migration algorithm in the model training stage to achieve accurate recognition of denoised images. The results show that the denoised image repaired by this paper's algorithm has the best quality and structural similarity, and its PSNR and SSIM values are improved by 2.28dB and 0.124 respectively compared with those of the traditional CGAN model, which can significantly reduce the checkerboard effect phenomenon and improve the quality of the image. The improved convolutional neural network model achieves a correct rate of 99.47% for image recognition, and the recognition accuracy of 10 different types of images in the MNIST dataset is improved by 2.87% to 17.85% compared with the traditional convolutional neural network model. Comprehensive analysis shows that the model proposed in this paper has good application in both image denoising and recognition accuracy.*

KEYWORDS: *CGAN algorithm; inductive migration algorithm; convolutional neural network; image denoising; image recognition*

1 Introduction

With the rapid development of computer technology, computer vision has been widely used in industrial inspection, medical research, material science and other fields [1]. Due to the extremely high requirements on image quality in these fields, image noise has become one of the key factors limiting the performance of its applications. Image noise is mainly categorized into the following types: Gaussian noise, pretzel noise, Poisson noise, and scattering noise. Gaussian noise is usually caused by the thermal noise of electronic devices, which is manifested as random gray scale fluctuations in the image; pretzel noise is an obvious white or black spot triggered by the image sensor, transmission error, or bit error in the compression process; Poisson noise is mainly related to the random arrival process of photons, which is commonly found in the imaging under the low-light conditions; and the scattering noise is a peculiar phenomenon caused by the coherence of the light wave in optical imaging [2-5]. In addition, hardware noise, environmental disturbances and signal transmission errors are also common

*wingsing111@163.com
https://doi.org/10.65102/is2026246 Published: 30 April, 2026

sources of noise [6]. Together, these factors affect the image quality and accuracy of analysis results in computer vision applications.

Image noise not only affects image clarity and makes it difficult to recognize, but also may lead to incorrect interpretation of information, and in serious cases, even affects the decision-making output of the whole system [7, 8]. For example, in industrial inspection, noise may lead to misjudgment or omission of defect detection; in the field of life sciences, it may affect the correct understanding and analysis of microstructures such as cells. The presence of noise increases the computational complexity of image processing algorithms and requires more complex algorithms to correct the effects of noise, which not only consumes more computational resources, but also may reduce the processing speed of the system, thus affecting the real-time and stability of computer vision [9-12]. Therefore, effective noise removal and recognition techniques after denoising are the key to improve the performance of computer vision applications.

Image denoising techniques with the development of deep learning techniques shifted from traditional methods such as spatial domain filtering, frequency domain filtering, and model-based methods to methods such as Convolutional Neural Networks (CNN), Generative Adversarial Networks (GAN), and Transformer Architecture, etc., Kumar and Nachamai showed that, for medical images, the median filter was superior in removing peppered noise, Poisson noise, and blurring noise, while Wiener filter is better in suppressing speckle and Gaussian noise, while the performance is measured by a combination of file size, histogram and sharpness level [13]. Geng et al. used a fractional-order sparse representation model for image denoising by clustering image blocks, computing singular values, and adaptively learning fractional-order parameters for each cluster to construct a new sample space in the wavelet domain to obtain more accurate sparse coefficients with dictionaries [14]. However, this model-based denoising method has high computational complexity.Choi and Jeong proposed a synthetic aperture radar (SAR) image denoising algorithm based on discrete wavelet transform, which transforms multiplicative noise into additive noise and filters it after multi-stage processing such as noise-canceling anisotropic diffusion, soft-thresholding with an improved bootstrap filter, which can better retain image edge features while significantly suppressing speckle noise and has a low computational complexity [15]. However, this kind of frequency domain filtering denoising method has limited effect on the processing of non-periodic noise.

Among the deep learning based denoising methods, Singh et al. combined method noise with CNN for efficiently removing Gaussian additive white noise and preserving details in CT images [16]. Xie et al. designed a DnCNN-based 2D image denoising method for log-transformed phase images for laser imaging, overcoming data scarcity through neural network training, effectively suppressing the Goulburn distribution noise, and significantly decreasing the cloud-to-grid distance of the denoised point cloud, which improves the accuracy and reliability of 3D reconstruction on complex microscale targets [17]. Wang et al. proposed an infrared image hybrid noise removal algorithm using improved CycleGAN, which effectively improves denoising and detail retention, and the algorithm outperforms similar methods on both synthetic and real noisy images [18]. Simon and Kapileswar constructed a two-branch GAN system, which combines an attention feature aggregation module and a noise model unit through dual paths of texture and edge recovery, and introduces the radiologist consistency loss for optimization, effectively suppresses Poisson-Gaussian noise, and improves the quality of the image with a good generalization ability at the same time [19]. Zhang et al. utilized a self-encoder for spectral domain denoising of near-infrared hyperspectral images, which outperformed traditional smoothing and wavelet methods [20]. Yao et al. reported a Transformer image denoising network called DenSformer, which achieved excellent performance on both synthetic and real noisy data, and the model parameters could be reduced

by up to 40% [21]. Deep learning based image denoising methods are more effective, but the computational complexity remains a challenge and the generalization ability of the model needs to be improved.

For the study of image clarity recognition after denoising, Liang et al. developed a fast defogging algorithm based on bilateral hybrid filtering, which effectively enhances the image denoising and clarity, significantly reduces the computational complexity, and the recognition rate of the image after defogging can be up to 98.8%, which takes into account the real-time and reliability [22]. Berrimi et al. separated the large and small features by motion compensation and adaptive thresholding, and used kernel principal component analysis to reconstruct the details and anisotropic diffusion to smooth the homogeneous region, respectively, which effectively retained the key details of the recognition, and improved the quality of the image restoration and the recognition performance under different noise and blurring conditions [23]. Gonwirat and Surinta fused robust deblurring GAN and CNN to build an image denoising architecture for dealing with multiple noise problems in handwritten character images, converting noisy characters into clear images and completing the recognition at the same time, achieving a high recognition accuracy of more than 98% [24].

This paper firstly introduces the basic working principle of neural network model and its structural components, and optimizes the problem of network instability during the training process of neural network model with least squares loss function. After that, the dense module is added for feature conversion, and then APNet decoder is introduced to reduce the computation of the network; the generator is motivated to produce higher quality images by introducing PatchGAN discriminator network. Further, an improved convolutional neural network image recognition algorithm is proposed, which uses its advantages of sharing weights, self-learning to extract classification features, and completing network training, and combines with the inductive migration algorithm to construct the model network and shared feature parameters to improve the generalization of the model as well as the accuracy of classification. Finally, the MNIST dataset is used as an example to analyze the application effect of the research in this paper in image denoising and recognition.

