

Received December 24, 2021, accepted January 18, 2022, date of publication January 27, 2022, date of current version February 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3147242

# Seismic Random Noise Removal Based on a Multiscale Convolution and Densely Connected Network for Noise Level Evaluation

LIANG GUO<sup>1</sup>, RENZE LUO<sup>1</sup>, XINGYU LI<sup>1</sup>, YANG ZHOU<sup>1</sup>, TUO JUANJUAN<sup>1</sup>, AND CANRU LEI<sup>1,2</sup>

<sup>1</sup>State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation, School of Earth Sciences and Technology, Southwest Petroleum University, Chengdu 610500, China

<sup>2</sup>School of Earth Sciences and Technology, Southwest Petroleum University, Chengdu 610500, China

Corresponding author: Liang Guo (hihiwoo@163.com)

This work was supported in part by the Deep Earthquake Special Project of National Key Research and Development Program under Grant 2016YFC0601100, and in part by the Science and Technology Project of Sichuan Province under Grant 2019CXRC0027.

**ABSTRACT** Traditional denoising methods for seismic exploration data design a corresponding mathematical denoising model batch according to the different properties of different random noises, which is a tedious and time-consuming process. To solve this problem, this paper proposes a deep convolutional neural network denoising model based on noise estimation (MCD-DCNN). This model is primarily composed of two modules, the noise estimation module and the denoising module. The noise estimation module uses a multiscale convolutional neural network to better extract the characteristics of random noise in the seismic data. To make full use of the extracted features, a dense connection method is adopted between the multiscale convolutions in the noise estimation module. In the denoising module, we use multiscale convolutions and dense connections to replace the original convolutional neural network and use the residual structure (ResNet) and batch normalization (BN) to improve the denoising effect and running speed of the model. In this experiment, single trace and simple and complex profile data are used as input to simulate the real data processing environment. Finally, we compare the denoising effects of the MCD-DCNN model proposed in this paper with the current mainstream feed-forward denoising convolutional neural network (DnCNN) and a fast and flexible denoising convolutional neural network (FFDNet) models. The comprehensive results show that under the condition of a given prior noise level, the denoising performance of the FFDNet and MCD-DCNN models are comparable. In the absence of a priori noise level, the denoising performance of the FFDNet model drops sharply, while the denoising performance of MCD-DCNN is not affected; therefore, MCD-DCNN is more in line with actual seismic denoising.

**INDEX TERMS** Deep learning, seismic data denoising, noise estimate, MCD-DCNN.

## I. INTRODUCTION

In the data acquisition stage of seismic exploration, large amounts of noise are present in addition to the effective signals due to external environmental interference. For example, random noise is distributed in each frequency band of the seismic signal. Because random noise is wider than the frequency band of the effective signal, filtering can be performed in the frequency domain to remove noise outside of the effective signal frequency bandwidth. However, when the random signal overlaps the frequency band of the effective signal, the signal-to-noise ratio can be improved only by

increasing the signal energy. Removing random noise in this manner is often cumbersome and complex, and the results are inadequate [1], [2]. With the improvement of mathematical theory and the rapid development of computer hardware, machine learning can be applied to perform noise removal in seismic data.

In the traditional method of noise removal, a wavelet transform uses the difference between the effective seismic signal and the interference noise in the wavelet domain to suppress noise by setting different thresholds, which enhances the effective signal [3]. In the wavelet threshold denoising method, wavelet decomposition is performed on the acquired seismic signals, and a set of wavelet coefficients are estimated. The removal of noise is achieved through

The associate editor coordinating the review of this manuscript and approving it for publication was Naveed Ur Rehman <sup>1</sup>.

threshold modification, wavelet reconstruction and other operations. Using the frequency contents in the seismic data, Chen *et al* adaptively decomposed nonstationary seismic data into empirical components through empirical wavelet transform (EWT) and selected the first component to represent the useful signal. This idea was adopted to denoise seismic data and achieved good results [4]. Mortezaejad and Gholami converted the seismic signal into the wavelet domain and adjusted the threshold to denoising seismic data [5]. However, the artificial control threshold is not efficient because of the complex seismic signal features and large quantity of data.

Deep learning based on probability and statistics has been widely used in computer vision processing, speech processing and other fields. Deep learning methods, which can mine the hidden features of complex data, map low-dimensional data into high-dimensional data [6]–[8]. When deep learning is applied to seismic signal processing, a single trace dataset can deepen the learning network to realize global (profile) feature extraction and suppress random noise and other interference signals in seismic data [9]. The deep learning methods that are used to remove noise from seismic signals have the following problems. First, processing data in seismic exploration requires extremely deep networks [10]. However, as the network deepens, gradient disappearance and explosion problems are prone to occur. This problem can be solved by adding a residual module to the network [11]–[13]. Second, when a deep learning network is used to denoise seismic data, it is necessary to conduct an artificial evaluation of the noise in the data. However, it is very difficult to use artificial methods to estimate the noise of seismic signals [14], [15] because the noise frequency bandwidth and other components of seismic data are complex [16], [17]. An unsupervised learning neural network does not need to estimate noise, and the features of noise are directly obtained from the training data to quickly build a network [18], [19]. In a deep convolutional autoencoder network, the performance of the network is optimized through sample training, to achieve the purpose of noise removal [20], [21]. The autoencoder network consists of an encoder and a decoder. The encoder consists of a convolutional layer and a pooling layer, which can map high-dimensional seismic data to low-dimensional seismic data. The encoder is sparsely represented to obtain the features of the effective signal and noise in seismic data. The decoder consists of a convolutional layer and an upsampling layer and can amplify all the extracted features and perform signal-to-noise separation. After suppressing noise, the signal is recombined [22]. This method can handle general random noise; however, when dealing with complex noise, the signal is often greatly reduced. Moreover, poor noise removal effects occur when processing seismic data with multifrequency random noise.

In response to the above problems, experts have improved the denoising effect by deepening the convolutional neural network. For example, the feed-forward denoising convolutional neural network (DnCNN) based on discriminative

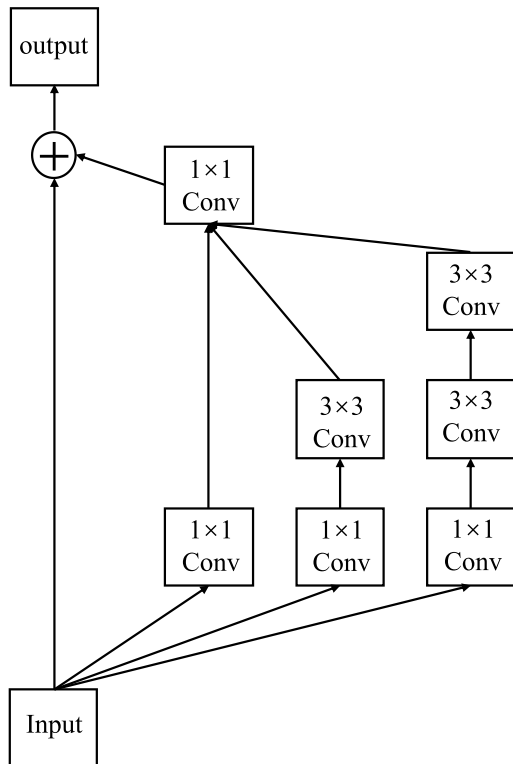
learning uses residual units to predict noise, output residual data and remove noise [23], [24]. Instead of training the input data, the DnCNN trains the residual between the input data and the predicted data to reduce the number of calculations. Using this training method will increase the depth of the network while maintaining the performance of the network. However, the DnCNN has limited flexibility, and the learned model can only target a specific noise level. The effect of removing Gaussian noise is excellent in the DnCNN-B model, but it is difficult to apply to seismic data containing various complex noises [25], [26]. Therefore, experts expanded the noise level as part of the network input. For example, Zhang *et al.* proposed a fast and flexible neural network (FFDNet) to remove random noise from seismic signals [27]. Unlike the DnCNN, the FFDNet expands the noise level to the dimension of the input data, and then the noise and input data are input into the CNN together. Additionally, the input data are downsampled and output after being upsampled. In this way, while ensuring the effect of noise removal, the process of noise removal is more efficient. However, when the input noise level does not match the data noise level, the performance of the model will be greatly reduced.

In this regard, this paper proposes a seismic data denoising method based on a noise estimation and deep convolutional neural network (MCD-DCNN). Different from the conventional deep convolutional network denoising (DCNN) model, the MCD-DCNN is primarily composed of two modules: the noise estimation module and the denoising module. The noise estimation module adopts a multiscale convolutional model to better extract the characteristics of complex noise in seismic data. We use dense connections between each multiscale convolutional layer to avoid the problem of vanishing gradients. In addition, dense connections can make better use of the extracted features. We use the automatically estimated noise level as part of the input to maximize the performance of the denoising model. In the denoising module, we also use a multiscale convolutional model with dense connections instead of the conventional convolutional module in the convolutional neural network (CNN).

## II. THEORY AND PROCESS

### A. MULTI-SCALE DENSELY CONNECTED NOISE ESTIMATION MODULE

In a CNN, different convolution kernels have different receptive fields, and the features of the extracted data are also different. The large-scale convolution kernel is suitable for extracting global information, and the small-scale convolution kernel is suitable for extracting local information. In the noise estimation of seismic data, convolution kernels of different scales are used to extract the features of the data, which results in a more comprehensive seismic data feature extraction. The multiscale convolutional structure is shown in Fig. 1.



**FIGURE 1.** The framework of the multiscale convolutional model. The multiscale convolutional structure contains three branches, which are constructed using  $1 \times 1$ ,  $3 \times 3$ , and  $5 \times 5$  convolutions. Among them, the  $5 \times 5$  convolution is replaced by two  $3 \times 3$  convolutions, which reduces the number of calculations.

The original data collected by seismic exploration consists of effective signals and noise and can be expressed as:

$$y(t) = x(t) + v(t) \quad (1)$$

where  $y(t)$  represents the original data collected by seismic exploration,  $x(t)$  is the effective signal component in the data, and  $v(t)$  is the noise. The calculation of the three channels is as follows:

$$y_1(t) = \text{Conv}_{1 \times 1}(y_{in}(t)) \quad (2)$$

$$y_2(t) = \text{Conv}_{3 \times 3}(\text{Conv}_{1 \times 1}(y_{in}(t))) \quad (3)$$

$$y_3(t) = \text{Conv}_{3 \times 3}(\text{Conv}_{3 \times 3}(\text{Conv}_{1 \times 1}(y_{in}(t)))) \quad (4)$$

where  $y_{in}(t)$  is the input seismic noise data,  $y_1(t)$ ,  $y_2(t)$ ,  $y_3(t)$  are the feature data extracted by the three channels and  $\text{Conv}$  is the convolution operation. After each convolution operation, ReLU and batch normalization are used to increase nonlinearity, prevent overfitting, and reduce the amount of calculation. Then, the results of the 3 branch convolutions are feature spliced, and a  $1 \times 1$  convolution is used to compress the number of channels and reduce the amount of network operations. Finally, the residual structure is used to fuse the input features after the  $1 \times 1$  convolution with the spliced and compressed features to restore some of the original

features [28]. The specific calculation process is as follows:

$$\hat{y}(t) = \text{Conv}_{1 \times 1}(\text{concat}(y_1(t), y_2(t), y_3(t))) \quad (5)$$

$$y_{out} = y_{in}(t) \oplus \hat{y}(t) \quad (6)$$

where  $\text{concat}$  represents the fusion of features, and  $\hat{y}(t)$  is the data result of the  $1 \times 1$  convolution operation after feature fusion. Eqn. 6 is equivalent to the residual calculation, where  $\oplus$  represents the addition operation of the data, and  $y_{out}$  is the output of the entire structure of the multiscale convolution. To increase the utilization of data features, a dense connection method is adopted between multiscale convolutions. The structure of the dense connection network is shown in Fig. 2. This structure can combine the information features of one convolutional layer with the information extracted by the previous convolutional layer through feature splicing to strengthen the ability of feature extraction. In addition, after each convolution, the batch normalization layer and ReLU are used to increase the operating efficiency and nonlinear expression of the model. This connection method results in a very large amount of calculations while reusing feature information. Therefore, after each multiscale convolution, a  $1 \times 1$  convolution is used to compress the number of channels and reduce the amount of network operations [29].

In a dense connection, the input of the  $i$ -th layer is  $\hat{y}_{i-1}$ , and the output is  $\hat{y}_i$ .

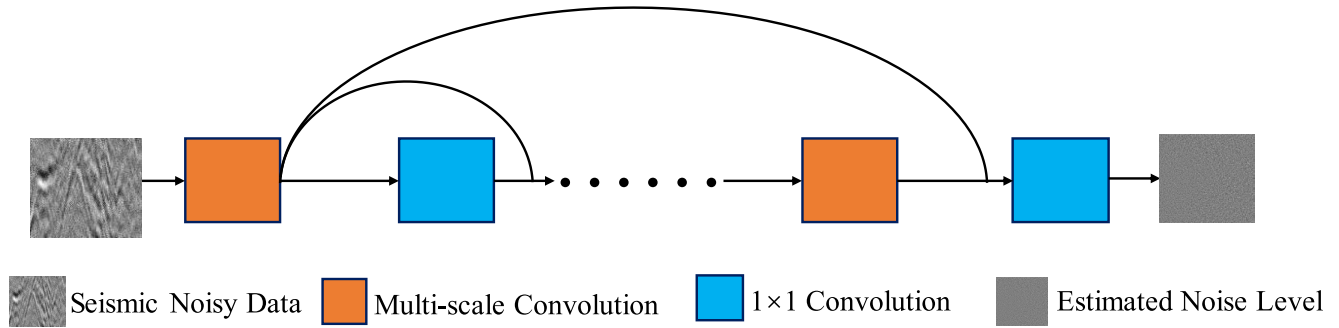
$$\hat{y}_i = H_i([\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{i-1}]) \quad (7)$$

where  $[\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{i-1}]$  are all the features in front of the concatenation of  $i$  layer, and  $H_i$  represents the nonlinear mapping combination of batch normalization and ReLU operations.

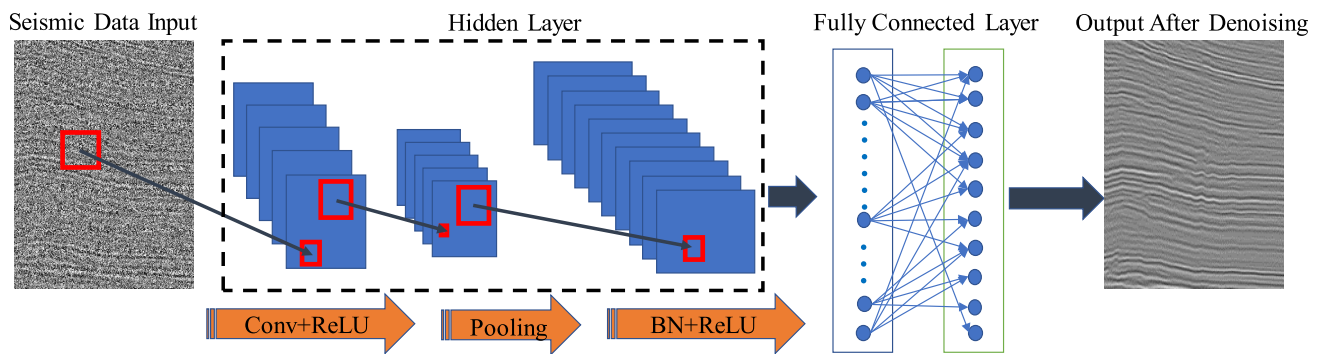
In general, there are two structures to estimate seismic noise data: multiscale convolution and dense connection. The dense connection can be regarded as the main framework of the noise estimation module, and the multiscale convolution is the basic operational unit of the module. We use the multiscale convolution to fully extract the features of seismic data and use the dense connections to ensure that these features are fully utilized. This structure combination will increase the number of calculations. Therefore, a batch normalization and  $1 \times 1$  convolution operation are used after each operation. Batch normalization can maintain the same distribution of seismic data after each operation, which can effectively prevent overfitting. The  $1 \times 1$  convolution can compress the channel and speed up the training of the model. In the noise evaluation module, we use the mean square error (MSE) as the loss function. Assuming that the input is noisy seismic data related to the noise level  $\sigma$ , our optimization goal is represented by Eqn. 8. Finally, we use an adaptive moment estimation (Adam) algorithm [30] to optimize the loss function  $L(\sigma)$ .

$$L(\sigma) = \frac{1}{2N} \sum_{i=1}^N \|y_i - \hat{y}_i\|^2 \quad (8)$$

It should be noted that the estimated noise level  $\sigma$  is related to the predicted value  $\hat{y}_i$ . The predicted value  $\hat{y}_i$  is determined



**FIGURE 2.** The structure of the dense connection network. A densely connected block contains a multiscale convolution structure, a batch normalization method and an activation function. Batch normalization helps to pull the input distribution from the saturated area to the unsaturated area. This reduces the gradient dispersion, improves the training speed, and greatly accelerates the convergence process. The ReLU is more conducive to gradient descent and back propagation and avoids gradient explosion and disappearance problems.