2 Machine vision-based network image denoising and recognition models

2.1 Neural network modeling

2.1.1 M-P model

The M-P model is the first simplified model built according to the structure and working principle of human neurons.The M-P model consists of several input nodes $\{x_i \mid i=1,...,n\}$ corresponding to an output node y, the input x_i is multiplied by the corresponding connection weight w_i and then summed to get the computation result, and then the computation result is compared with the set threshold to output the result y. The value of the threshold h is set in advance, if the calculation result is greater than the threshold h, then output 1, and vice versa output 0. i.e.:

$$y = f\left(\sum_{i=1}^{n} w_i x_i - h\right) \tag{1}$$

Eq:

y - network output;

x_i - network input;

w_i - input connection weights;

h - set comparison threshold.

2.1.2 Perceptron

When training samples for the M-P model perceptron, the desired output value needs to be set in advance and adjusted repeatedly according to the difference between the actual output and the desired output to complete the updating of the weight parameters, and this parameter updating method is called error correction learning. The specific operation process of error correction learning is:

$$w_{i+1} = w_i + \alpha(r - y)x_i \tag{2}$$

$$h_{i+1} = h_i - \alpha\left(r - y\right) \tag{3}$$

α - the parameter that determines the value of the connection weight adjustment;

x_i - input nodes of the network;

r - the output expectation of the network;

y - the output units of the network;

w_i - input connection weights at the i th adjustment;

h_i - comparison threshold at the i th adjustment;

Where α is equivalent to the learning rate and is mainly used to constrain the speed of correction. The perceptron uses random numbers to initialize the parameters, so the weight parameters may not be the same for each training. The weight adjustment method in the perceptron model is its innovation, the specific ideas are as follows:

(1) If the actual output y is equal to the desired output r, there is no need to adjust the values of w_i and h;

(2) If the actual output y is not equal to the desired output r, adjust the values of w_i and h to account for two further cases:

(a) The actual output $y = 0$ and the desired output $r = 1$ is inactive, the connection weight w_i should be decreased for h and increased for $x_i = 1$, and the connection weight for $x_i = 0$ should remain unchanged;

(b) The actual output $y = 1$ and the desired output $r = 0$ is an over-activated state, which should increase h, decrease the connection weight w_i for $x_i = 1$, and leave the connection weight for $x_i = 0$ unchanged.

2.1.3 Multilayer Perceptron

In order to solve complex classification problems that are linearly indistinguishable, multilayer perceptron (MLP) models have been proposed.MLP models usually consist of an input layer, an intermediate layer, and an output layer.The MLP model is a network that propagates the input values forward, and is also known as feed-forward or forward-propagation network because of its ability to estimate the error of the previous layer by the error value of the current layer.

The data in the MLP model is transmitted forward in one direction, from the input layer,

propagated through the intermediate layers, and finally output by the output layer, there is error propagation during the transmission of data in each layer, and the model gets different error signals in each layer. The gradient descent method can adjust the weights w, which need to be adjusted repeatedly with the help of the error between the actual output and the desired output to minimize the error and finally get the optimal connection weights w^{opt}.

The error is usually calculated with the help of an error function, the most commonly used is the least squares error function, which is obtained by operating on the desired output r and the actual output of the network y to get the error E. The closer E tends to 0, indicating that the results of the training and the more the desired value is trained, the better the training effect is. Thus, the training process of the MLP model is equivalent to ensuring that the least squares error function can be gradually approximated to 0 through the continuous adjustment of the connection weights w. The gradient at a given point can be obtained by derivation of the error function, and the value of the connection weight adjustment Δw can be summarized as:

$$\Delta w = -\eta \frac{\partial E}{\partial w} \tag{4}$$

η - learning rate, weights values adjusted according to the degree of deviation from the error;

Δw - weight adjustment change value;

E - least squares error function;

w - the value of connection weights.

When adjusting the connection weights of the MLP model through the error back-propagation algorithm, the selection of the activation function also needs to be considered.The M-P model uses the STEP function as the activation function, which can only output 0 or 1, and the output is not continuous. In order to enable continuous propagation of error, Rumelhart et al. proposed to use the derivable function sigmoid as the activation function.

2.2 Image denoising algorithm based on improved conditional generative adversarial networks

2.2.1 CGAN Principles

The principle of Conditional Generative Adversarial Network (CGAN) is shown in Fig. 1. Random noise z and conditional variable y are used as inputs to generator G, which generates image $fake\_x$ after processing by the generator. Subsequently the false image $fake\_x$ and conditional variable y generated by the generator will be input to the discriminator D, which will determine whether the input image is a real image. Alternatively, the real image $real\_x$ and the condition variable y will be input into the discriminator D, and the discriminator will determine whether the input image is a real image or not.

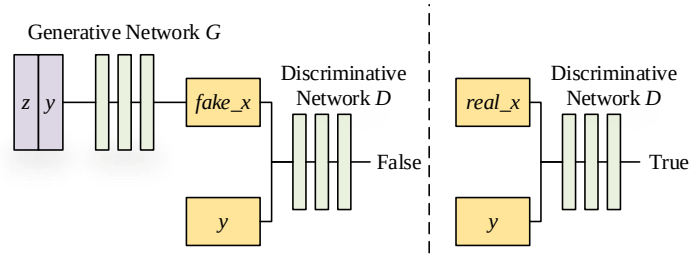

Figure 1: CGAN schematic diagram

The objective function of CGAN adds conditional information y to GAN and is calculated as follows:

$$L_{CGAN}(G,D)=E_{x,y}\left[\lg D(x,y)\right]+E_{x,z}\left[\lg(1-D(x,G(x,z)))\right] \tag{5}$$

2.2.2 CGAN improvements

(1) Least squares loss function

The adversarial loss function of the original CGAN network follows the sigmoid cross-entropy loss function adopted by GAN. In this paper, the least squares loss function is used to replace the sigmoid cross-entropy loss function as the adversarial loss. In addition, using the least squares loss function as the adversarial loss also makes the training process of the network more stable. The least squares loss effectively solves the problem of vanishing gradient and improves the stability of the network by updating and optimizing the generator. Sigmoid cross-entropy loss function and least squares loss function, the saturation state of sigmoid cross-entropy loss is very long, while the saturation state of the least squares loss is only one point.