**FIGURE 3.** The CNN framework. The structure of the CNN can be divided into five layers: input, convolutional, activation, pooling and fully connected. In the input layer, we can improve the running speed of the network through the processes of upsampling and downsampling. The convolutional layer is the core module of the CNN because it realizes the extraction of data features through combining local perception areas, sharing weight, and using multiple convolution kernels, and then transfers the extracted features to higher dimensions. The activation layer increases the nonlinear output of the network through the excitation function and improves the performance of the network. Batch normalization (BN) unifies the data to a similar distribution range, reducing the probability of gradient explosion or disappearance. The pooling layer reduces the number of parameters and the dimensionality of the features extracted by the network. This layer also compresses the data to reduce overfitting and improve the fault tolerance of the model. The fully connected layer integrates the extracted features and outputs them. The convolutional, BN and activation layers are considered hidden layers.

by the weight  $w_i$  and bias  $b_i$ , and the specific calculation is shown in formula 9. Therefore, the module’s evaluation of noise is essentially the result of optimizing parameters  $w_i$  and  $b_i$ .

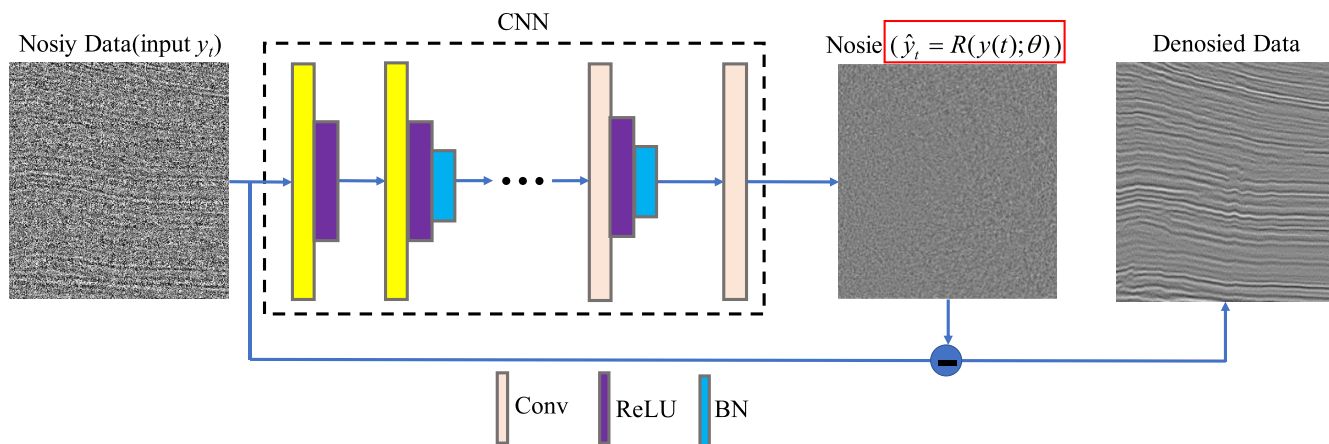
**B. DENOISING MODULE**

The main purpose of the CNN that is used in seismic data denoising is to extract the characteristics of seismic data through the convolution operation. As the number of convolutions increase, the features extracted by the network rise from a low-dimensional space to a high-dimensional space, and features become increasingly more abstract. The overall performance of the network will also improve. Feature extraction is a process of mathematical mapping. This mapping is related to the weight and bias terms. The mapping relationship is given by Eqn. 9.

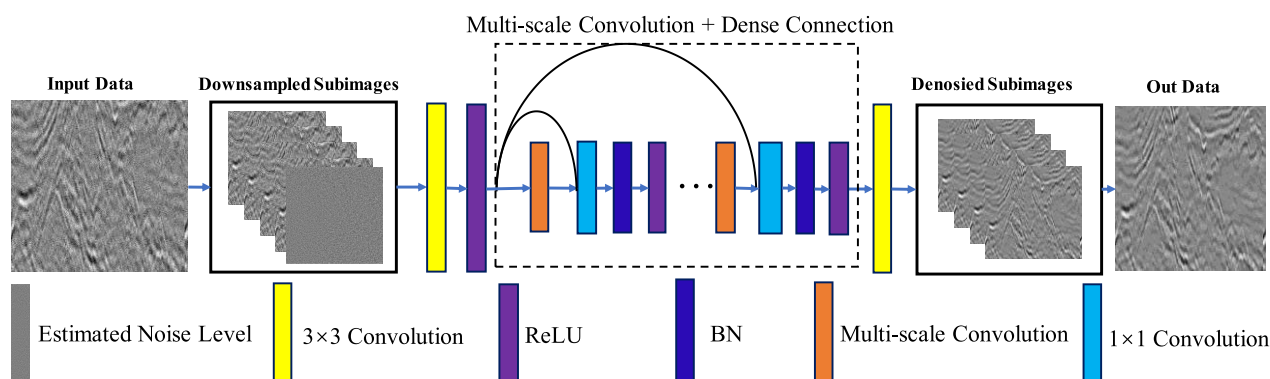
$$\hat{y}_i = H\left(\sum_{i=1}^n w_i * x_i + b_i\right) \tag{9}$$

where  $w_i$  is the weight term,  $b$  is the bias term. and  $*$  is a convolution operation. To increase the nonlinear ability of the network, we add the ReLU after each convolutional layer. The variable  $H$  in Eqn. 9 represents the nonlinear mapping. The output of the hidden layer is used as the input of the fully connected layer at the final part of the network. The extracted features pass through the fully connected layer to output denoised seismic data. The specific process is shown in Fig. 3.

The features extracted by the CNN are used to reconstruct denoised effective seismic data [31]. There are two disadvantages of this process. First, the feature complexity of the effective signal data are much greater than that of the noise, which requires very high network performance. Second, directly reconstructing effective signal data often results in the loss of a large amount of characteristic information [32]. Therefore, using the idea of noise separation, the cost of directly predicting noise is far less than the cost of directly predicting effective signal data [33]. The specific structure is shown in Fig. 4.



**FIGURE 4.** The D-CNN framework. The structure of the D-CNN is similar to the CNN. The only difference in the two networks is that the clean and effective signal data of the prediction object in the D-CNN becomes the direct prediction noise in the CNN. Then, the denosied data are output by the residual method.



**FIGURE 5.** The denoising module framework. In the architecture of the entire model, we refer to the model of reference [27]. The denoising model divides the seismic data into 4 sub-data through down-sampling. The sub-data and the noise level estimated by the noise evaluation model are used as the input of the denoising model. In order to fully extract the features of seismic data and make full use of these features, we replaced the convolutional network in the original model with multi-scale convolution and dense connection.

Problems with the CNN, such as gradient explosion or disappearance and network performance degradation, occur as the network deepens. Zhang, et al proposed the DnCNN, which uses residual learning and batch normalization to deepen the network, increase the training speed, and improve the performance of the network. Since the network directly trains the noise, formula 1 can be rewritten as:

$$v(t) = y(t) - x(t) \tag{10}$$

We use the residual data  $v(t)$  and the mean square error of the pretrained residual data as the loss function. The output of the model is shown in the red rectangular circle in Fig 4.

$$L(\theta) = \frac{1}{2N} \sum_{i=1}^N \|R(y(t); \theta) - v(t)\|^2 \tag{11}$$

where  $\theta$  is the parameter of the denoising module, whose optimization satisfies the goal of the entire network,  $R(y(t); \theta)$  is the pretrained residual data, and  $v(t)$  is the residual data related to the noise in the original data.