After changing to the least squares loss function to calculate the adversarial loss of CGAN, the formula for the CGAN objective loss function is as follows:

$$L_{CGAN}(G,D)=E_{x,y}\left[(D(x,y)-1)^2\right]+E_{x,z}\left[D(x,G(x,z))^2\right] \tag{6}$$

(2) Encoder-Decoder Structure (DenseNet)

The generator network in this paper adopts the encoder-decoder structure, and combines the idea of CycleGAN to add a feature conversion module between the encoder and decoder. However, CycleGAN adopts the residual module as the feature conversion module, while this paper adopts the dense module for feature conversion, the specific reasons are analyzed as follows.

In the traditional convolutional neural network, the output X_i of the i th layer is calculated as follows:

$$X_i=H_i\left(X_{i-1}\right) \tag{7}$$

where H_i denotes the nonlinear transformation of the i th layer, which is a combinatorial operation that may contain the activation function ReLU or LReLU, the normalization and pooling layers, etc. The X_{i-1} is the input of the i th layer, i.e., the output of the $i-1$ th layer.

ResNet enhances the training of deep networks by adding “shortcut” connections, which allow the input information to be passed directly to the output without intermediate processing, so that the input information is not corrupted during intermediate transmissions, and the learning process of the network becomes simpler. For ResNet, the output of the i th layer has:

$$X_i=H_i\left(X_{i-1}\right)+X_{i-1} \tag{8}$$

The gradient of ResNet can flow directly from the layers behind to the layers in front of it through a constant function.DenseNet differs from ResNet in that the output of ResNet is constantly accumulated, while DenseNet passes the input of the layer in front of it directly to the output of the layer behind it, establishing a dense connection between the layer in front of it and all the layers behind it. For DenseNet, the output X_i of the i th layer is:

$$X_i = H_i\left([X_0, X_1, X_2, \cdots, X_{i-1}]\right) \tag{9}$$

where $[X_0, X_1, X_2, \cdots, X_{i-1}]$ is the feature map obtained by connecting the outputs of the X_0 th layer to the X_{i-1} th layer.

(3) Attention Pyramid Network (APNet)

In this paper, we use a channel attention model implemented with APNet, which gives more weight to the channels with higher response, thus making the model more attentive to the salient features in the image. The channel attention module uses SE layer which consists of a global average pooling layer and two consecutive fully connected layers.

Given a feature mapping X , the channel attention is computed as follows:

$$A^C = \sigma\left(W_2^c \operatorname{Re} LU\left(W_1^c Pool_{avg}(X)\right)\right) \tag{10}$$

where W_1^c and W_2^c are the parameters of the two fully-connected layers, $Pool_{avg}$ is the average pooling layer, and the final channel attention A^C will be applied to the original feature mapping by channel multiplication.

(4) PatchGAN Discriminator

The discriminator of CGAN outputs a scalar between 0 and 1, which represents the probability that the image generated by the generator is a real image. PatchGAN, on the other hand, outputs an N×N matrix M. Each element of this matrix represents a patch, corresponding to a sensory field of the image, and finally the average of each patch is taken to represent the probability that the image generated by the generator is a real image. Therefore, in order to generate denoised images with higher quality, PatchGAN is used as the discriminator network in this paper.

2.2.3 Improving the network structure of the model

The generator network used in the experiments of this paper adopts the encoding-feature transformation-decoding structure, which contains, in turn, the BN layer, the activation function ReLU, the convolutional layer consisting of 1024 convolutional kernels of size 1×1, the BN layer, the activation function ReLU, and the convolutional layer consisting of 256 convolutional kernels of size 3×3.

The encoder in the generator contains, in order, 64 convolutional layers with convolutional kernels of size 7 × 7 and step size 1, 128 convolutional layers with convolutional kernels of size 3 × 3 and step size 2, 256 convolutional layers with convolutional kernels of size 3 × 3 and step size 2, the feature converter contains nine sense modules, and the decoder is a channel attention network implemented in APNet. All convolutional layers in the generator contain BN layers and ReLU activation functions.

The discriminator in this paper is PatchGAN.This discriminator network contains 5 convolutional layers.The image generated by the generator and the input real image will first be spliced and then after three times downsampling, and then after convolutional layers, BN layer and ReLU activation function etc., the image will be divided into N×N regions and these regions will be scored individually to form a N×N score matrix and finally through the sigmoid outputs the average of this score matrix as a reflection of the similarity between the generated image and the real image.

2.3 Improved image recognition algorithm with convolutional neural network

2.3.1 Convolutional Neural Networks

The core principle of the convolutional neural network structure is to use convolution and pooling operations to construct supervised learning style training nets, which are classified into two types of training propagation for the forward training as well as the backward training propagation, which are combined with each other as inputs, viz:

$$s(i,j)=\sum_{m=0}^{M}\sum_{n=0}^{N}\left(w_{m,n}x_{i+m}+w_b\right) \tag{11}$$

The feature vector matrix is identified using $s(i,j)$, the 2D input matrix is labeled using the 2D matrix $M\times N$, $w_{m,n}$ represents the convolution kernel matrix of size $M\times N$, and w_b represents the offset vector. The loss function used in the network model construction process is calculated using loss entropy:

$$L=\frac{1}{n}\sum\nolimits_x\left[y\ln(a)+(1-y)\ln(1-a)\right] \tag{12}$$

The formula uses L to represent the loss, x represents the input sample data, a is the result of the computation, y represents the labeled term value, and n is the total number of samples.

2.3.2 Generalized Migration Model Training

(1) Migration model training process

In this paper, a parameter-based inductive migration algorithm is used for model training. The migration model training process is as follows:

Step 1:Carry out the loading work with the training model, get the tensor value of the input data as well as calculate the tensor value of the existence of bottleneck layer.

Step 2:Complete the CNN network initialization input work, initialize the learning rate, the number of training times, the number of batches and the Dropout value of each layer of the network.

Step 3:Extract training image features. The sample image to be classified is used as input to the CNN model, and the feed-forward neural network model is used to calculate the tensor value of the bottleneck layer again.

Step 4:Complete the classification task by classifying the fully connected layer for the target task and constructing the loss layer using the cross-entropy loss function.

Step 5:Introduce the sample set of data images to be tested for feature classification and obtain the accuracy of the model.

(2) Group normalization

The normalization process is completed after first performing channel grouping and calculating the variance and mean within the group, using the standard deviation and its mean corresponding to the set of image pixel points S_i:

$$S_i=\left\{k\,|k_N=i_N,\frac{k_c}{C/G}=\frac{i_c}{C/G}\right\} \tag{13}$$

i and k are representing the image index coordinates which are in the same channel, N is the number of samples and C/G represents the number of channels in each group.

The mean and variance corresponding to the pixel points are calculated as:

$$\mu_i = \frac{1}{m}\sum_{k\in S_i} x_k \tag{14}$$

$$\sigma_i = \sqrt{\frac{1}{m}\sum_{k\in S_i}(x_k - \mu_i)^2 + \varepsilon} \tag{15}$$

The x_k in the formula represents the pixel points that need to be calculated, ε represents the parameter that maintains the stability of the data, and m represents the length value of S_i.

After the normalization process is completed for the pixels within each channel group using Eq. After the normalization process is completed for the pixels within each channel group using Eq. Since the learned feature distributions cannot be corrupted, normalization of the data is required, which is accomplished by reconstruction and scale offset changes:

$$\hat{x}_i = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \varepsilon}} \tag{16}$$

$$y_i = \gamma x_i + \beta \tag{17}$$

x represents the pixels to be grouped for normalization, $\hat{x}_i$ represents the feature distribution after completing the normalization process, and γ as well as β represent the reconfigurable learning parameters.

(3) Optimize the loss function

Improve the cross-entropy function for the deep mining of positive samples, increase the weight and occurrence probability of positive samples, and further improve the accuracy of image recognition:

$$L = -y\log y' - (1-y') = \begin{cases} -\log y', & y=1 \\ -\log(1-y'), & y=0 \end{cases} \tag{18}$$

Since the positive samples of this loss function show negative correlation with the loss function, and the opposite negative samples y' are positively correlated with the loss function, so it can not adapt to a large number of sample data, based on this, the loss value of easy to categorize samples is reduced by adding the adjustment parameters γ and α, balancing the imbalance in the distribution of the existence of positive and negative samples:

$$L = \begin{cases} -\alpha(1-y)^{\gamma}\log y', & y=1 \\ -\alpha(1-\alpha)y^{\gamma} - \log(1-y'), & y=0 \end{cases} \tag{19}$$

where the experiment confirms that the adjustment parameter γ takes the value of 2 and α takes the value of 0.25.

The overall process of image recognition is shown in Figure 2. The core of the algorithm is

to first remove the classification layer of the pre-training model, and then freeze the previous layer to extract features, followed by group normalization to complete the construction of the network, adjust the sample training weights, complete the optimization of the loss layer, and finally complete the target task classification.

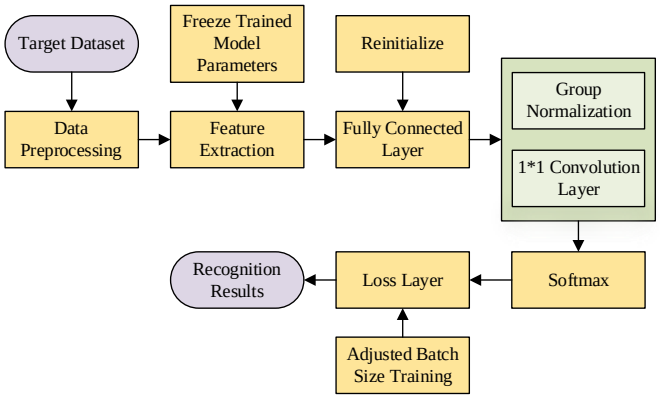

Figure 2: Image recognition workflow

3 Image noise removal effect and recognition results analysis

3.1 Analysis of image denoising results based on generative adversarial network

3.1.1 Experimental setup

(1) System environment construction

The model in this paper is based on Tensorflow deep learning framework, Tensorflow has a high degree of flexibility, portability, its internal implementation of the automatic derivatives for a variety of given objective function, support for Python, C++ and other languages and highly optimized performance.

(2) Experimental data set

The basic experimental environment of this paper is Linux operating system, and the development language is Python. The experimental data of this paper is MNIST dataset, and the experimental content mainly focuses on the experiments of the improved CGAN model. The dataset has 6,000 images of each type, and the sheets are color images of 32×32 pixel size.

(3) Image denoising quality evaluation metrics

In this paper, PSNR and SSIM are used as objective evaluation indexes to calculate the objective data of image quality and evaluate them. The formulas are as follows:

The peak signal-to-noise ratio (PSNR) formula is:

$$PSNR = 10\log\left(\frac{L^2}{MSE}\right) \tag{20}$$

The expression for structural similarity (SSIM) is:

$$SSIM\left(x,y\right) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{21}$$

The formula:

μ_x and μ_y - the mean of the image x;

σ_x^2 and σ_y^2 - the variance of the image;

σ_{xy} - the covariance of the image and;

c_1 and c_2 - constants used to maintain stability, with values equal to $(k_1 L)^2$, $k_1 = 0.01, k_2 = 0.03$;

L - dynamic range of pixel values.

3.1.2 Image denoising result analysis

In order to show the effect of this paper's model on image denoising under different noise variance, 1 image from MNIST dataset is taken as standard, and Gaussian noise of 10,20,30,40,50 is added respectively, and denoising restoration is carried out using this paper's model.

The PSNR and SSIM of this paper's model on MNIST under different noises are shown in Table 1. The results show that under the condition of noise of 10dB, the value of PSNR, an objective evaluation index, can be maintained above 30dB, and overall better denoising results are achieved. Even under the conditions of 40dB and 50dB noise environment, the denoising effect of this paper's model is still significant, and the value of PSNR is still able to maintain a medium-high level of more than 25dB although it decreases, which reflects the strong generalization ability of this paper's model.