We improve the CNN structure in Fig. 4 by replacing the CNN in the network with multiscale convolutions and

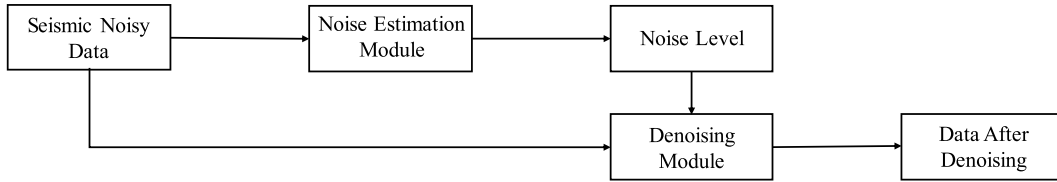
dense connections. The multiscale convolutions can extract richer seismic data features, and the dense connections can make full use of the extracted features. The denoising module framework is shown in Fig. 5.

Regardless of how the CNN network is improved, it is necessary to estimate the noise level in the original seismic data. The overall process of seismic data denoising is shown in Fig. 6. The noisy seismic data are input into the noise estimation module. The noise estimation module outputs the noise estimation. We splice the noise estimate with the original seismic data and input the result into the denoising module to obtain clean seismic data. The densely connected structure is adopted in the noise estimation module so that the noise estimation can reflect the details of the original seismic data to be obtained.

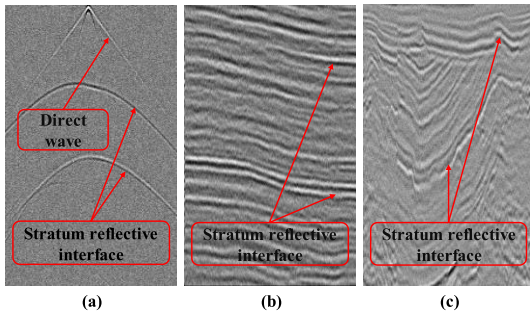
### III. SEISMIC NOISY DATA CLASSIFICATION AND MODEL EVALUATION INDEX

#### A. CLASSIFICATION OF SEISMIC NOISY DATA

Noise present in seismic exploration data can be roughly divided into two categories. The first category is regular



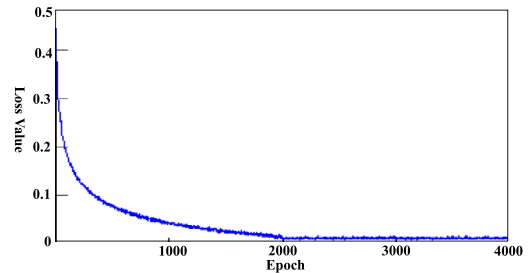
**FIGURE 6.** The seismic data denoising process. The specific steps are: 1) Prepare the seismic input data. 2) Input the noisy seismic data into the denoising module. 3) Start the optimization with Eqn. 8 as the loss function. 4) Back-propagate the error using the noise estimation module. 5) Repeat steps 2-4 until the noise estimation module reaches the optimum level. 6) Input the estimated noise level and the corresponding noisy seismic data into the denoising module. 7) Start the optimization with Eqn. 11 as the loss function. 8) Back-propagate error using the denoising module. 9) Repeat steps 6-8 until the denoising module reaches the optimal level. 10) Output the denoised seismic data.



**FIGURE 7.** The categories of noisy seismic data. (a) The simple synthetic seismic profile data (before migration stack). (b) The shallow simple stratum seismic profile data (after migration stack). (c) The deep complex stratum seismic profile data (after migration stack). In (a), the upper red circle is the position of the direct wave (after the seismic source is excited, and the seismic wave has directly reached the wave form of the geophone along the ground surface). The rest of the red circle marks the stratum reflection interface.

noise, and an example is industrial electrical interference. There is a tendency for the frequency or apparent speed of this type of noise to occur at a specific value. The conventional denoising method removes regular noise based on the difference between the regular noise and effective signal. The second category is irregular or random noise. Random noise is not only related to the field collection environment (ocean, land, mountains, plains, etc.) but is also related to the underground geological environment. The difficulty of random noise removal is different for different geological environments, but it is especially difficult in the geological environment, which is the focus of this paper.

We roughly divide the geological environment into two types: the horizontal strata in the subsurface and the deep complex units, as shown in Fig. 7. Although the random noise is complicated in the horizontal strata in the subsurface, the effective seismic signal features are obvious. This type of data distribution is very beneficial for denoising. In the deep complex units, the seismic data are very complex. This phenomenon makes it very difficult to remove random noise from seismic data. In summary, we divide noisy seismic data into simple and complex categories. The term simple corresponds to horizontal strata in the subsurface, and the term complex corresponds to deep complex units.



**FIGURE 8.** The denoising module stability test. The loss function converges when the epoch is set to 2000. When the epoch is set to 4000, the range of change is stable at approximately 0.01, which meets the requirements of this experiment. When the experiment needs to retrain the model, the epoch is set 4000 to save time while ensuring the performance of the noise evaluation module.

## B. EVALUATION INDEX

In this experiment, the peak signal-to-noise ratio (PSNR) and the root mean square error (RMSE) were used to evaluate the denoising effect. The PSNR and RMSE are calculated as follows:

$$PSNR = 10 \log_{10} \frac{\|x\|^2}{\|\hat{y} - x\|^2} \quad (12)$$

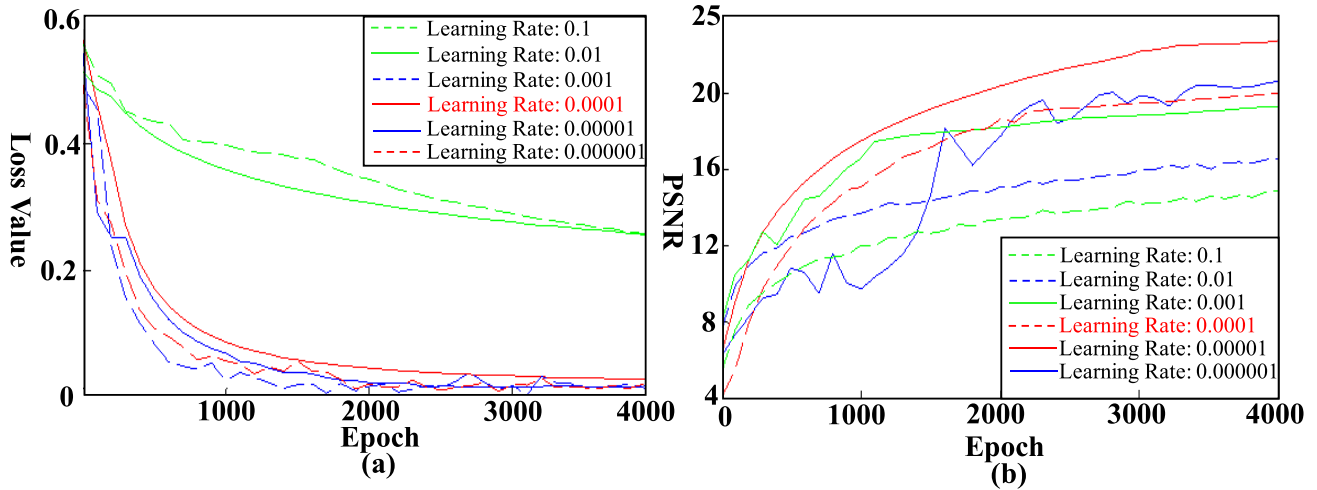
$$RMSE = \sqrt{\frac{\|\hat{y} - x\|_2^2}{T \times J}} \quad (13)$$

where  $\|x\|^2$  is the power of effective signal,  $\|\hat{y} - x\|^2$  is the power of noise,  $J$  is the number of sampling points per track, and  $T$  is the number of tracks.