Table 1: PSNR and SSIM of the proposed model on MNIST under different noises

Test sample	Gaussian noise value									
	10		20		30		40		50	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	35.28	0.8	30.46	0.86	30.75	0.81	26.15	0.82	27.38	0.75
2	33.24	0.85	33.71	0.93	29.1	0.85	26.22	0.76	33.01	0.83
3	33.14	0.85	32.14	0.89	29.56	0.8	28.25	0.82	28.29	0.68
4	32.44	0.92	33.13	0.91	31.57	0.85	28.24	0.78	27.53	0.81
5	34.42	0.87	30.32	0.95	27.52	0.78	26.69	0.84	29.24	0.83
6	34.42	0.89	32.98	0.93	31.5	0.7	27.71	0.81	29.2	0.8
7	34.95	0.82	33.28	0.84	27.97	0.84	28.29	0.74	27.28	0.83
8	32.4	0.91	31.76	0.93	31.67	0.87	29.78	0.71	30.26	0.74
9	31.75	0.95	34.25	0.89	30.54	0.93	25.6	0.86	28.11	0.77
10	36.69	0.87	31.2	0.85	29.86	0.59	27.69	0.82	30.88	0.79
11	34.02	0.97	30.7	0.88	28.01	0.93	28.67	0.8	27.86	0.82
12	31.81	0.94	32.96	0.81	28.91	0.87	27.12	0.83	28.49	0.8
13	34.24	0.85	32.5	0.93	30.98	0.9	28.32	0.92	27.78	0.82
14	32.51	0.79	32.42	0.87	33.71	0.92	26.83	0.76	30.43	0.82
15	33.08	0.95	28.63	0.84	30.2	0.96	27.58	0.91	31.11	0.77
16	33.61	0.95	31.6	0.91	28.27	0.83	29.18	0.81	28.37	0.82
17	33.28	1.06	32.51	0.91	28.5	0.95	29.12	0.75	29.34	0.83
18	36.07	0.96	31.61	0.88	28.61	0.58	29.51	0.72	27.36	0.88
19	33.85	0.8	34.54	0.97	29.61	0.79	28.59	0.75	28.88	0.83
20	33.62	0.89	28.49	0.89	31.76	0.76	29.76	0.89	26.3	0.82
21	33.7	0.96	32.22	0.91	26.46	0.64	27.59	0.76	28.47	0.84
22	33.16	0.98	35.13	0.93	34.29	0.92	29.26	0.88	27.9	0.84
23	37.22	0.86	29.67	0.87	30.25	0.89	28.04	0.77	27.93	0.75
24	31.97	0.99	32.27	0.98	31.12	0.72	29.35	0.78	29.78	0.81

Taking the Gaussian noise value of 20 as an example, the PSNR and SSIM of this paper's algorithm on MNIST are shown in Fig. 3.The model in this paper denoises 24 images, and the quality of the images obtained can all be maintained in a relatively stable state. Its PSNR is maintained between 28.49dB~35.13dB, and SSIM is maintained between 0.81~0.98, and the restoration effect of the model is stable, and will not have a large deviation because of the image content.

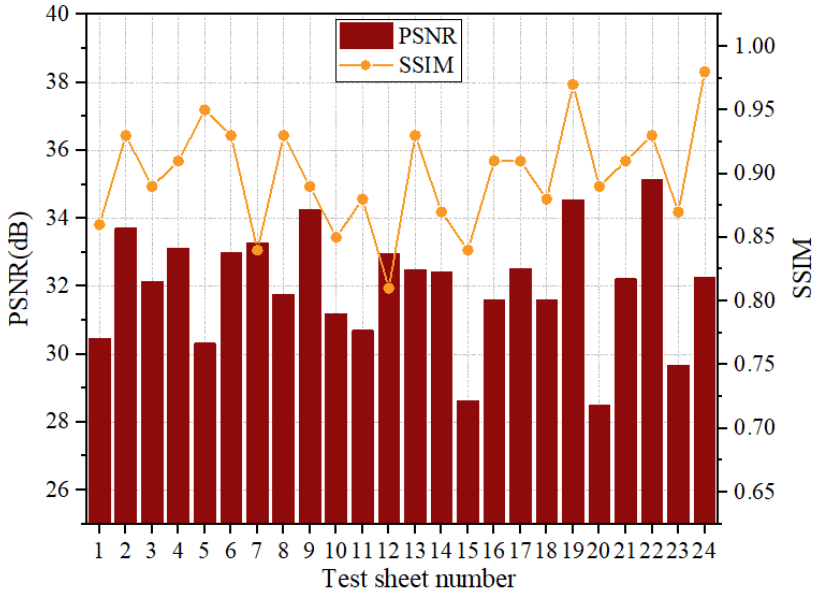

Figure 3: The PSNR and SSIM of the proposed algorithm on MNIST

The model in this paper is analyzed for the average PSNR and SSIM of the images obtained after denoising 20,000 images on the MNIST test set, respectively. The average PSNR and SSIM of this paper's model under different noises are shown in Table 2. It can be seen that after using the algorithmic model of this paper for denoising and repairing the images, the PSNR value of the model is above 28dB and the SSIM value is greater than 0.85 in five noise levels of 10, 20, 30, 40 and 50, which shows that the improved model of this paper can also obtain good results in the objective evaluation index.

Table 2: Average PSNR and SSIM of the proposed model under different noises

Test set	Noise variance	Gaussian noise value				
		10	20	30	40	50
MNIST	PSNR (dB)	38.71	31.58	30.99	29.37	28.56
	SSIM	0.97	0.95	0.93	0.89	0.86

In order to demonstrate the excellent performance of the improved model results in this paper, this paper also combines two models, SRGAN and traditional CGAN, and experimentally compares them in the MNIST test set. The average PSNR and SSIM values of different models are shown in Table 3. Among the three noise levels, the PSNR and SSIM values of this paper's model are larger than those of SRGAN and traditional CGAN models, which indicates that this paper's model can remove the presence of noise more cleanly, retain more detail information and texture features in the image, and the model has better denoising effect on the image.

Table 3: Average PSNR and SSIM values of different models

Gaussian noise value	Evaluating indicator	SRGAN	Tradition CGAN	Improve CGAN
10	PSNR (dB)	31.76	31.38	40.21
	SSIM	0.8741	0.922	0.9152
20	PSNR (dB)	29.17	28.18	34.35
	SSIM	0.8384	0.8114	0.857
50	PSNR (dB)	26.94	26.66	28.45
	SSIM	0.8057	0.7237	0.7948

3.1.3 Before and after loss function improvement

In this paper, 2000 randomly selected preprocessed images from the MNIST dataset form the training set, and the denoising results of images with noise levels of 10, 20 and 50 are compared. The average PSNR and SSIM of the model before and after the loss function improvement are shown in Table 4. The results show that the average PSNR of the model in this paper is 32.45 dB, which is 2.04 dB higher than that of the traditional CGAN model, and the SSIM value proves that the denoised result images of the model in this paper have higher structural similarity with the original ones, which indicates that the image denoising results of the improved CGAN model have good image quality and structural properties.