## IV. MODEL TRAINING AND PARAMETER SETTING

### A. NOISE ESTIMATION MODULE PARAMETER SETTINGS

For the noise evaluation module, the main parameters are the convolution kernel size of the multiscale convolutional neural network, the number of channels, and the length of dense connections. Under the premise of fully extracting seismic data features and to take into account the efficiency of model training, we set the convolution kernel size of the multiscale convolutional network to  $3 \times 3$  and adopt the three-channel mode. For dense connections, we use 4 multiscale convolution modules. Equation 8 is used as the loss function to test the stability of the module, that is, the relationship between



**FIGURE 9.** The effect of different learning rates. (a) The loss value of the denoising model with different epoch settings (b) The PSNR of the denoising model with different epoch settings.

epoch and loss value, as shown in Fig. 8. Through multiple iterations, we set the number of epochs of the denoising module to 4000.

### B. DENOISING MODULE PARAMETER SETTINGS

Since the denoising module uses a deeper network, the parameters of the module not only affect the performance, but also affect the calculation speed. Therefore, we must consider operating time and cost when selecting parameters. In this optimization process, the PSNR was used as the evaluation index.

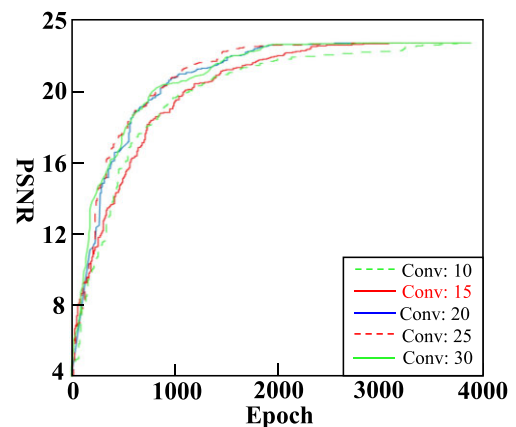
#### 1) LEARNING RATE

As an important parameter of the denoising module, the initial learning rate determines the convergence effect and speed of the algorithm, as well as the effect of noise removal. In the experiment, the model is tested with different learning rates, as shown in Fig. 9.

When the learning rate is set to 0.1 or 0.01, the performance of the model is not optimal, and the PSNR does not reach the expected goal of the experiment. When the learning rate is set to  $1 \times 10^{-5}$  or  $1 \times 10^{-6}$ , the performance of the denoising module improves but is still not optimal; however, the training time of the module is twice as long when the learning rate is set to 0.0001. When the learning rate is  $10^{-6}$ , the PSNR of the denoising module greatly fluctuates and the performance is unstable. Based on the above factors, we set the learning rate to 0.0001.

#### 2) CONVOLUTION DEPTH (CONVOLUTIONAL LAYERS)

To pursue better nonlinear expression ability, it is generally necessary to deepen the network and learn more complex transformations to handle more complex feature inputs. This is especially important for complex seismic data from similar seismic explorations. Therefore, deepening the network could



**FIGURE 10.** The denoising effect of different convolutional layers.

improve the denoising ability of the model to a certain extent. However, as the network deepens, problems such as gradient instability and network performance degradation occur. The performance of the denoising module is tested at different network depths (10, 15, 20, 25 and 30 layers). The test results are shown in Fig. 10.

When the number of network layers is set to 10, the denoising performance is significantly lower than the other networks. Considering the denoising effect from 15 to 30 layers, the performance of the model does not decrease with the deepening of network layers; however, there is no significant improvement. Based on the above factors, the depth is generally set to 15 layers.

#### 3) PATCH SIZE AND CONVOLUTION KERNEL SIZE

The patch size is the size of the data input at one time and represents a small piece of the entire seismic data. To test the impact of different data size inputs on the network, the depth

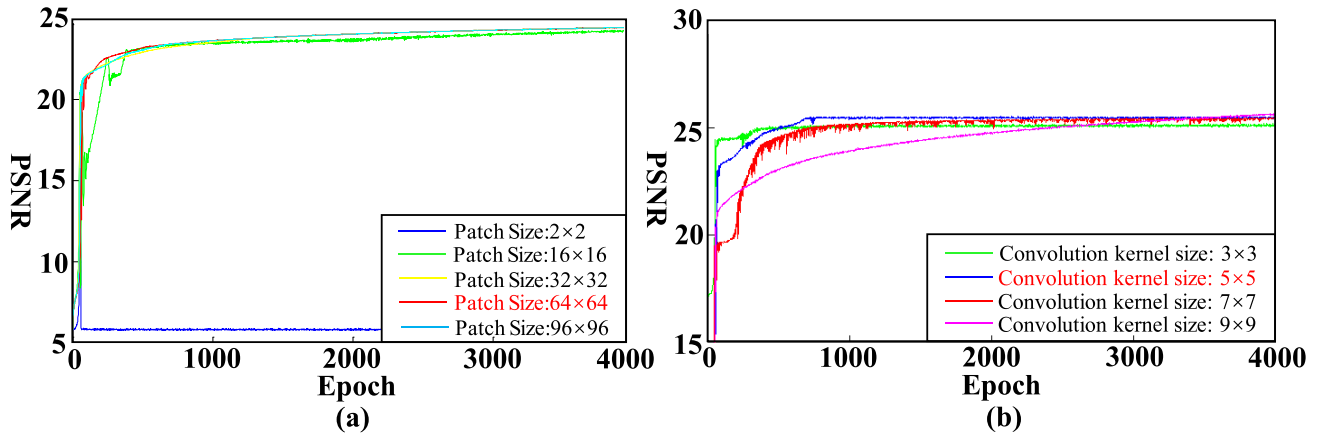


FIGURE 11. The effect of different patch sizes and convolution kernel sizes. (a) is the effect of patch size, (b) is the convolution kernel size.

TABLE 1. Model training time based on convolution kernel size.

| Convolution kernel size | 3×3  | 5×5  | 7×7  | 9×9  |
|-------------------------|------|------|------|------|
| Time (h)                | 2.15 | 2.54 | 4.76 | 5.80 |

of the network was set to 15, and the size of the convolution kernel was set to  $5 \times 5$ . Different data sizes ( $2 \times 2$ ,  $16 \times 16$ ,  $32 \times 32$ ,  $64 \times 64$ ,  $96 \times 96$ ) were used as input. The denoising module uses the same data. The performance is shown in Fig. 11(a). The  $2 \times 2$  data input does not improve the model. When the input data size changes from small to large, the difference in the denoising effect is not obvious; however, the stability of the network is different. For example, when the patch size is  $16 \times 16$ , the PSNR fluctuates greatly. The patch size of  $64 \times 64$  is selected as the size of the data input after considering the comprehensive results.

In general, the larger the size of the convolution kernel is, the larger the receiving field of view, the greater amount of information received, and the better the global features obtained. However, as the size of the convolution kernel becomes larger, the number of calculations required by the network model increases sharply. In the case of the same computer hardware, the depth of the model is affected, and the calculation performance may also be reduced. Convolutional kernels of  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$  are used to verify the performance of denoising module in seismic data processing, as shown in Fig. 11(b).

For an input data size of  $64 \times 64$ , when the convolution kernel is increased from  $3 \times 3$  to  $5 \times 5$ , the effect is significantly improved. After a convolution kernel size of  $5 \times 5$ , as the size of the convolution kernel increases, the effect does not significantly change; however, the training time increases significantly, as shown in Table 1. Considering time as a factor, we set the size of the convolution kernel to  $5 \times 5$ . After a series of model training, the parameter settings of MCD-DCNN are shown in Table 2.

TABLE 2. Main parameter settings of the MCD-DCNN.

| Module                  | Parameter name                     | Parameter settings |
|-------------------------|------------------------------------|--------------------|
| Noise evaluation module | Convolution kernel size            | $3 \times 3$       |
|                         | Number of channels                 | 3                  |
|                         | Number of densely connected layers | 4                  |
| Denoising module        | Learning rate                      | 0.0001             |
|                         | Convolutional layers               | 15                 |
|                         | Patch size                         | $64 \times 64$     |
|                         | Convolution kernel size            | $5 \times 5$       |

## V. MODEL TEST AND RESULT ANALYSIS

To test the application of this model, we divided the experiment into two stages. The first stage was the noise evaluation experiment. The second stage was the MCD-DCNN denoising performance experiment. In the second stage of the experiment, we used single-trace seismic data as the test object to verify the denoising ability of the MCD-DCNN model because the feature extraction of single-trace seismic data only involves the horizontal and vertical directions. Second, we used simple seismic profile data to test the performance of the MCD-DCNN model. The simple profile data included synthetic data and actual shallow seismic profile data. These two types of seismic data are shown in Fig. 7 (a) and Fig. 7 (b) in the second part of this paper. Finally, we changed the test object into complex deep seismic data which are shown in Fig. 7(c). The seismic wave reflection interface of this kind of seismic data fluctuates greatly and is discontinuous, and the noise is difficult to remove. The experimental research environment was a CentOS7, an Intel Core i5-8400 processor, and an NVIDIA RTX2080Ti GPU. The programming language was Python 3.6.6.

### A. NOISE EVALUATION MODULE EXPERIMENT

In this portion of the experiment, we selected 100 seismic profile data. Among them, type (a) data contains 40 profiles, and types (b) and (c) data contain 30 profiles each. Each seismic



**TABLE 3. Noise estimation of three kinds of seismic data under different noise levels.**

| Type of data | Noise level | HBNE            |      | QNE             |             | DWTNE           |             | This paper      |             |
|--------------|-------------|-----------------|------|-----------------|-------------|-----------------|-------------|-----------------|-------------|
|              |             | Estimated value | RMSE | Estimated value | RMSE        | Estimated value | RMSE        | Estimated value | RMSE        |
| (a)          | 10          | 9.32            | 0.48 | <b>9.75</b>     | <b>0.18</b> | 10.72           | 0.51        | 10.97           | 0.69        |
|              | 20          | 21.32           | 0.93 | 20.71           | 0.50        | 18.98           | 0.72        | <b>20.46</b>    | <b>0.33</b> |
|              | 30          | 29.64           | 0.25 | 29.36           | 0.45        | 28.35           | 1.17        | <b>30.17</b>    | <b>0.12</b> |
|              | 40          | 38.58           | 1.00 | 37.51           | 1.76        | 38.41           | 1.12        | <b>40.98</b>    | <b>0.69</b> |
|              | 50          | 46.69           | 2.24 | 47.54           | 1.74        | 46.64           | 2.38        | <b>50.63</b>    | <b>0.45</b> |
|              | 60          | 54.21           | 4.09 | 55.17           | 3.42        | <b>58.73</b>    | <b>0.90</b> | 61.73           | 1.22        |
|              | 70          | 61.13           | 6.27 | 59.35           | 7.53        | 59.57           | 7.38        | <b>70.51</b>    | <b>0.36</b> |
| (b)          | 10          | 10.73           | 0.52 | <b>9.54</b>     | <b>0.33</b> | 9.13            | 0.62        | 10.52           | 0.37        |
|              | 20          | 20.32           | 0.23 | 19.11           | 0.63        | 18.29           | 1.21        | <b>20.13</b>    | <b>0.09</b> |
|              | 30          | 28.31           | 1.20 | 27.71           | 1.62        | 28.14           | 1.32        | <b>31.01</b>    | <b>0.71</b> |
|              | 40          | 38.56           | 1.02 | 37.85           | 1.52        | 37.47           | 1.79        | <b>40.97</b>    | <b>0.69</b> |
|              | 50          | 44.37           | 3.98 | 45.41           | 3.25        | 46.62           | 2.39        | <b>50.35</b>    | <b>0.25</b> |
|              | 60          | 55.79           | 2.98 | 55.38           | 3.27        | 53.59           | 4.53        | <b>61.57</b>    | <b>1.11</b> |
|              | 70          | 59.92           | 7.13 | 60.69           | 6.58        | 64.78           | 3.69        | <b>72.17</b>    | <b>1.53</b> |
| (c)          | 10          | 11.26           | 0.89 | <b>10.32</b>    | <b>0.23</b> | 9.37            | 0.45        | 10.86           | 0.61        |
|              | 20          | 18.33           | 1.18 | 18.76           | 0.88        | 20.88           | 0.62        | <b>20.73</b>    | <b>0.52</b> |
|              | 30          | 27.84           | 1.53 | 27.48           | 1.78        | 26.98           | 2.14        | <b>30.27</b>    | <b>0.19</b> |
|              | 40          | 36.54           | 2.45 | 35.91           | 2.89        | 36.79           | 2.27        | <b>40.93</b>    | <b>0.66</b> |
|              | 50          | 43.22           | 4.79 | 44.48           | 3.90        | 45.66           | 3.07        | <b>50.71</b>    | <b>0.50</b> |
|              | 60          | 50.2            | 6.93 | 52.57           | 5.25        | 53.22           | 4.79        | <b>61.64</b>    | <b>1.16</b> |
|              | 70          | 59.82           | 7.20 | 57.98           | 8.50        | 60.39           | 6.80        | <b>68.35</b>    | <b>1.17</b> |

profile consists of 500 single-trace data. Each seismic trace has 1201 sampling points, and the sampling interval is 2 ms. Part of the data are shown in Fig. 12.

In the experiment, we selected 10 profiles for each of the three types of data as the test set, and the rest of the data as the training set. In the three types of data, we added different levels of noise ( $\sigma$ : 10, 20, 30, 40, 50, 60, 70) and used the histogram-based noise estimation (HBNE) algorithm, quantile noise estimation (QNE) and discrete wavelet transform noise estimation (DWTNE) to compare with the model proposed in this paper. Table 3 shows the noise estimates and RMSE of each method on the three data types under different noise levels in detail.

In Table 3, we display the results with the best evaluation results in **black** font. It can be seen from Table 3 that the noise estimation model proposed in this paper has the best result. Integrating Table 3 and Fig. 12, the HBNE, QNE and DWTNE show three major characteristics. First, the HBNE, QNE and DWTNE have different noise estimation results for noisy seismic data with different noise levels. When the noise level is in the range of 10 to 50, and the estimation results of the HBNE, QNE and DWTNE are within the acceptable range (RMSE less than 3.5). When the noise levels are 60 and 70, the evaluation performance of the HBNE, QNE and DWTNE decrease rapidly because it is difficult to distinguish the data of the effective seismic signal from the data of the noise signal when the noise level increases. In addition, when the test data changes from simple to complex, the evaluation effect of the HBNE, QNE and DWTNE also decreases. However, the decline is not obvious. This is a common problem of conventional noise estimation models when dealing with complex data. Finally, when the noise level in the seismic data increases, the estimated results of

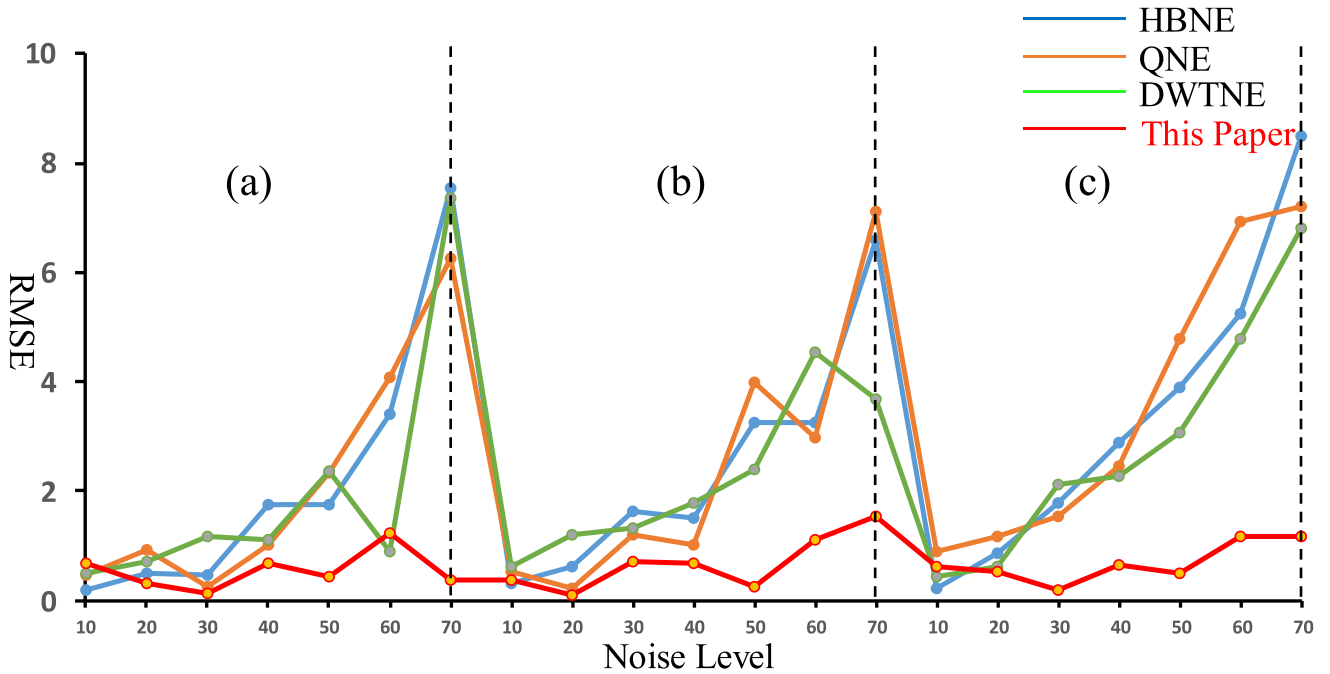
the HBNE, QNE and DWTNE are generally lower. Similar to the previous reason, when the effective signal data and the noise signal data are difficult to distinguish, the traditional method easily treats the effective signal as noise, which will lead to a low evaluation result. The model proposed in this paper adopts multiscale convolutions and dense connections, which can fully extract the characteristics of seismic signals. In particular, global features of seismic data can be extracted for dense connections. This is different from traditional methods that only use local features of seismic data. Therefore, when given complex and highly noisy seismic data, the performance of the noise evaluation model proposed in this paper is still considered. For example, when the noise level is 70, the RMSE of the evaluation results of the model in this paper on the three types of data are 0.36, 1.56, and 1.17, respectively.