Table 4: Average PSNR and SSIM of the Improved and Conventional Models

Gaussian noise value	Evaluating indicator	Tradition CGAN	Improve CGAN
10	PSNR (dB)	32.16	34.65
	SSIM	0.8317	0.8978
20	PSNR (dB)	31.56	32.93
	SSIM	0.8101	0.8512
50	PSNR (dB)	27.52	29.77
	SSIM	0.8035	0.8377

3.1.4 Comparison of experimental results of image denoising algorithms

In this paper, we use three-channel joint denoising color image denoising algorithms: BM3D and DnCNN for comparative validation in experiments. The objective evaluation results in the test set at a noise variance of 50 are shown in Table 5. The results show that the denoising effect of this paper's algorithm model for each image in the MNIST test set is better than that of BM3D and DnCNN. at a noise variance of 50, the PSNR of this paper's algorithm is higher than that of the BM3D algorithm and DnCNN by an average of 4.16dB and 2.28dB; and the SSIM value is higher than that of 0.1605 and 0.124, respectively, which indicates that the denoised image repaired by this paper's algorithm has the best quality and structural similarity are the best.

Table 5: Objective evaluation results of test set with noise variance of 50

Images	BM3D		DnCNN		Improve CGAN	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	23.4	0.7268	25.74	0.7323	28.23	0.8163
2	23.91	0.6816	25.44	0.7797	28.03	0.9036
3	23.53	0.7125	24.86	0.7304	28.05	0.8946
4	22.83	0.678	25.31	0.6768	29.14	0.8746
5	24.5	0.7018	25.87	0.7186	28.3	0.9553
6	24	0.7951	26.24	0.7368	27.91	0.8048
7	24.19	0.6747	26.07	0.764	28.14	0.8033
8	23.55	0.7381	25.54	0.7508	27.68	0.9003
9	25.88	0.6699	27.89	0.8117	27.21	0.9257
10	23.46	0.7266	25.64	0.7462	30	0.8700
11	25.46	0.7225	25.68	0.7722	27.7	0.8629
12	24.05	0.7151	26	0.7194	28.02	0.9932
13	24.75	0.6676	24.91	0.7539	26.94	0.9111
14	23.83	0.7631	27.04	0.8863	27.4	0.8098
15	24.36	0.7272	26.64	0.7958	27.74	0.9059
16	23.32	0.7071	27.39	0.8768	28.38	0.8029
17	22.92	0.7444	26.24	0.7295	29.21	0.9158
18	23.91	0.7352	25.07	0.733	28.93	0.8767
19	25.06	0.6995	25.91	0.7036	27.66	0.9092
20	23.28	0.7053	25.17	0.6608	28.46	0.9236
21	24.1	0.711	25.66	0.7199	28.93	0.9455
22	23.64	0.7575	25.2	0.8023	27.5	0.7548
23	24.17	0.7153	25.5	0.7303	28.68	0.9194
24	23.53	0.7208	25.66	0.7416	27.19	0.7679
Mean	23.98	0.7165	25.86	0.7530	28.14	0.8770

3.2 Analysis of image recognition results based on convolutional neural network

This section continues with experiments on the MNIST dataset, which are divided into two parts; the first part is an experimental analysis of the improved activation function, and the second part is an experiment on the improved convolutional neural network model.

3.2.1 Experimental analysis of the improved activation function

The more commonly used activation functions in deep learning are sigmoid, ReLU function, here we set sigmoid, ReLU and improved sigmoid_ReLU function as the activation function in the improved convolutional neural network for experiments respectively. The image recognition accuracies of the three activation functions are compared on MNIST, respectively.

Figure 4 shows the recognition correctness of different activation functions on MNIST. It can be seen that the correct rate using the improved sigmoid_ReLU function as the activation function in the improved convolutional neural network is higher than that of the sigmoid and ReLU functions. As the number of iterations increases, the recognition rate of all three functions as activation function increases. On the MNIST dataset, the highest correct rate of the sigmoid activation function is 98.73%; the ReLU function works better than the sigmoid function, with the highest correct rate of 98.94%; and the improved sigmoid_ReLU activation function has

the highest correct rate of 99.09%, and converges faster. Therefore, from the experimental results, it can be seen that the improved sigmoid_ReLU function as an activation function is better than the sigmoid and ReLU functions, which can improve the recognition rate of the improved convolutional neural network on the MNIST dataset.

In summary, the sigmoid_ReLU function proposed in this paper performs better as an activation function in the improved convolutional neural network through testing on MNIST. This function is better than the commonly used sigmoid and ReLU functions, which can improve the correct rate of image recognition and the network convergence speed is faster.

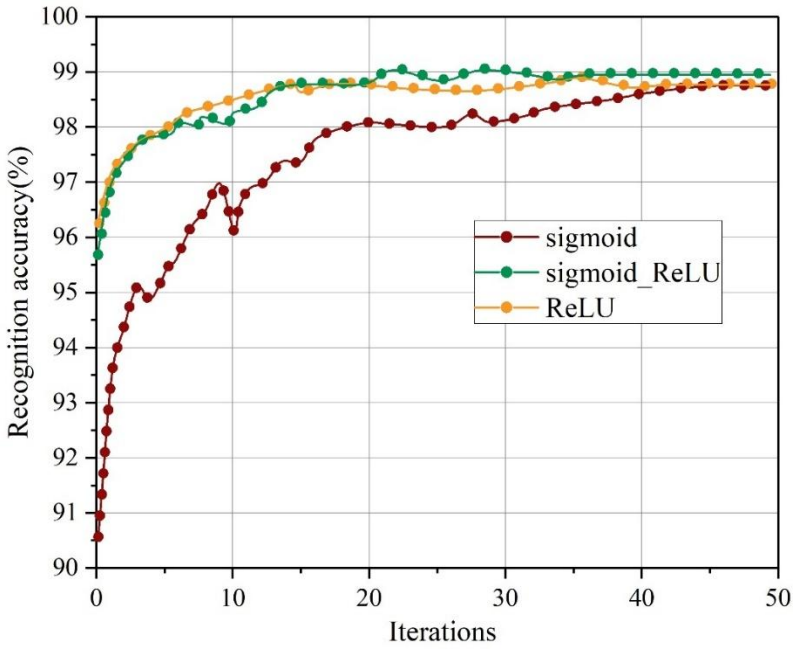

Figure 4: Recognition accuracy of different activation functions on MNIST

3.2.2 Experimental Analysis of Improved Convolutional Neural Network Models

In order to test the recognition effect of the improved convolutional neural network model, the improved convolutional neural network model is experimented on MNIST separately. Table 6 shows the experimental results of different algorithms on MNIST. It can be seen that the correct rate of the improved convolutional neural network algorithm is higher than other image recognition methods on MNIST dataset. The recognition based on fuzzy shape model has the lowest correct rate of 93.15%, followed by the improved SVM algorithm based on KNN (96.03%), the KL + BP algorithm (98.36%) and the improved convolutional neural network algorithm are closer to each other; the improved convolutional neural network algorithm achieves a correct rate of 99.47%. Therefore, the improved convolutional neural network algorithm is better than the other compared algorithms in recognizing on the MNIST dataset.