## B. DENOISING EXPERIMENT

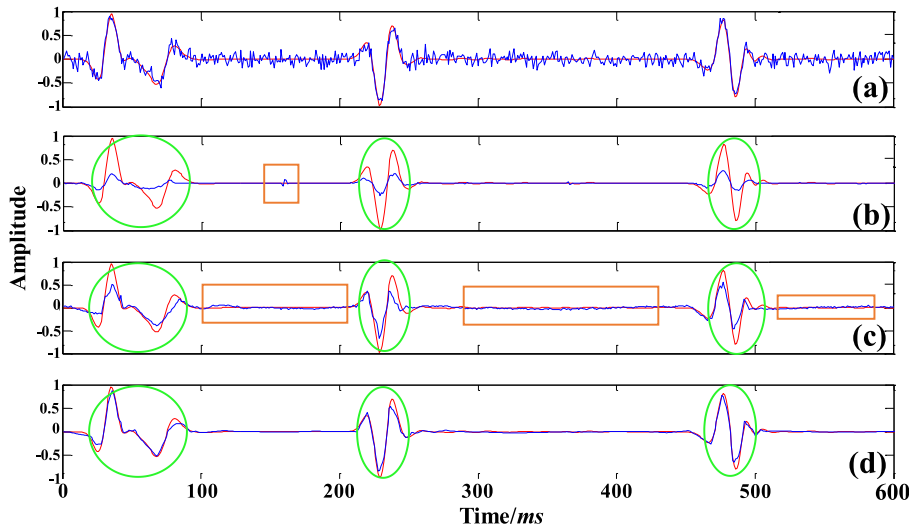
Through the first stage of the experiment, we found that the results of noise estimation are not only related to the noise level in the seismic data, but are also related to the complexity of the seismic data. Therefore, in the denoising experiment, we divided the test data into three parts: synthetic, simple and complex. Finally, we compared the DnCNN and FFDNet models with the MCD-DCNN model proposed in this paper.

### 1) DENOISING EXPERIMENT OF SINGLE-TRACE SEISMIC DATA

First, the characteristics of seismic single-trace data only involve the horizontal and vertical directions. The data structure is relatively simple. Second, the seismic single-trace data are a one-dimensional input, which does not involve the



**FIGURE 12.** The RMSE of the noise estimation of different methods. (a) The synthesized seismic profile data. (b) The submerged and simple seismic data. (c) The deep and complex seismic data. The figure shows the test results of 10 seismic profiles of each type of data. The noise level  $\sigma$  gradually rises from 10 to 70.

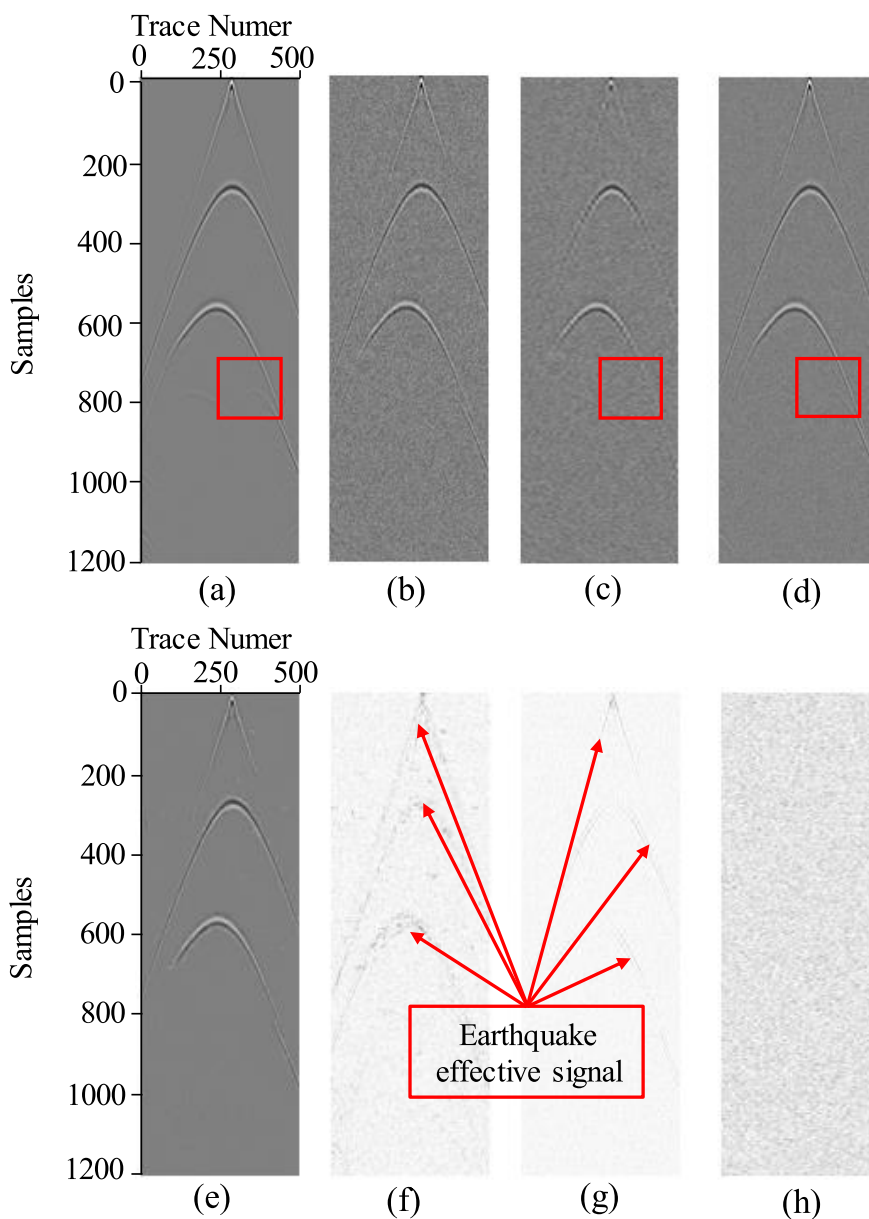


**FIGURE 13.** Comparison of the denoising effect of single trace data. (a) The original noisy data. (b) The denoising result of the DnCNN. (c) The denoising result of the FFDNet. (d) The denoising result of the MCD-DCNN. The red curve is the effective seismic signal data. The blue curve in (a) is the noisy data, and the blue in (b) and (c) are the clean signal data after denoising.

feature association of adjacent seismic traces. These characteristics make it easy to denoise this type of seismic data. The specific denoising effect is shown in Fig. 13.