Table 6: Performance of Different Algorithms on MNIST Dataset

Algorithm	Accuracy (%)
Recognition Based on Fuzzy Shape Model	93.15
Recognition of Improved SVM Algorithm Based on KNN	96.03
Recognition of KL+BP algorithm	98.36
CGAN	96.94
Improved convolutional neural network	99.47

Fig. 5 shows the recognition correctness of convolutional neural network and improved convolutional neural network on MNIST. It can be seen that a total of 50 iterations of training are performed, and the improved convolutional neural network algorithm has a much higher correct rate than the traditional convolutional neural network algorithm at the very beginning of training, with the former achieving a correct rate of 95.80% at the very beginning of training, and the latter having a correct rate of only 90.96%. The recognition correctness of both on the MNIST dataset increases with the number of iterations, and the improved convolutional neural network model has a higher recognition rate than the traditional convolutional neural network model, with the improved convolutional neural network reaching 99.52% correct and the unimproved convolutional neural network algorithm 98.94%. After 17 iterations, the network reaches convergence, and the traditional convolutional neural network algorithm converges slowly and unstably. Therefore, the improved convolutional neural network algorithm performs better than the unimproved one in terms of both correctness and convergence speed.

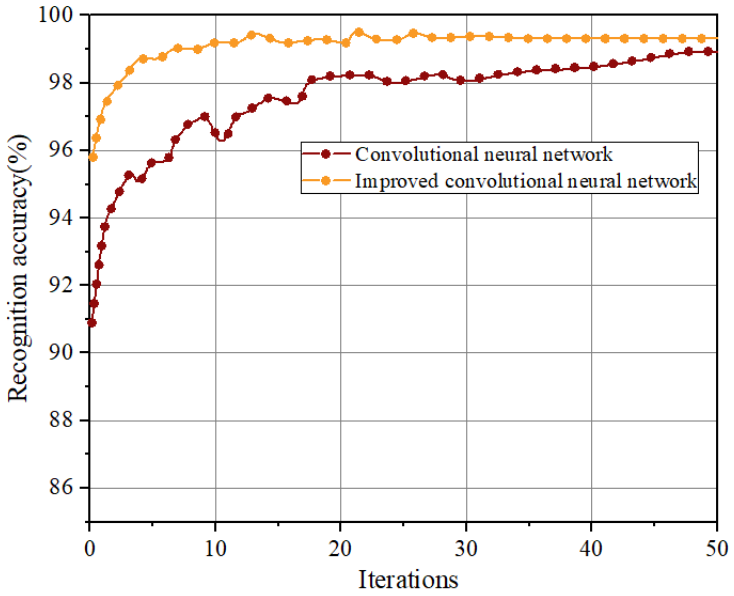

Figure 5: Recognition accuracy before and after the algorithm improvement

The average correctness of the MNIST dataset on each category of images is shown in Fig. 6, where 1 to 10 represents airplane, car, bird, cat, deer, dog, frog, horse, boat and truck respectively. It can be seen that on the MNIST dataset, the improved convolutional neural network and the unimproved convolutional neural network have higher recognition rates on top of the images of airplanes, cars, birds, frogs, and trucks, and lower recognition rates on the images of cats, dogs, horses, and boats. As a whole, the improved convolutional neural network has a higher correct recognition rate than the unimproved convolutional neural network on each category, and the average correct rate is higher than the unimproved convolutional neural network.

In summary, the feasibility of the improved improved convolutional neural network obtained by replacing the original softmax layer in the improved CGAN with an SVM classifier, mainly by using the improved Sigmoid_ReLU function, the Dropout idea, and the improved initialization weight method in the improved convolutional neural network, is experimentally verified. By comparing the experiments on MNIST dataset, it can be found that the improved convolutional neural network performs well and outperforms the unimproved convolutional neural network in terms of recognition rate and convergence speed.

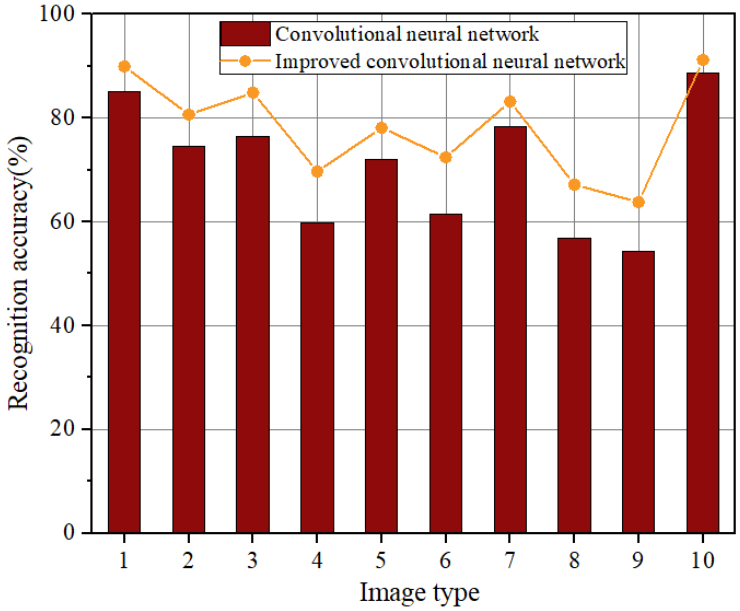

Figure 6: The average accuracy rate of MNIST dataset across all image categories

4 Conclusion

In this paper, taking MNIST dataset as an example, on the basis of neural network model, the image denoising algorithm based on improved conditional generative adversarial network is first proposed to complete the denoising of the image, and then the image recognition algorithm through convolutional neural network is used to realize the clear recognition of the image. The results are as follows:

(1) Compared with SRGAN and traditional CGAN algorithms, the model in this paper is able to remove the presence of noise more thoroughly, retain more detailed information and texture features in the image, and has better denoising effectiveness and excellent performance. The improved model using the least squares loss function can help the discriminator to better supervise the generator to generate denoised images with smaller gaps and better quality than the original image, reduce the artifacts of the image, and thus improve the recognition effect of the denoising network.

(2) The improved convolutional neural network model in this paper achieves a recognition rate of 99.47% on the MNIST dataset, which is an improvement of 2.61% relative to the pre-improvement period, and its recognition accuracy for different categories of images is higher than that of the traditional convolutional neural network model, so it can be seen that the improved model has a better performance for the application of image denoising and recognition.

About the Author

Ke Dong was born in Dali County, Shaanxi Province, China in 1980. He received the master's degree from Xian University of Science and Technology. Now, he works in is the School of Computer, XiHang University. His research interests include computer networks, network security, network management.

References

[1] Sager, C., Janiesch, C., & Zschech, P. (2021). A survey of image labelling for computer vision applications. Journal of Business Analytics, 4(2), 91-110.

[2] El Helou, M., & Süsstrunk, S. (2020). Blind universal Bayesian image denoising with Gaussian noise level learning. IEEE Transactions on Image Processing, 29, 4885-4897.

[3] Azzeh, J., Zahran, B., & Alqadi, Z. (2018). Salt and pepper noise: Effects and removal. JOIV: International Journal on Informatics Visualization, 2(4), 252-256.

[4] Xiang, J., Xiang, H., & Wang, L. (2023). Poisson noise image restoration method based on variational regularization. Signal, Image and Video Processing, 17(4), 1555-1562.

[5] Kumar, V., Dubey, A. K., Gupta, M., Singh, V., Butola, A., & Mehta, D. S. (2021). Speckle noise reduction strategies in laser-based projection imaging, fluorescence microscopy, and digital holography with uniform illumination, improved image sharpness, and resolution. Optics & laser technology, 141, 107079.

[6] Goyal, B., Agrawal, S., & Sohi, B. S. (2018). Noise issues prevailing in various types of medical images. Biomedical & Pharmacology Journal, 11(3), 1227.

[7] Gilmer, J., Ford, N., Carlini, N., & Cubuk, E. (2019, May). Adversarial examples are a natural consequence of test error in noise. In International Conference on Machine Learning (pp. 2280-2289). PMLR.

[8] Tsukui, T., Iguchi, S., Mitsuhashi, I., & Tadaki, K. (2023). Estimating the statistical uncertainty due to spatially correlated noise in interferometric images. Journal of Astronomical Telescopes, Instruments, and Systems, 9(1), 018001-018001.

[9] Steffens, C. R., Messias, L. R. V., Drews, P. L. J., & da Costa Botelho, S. S. (2019, October). Can exposure, noise and compression affect image recognition? an assessment of the impacts on state-of-the-art convnets. In 2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE) (pp. 61-66). IEEE.

[10] Rodríguez-Sánchez, Á., Thompson, A., Körner, L., Brierley, N., & Leach, R. (2020). Review of the influence of noise in X-ray computed tomography measurement uncertainty. Precision Engineering, 66, 382-391.

[11] Ostrowski, M., Blachowski, B., Mikułowski, G., & Jankowski, Ł. (2022). Influence of noise in computer-vision-based measurements on parameter identification in structural dynamics. Sensors, 23(1), 291.

[12] Ramesh, G., Logeshwaran, J., Gowri, J., & Mathew, A. (2022). THE MANAGEMENT AND REDUCTION OF DIGITAL NOISE IN VIDEO IMAGE PROCESSING BY USING TRANSMISSION BASED NOISE ELIMINATION SCHEME. ICTACT Journal on Image & Video Processing, 13(1).

[13] Kumar, N., & Nachamai, M. (2017). Noise removal and filtering techniques used in medical images. Orient. J. Comput. Sci. Technol, 10(1), 103-113.

[14] Geng, L., Ji, Z., Yuan, Y., & Yin, Y. (2017). Fractional-order sparse representation for image denoising. IEEE/CAA Journal of Automatica Sinica, 5(2), 555-563.

[15] Choi, H., & Jeong, J. (2019). Speckle noise reduction technique for SAR images using statistical characteristics of speckle noise and discrete wavelet transform. Remote Sensing, 11(10), 1184.

[16] Singh, P., Diwakar, M., Gupta, R., Kumar, S., Chakraborty, A., Bajal, E., ... & Paul, R. (2022). A method noise-based convolutional neural network technique for CT image denoising. Electronics, 11(21), 3535.

[17] Xie, Y., Cheng, J., Zhou, S., Fan, Q., Jia, Y., Xiao, J., & Liu, W. (2024). Denoising Phase-Unwrapped Images in Laser Imaging via Statistical Analysis and DnCNN. Micromachines, 15(11), 1372.

[18] Wang, H., Yang, X., Wang, Z., Yang, H., Wang, J., & Zhou, X. (2024). Improved CycleGAN for Mixed Noise Removal in Infrared Images. Applied Sciences, 14(14), 6122.

[19] Simon, J., & Kapileswar, N. (2025, August). Dual-Branch GAN-Driven Super-Resolution for Low-Dose CT Image Enhancement under Radiological Noise Constraints. In 2025 International Conference on Modern Sustainable Systems (CMSS) (pp. 710-715). IEEE.

[20] Zhang, C., Zhou, L., Zhao, Y., Zhu, S., Liu, F., & He, Y. (2020). Noise reduction in the spectral domain of hyperspectral images using denoising autoencoder methods. Chemometrics and Intelligent Laboratory Systems, 203, 104063.

[21] Yao, C., Jin, S., Liu, M., & Ban, X. (2022). Dense residual transformer for image denoising. Electronics, 11(3), 418.

[22] Liang, W., Long, J., Li, K. C., Xu, J., Ma, N., & Lei, X. (2021). A fast defogging image recognition algorithm based on bilateral hybrid filtering. ACM transactions on multimedia computing, communications, and applications (TOMM), 17(2), 1-16.

[23] Berrimi, F., Benmahammed, K., & Hedli, R. (2021). Denoising of degraded face images sequence in PCA domain for recognition. Journal of King Saud University-Computer and Information Sciences, 33(7), 836-843.

[24] Gonwirat, S., & Surinta, O. (2022). Deblurgan-cnn: effective image denoising and recognition for noisy handwritten characters. IEEE Access, 10, 90133-90148.