Although each of the three methods denoised the data, their denoising effects are different. This is mainly manifested in two aspects. First, data with relatively large seismic

signal amplitude changes, which usually correspond to the reflection interface of seismic waves, have different denoising effects. The details are shown in the green ellipses and circles in Fig. 13. Although the DnCNN retains the general amplitude trend, it excessively suppresses effective seismic signals due to excessive denoising.



**FIGURE 14.** The denoising effect of the seismic profile data. (a) Clean data. (b) Noisy data. (c) The DnCNN denoising effect. (d) The FFDNet denoising effect. (e) The MCD-DCNN denoising effect. (f) The noise removed by the DnCNN. (g) The noise removed by the FFDNet. (h) The noise removed by the MCD-DCNN.

The denoising effect of the FFDNet is better than that of the DnCNN. This is due to the input of the initial noise that is mixed with seismic noise data. The FFDNet model first sets an initial value of the noise level  $\hat{\sigma}$ , and then continuously adjusts the initial value through residual learning and artificially set thresholds. Finally, through repeated iterations, the model achieves the best denoising performance [34]. This method of data processing alleviates noise level problems in the unclear and noisy data, which would otherwise lead to the degradation of the model's denoising performance. However, this process requires a very large amount of time to repeatedly optimize the initial value of the noise level through

the threshold, which reduces the efficiency of denoising. The process of optimizing the initial value of the noise level by the FFDNet is accompanied by denoising. In addition, the setting of the initial value greatly affects the denoising performance of the FFDNet. The orange rectangle in Fig. 13 shows the shortcoming of FFDNet model. Due to the initial noise level setting problem, FFDnet has a poor denoising effect during data smoothing. Since the MCD-DCNN successfully evaluates the noise level, the denoising ability of the model reaches its optimal value. In addition, no incomplete denoising occurred in the DnCNN and FFDNet models when smoothing the amplitude data.

## 2) DENOISING EXPERIMENT OF SIMPLE SEISMIC PROFILE DATA

Although single-trace seismic data are easier to denoise, this process is inconsistent when denoising actual seismic data. In seismic exploration, data are generally processed and displayed in two-dimensional and three-dimensional forms. In addition, the idea of processing a single trace and then composing the profile abandons the horizontal characteristics of seismic data. The underground strata not only has vertical characteristics but also horizontal characteristics, which are more important. Therefore, in this stage of the experiment, we input multiple data together, strengthen the extraction of horizontal features, and complete the denoising of seismic profile data. This stage of the experiment is divided into denoising two types of profile data: simple and complex. When dealing with simple seismic profile data denoising, we select 100 synthetic profile data and 100 actual profile data. Each seismic profile data contains 500 seismic traces. Each seismic channel contains 1201 data, and the sampling interval is 2 ms. Among the data, we randomly select 10 profiles from the two types of data as the test set. Using synthetic seismic profile data as an example, the denoising effects of the DnCNN, FFDNet and MCD-DCNN when the noise level is 30 are shown in Fig. 14 and Table 4.

**TABLE 4. Denoising effect of simple seismic profile data.**

| Noise Level | Methods  | PSNR (dB) | RMSE |
|-------------|----------|-----------|------|
| 30          | DnCNN    | 21.39     | 0.35 |
|             | FFDNet   | 23.51     | 0.22 |
|             | MCD-DCNN | 27.49     | 0.20 |

We analyze the characteristics of each denoising method from three aspects of seismic waves: strong reflection interface, weak reflection interface and smooth data. For strongly reflective interfaces, the DnCNN maintains the basic shape of the seismic wave reflection interface when denoising. Although the DnCNN retains the trend of the amplitude data, the amplitude is compressed, which is a manifestation of excessive denoising. When this behavior is reflected on the seismic profile, the strong reflection interface is blurred and the details are shown in the green rectangle in Fig. 14. In the denoised data, we find obvious “traces” of the seismic reflection interface. The “trace” refers to the position indicated by the red arrow in Fig. 14(f). The weak reflection interface (as shown by the red rectangle in Fig. 14) was not retained but was instead directly removed by the DnCNN model. For the entire profile, the DnCNN model does not completely denoise the data as there is residual noise visible to the naked eye.

The denoising effect of the FFDNet is better than that of the DnCNN. The denoising effect of the FFDNet and the corresponding noise that is removed are shown in Fig. 14(d) and Fig. 14(g). When the FFDNet is denoised, the strong reflection interface remains relatively complete, and the weak reflection interface can also be retained, as shown in the red

rectangle in Fig. 14(d). There are only a few “traces” of the strong reflection interface in the removed noise. In the entire profile, there is only a small amount of residual noise. In general, the denoising effect of the FFDNet is within the acceptable range.

The denoising effect and amount of noise removed by the MCD-DCNN are shown in Fig. 14(e) and Fig. 14(h). Compared with the original clean data, the strong and weak reflection interfaces of the MCD-DCNN after noise removal can be completely retained. In the removed noise, no “trace” of the effective signal was found. This shows that the MCD-DCNN protects the original effective signal data to the utmost extent when denoising.

## 3) DENOISING EXPERIMENT OF COMPLEX PROFILE DATA

For complex seismic profile data, denoising becomes difficult due to the lack of obvious laws in the reflection interface. In this stage of the experiment, the dataset was replaced with a complex seismic profile. The number and structure of the dataset are the same as those of the simple seismic data. Similar to the denoising experiment of the simple seismic profile data, we randomly select 10 profiles as the test set. For the DnCNN and FFDNet, we clearly provide the noise level of each seismic profile data. For the MCD-DCNN, we do not specify the noise level in the data to test the difference between these models in an ideal state. The results of the test are shown in Table 5.

From Table 5, we can see that when the noise level in the noisy seismic data increases, the denoising performance of the DnCNN, FFDNet and MCD-DCNN models decrease. When the noise level is 30, the PSNR of the DnCNN, FFDNet and MCD-DCNN models in the simple data set denoising test are 21.39 dB, 23.51 dB, 27.49 dB, respectively. The PSNR of the denoising test on the complex data set are 18.83 dB, 21.91 dB, 22.02 dB, respectively. When the noise level of seismic data are given, the DnCNN has a better denoising effect on low-noise seismic data. For example, when the noise level is 10 and 20, the PSNR is 26.07 dB and 21.24 dB, respectively, and the denoising effect is above 20 dB. As the noise level in seismic data increases, the denoising performance of DnCNN declines rapidly. When the noise level is 70, the PSNR is only 9.21 dB.

Due to the accurate initial noise value, the denoising performance of the FFDNet can reach its best state. When the noise level in the seismic data are less than 50, the PSNR of the FFDNet is above 20 dB. Compared with the DnCNN, the denoising effect of the FFDNet is obviously better. As the noise level in the data increases, the performance gap becomes increasingly obvious. Since there is no noise level in the given data, the MCD-CNN needs to estimate the noise in the test set. The error of noise estimation will comparatively affect the denoising performance of the model. However, it can be seen from Table 5 that the denoising effect of the MCD-DCNN is slightly better than the FFDNet because we use multichannel convolutions and dense connections in the denoising module of MCD-DCNN. The features

**TABLE 5. PSNR of complex seismic profile data denoising results.**

| Profile number | 1             | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | Average value |
|----------------|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------------|
| Noise level    | $\sigma = 10$ |       |       |       |       |       |       |       |       |       |               |
| DnCNN (dB)     | 25.77         | 25.76 | 25.76 | 26.43 | 26.41 | 26.46 | 26.04 | 26.02 | 26.03 | 26.02 | 26.07         |
| FFDNet (dB)    | 27.38         | 27.39 | 27.38 | 27.90 | 27.88 | 27.93 | 27.64 | 27.62 | 27.63 | 27.62 | <b>27.64</b>  |
| MCD-DCNN (dB)  | 27.41         | 27.31 | 27.45 | 27.43 | 27.87 | 27.54 | 27.49 | 27.77 | 27.89 | 27.93 | 27.61         |
| Noise level    | $\sigma = 20$ |       |       |       |       |       |       |       |       |       |               |
| DnCNN (dB)     | 21.22         | 21.22 | 21.22 | 21.24 | 21.26 | 21.27 | 21.25 | 21.23 | 21.25 | 21.24 | 21.24         |
| FFDNet (dB)    | 24.91         | 24.91 | 24.91 | 24.91 | 24.92 | 24.94 | 24.91 | 24.90 | 24.91 | 24.91 | <b>24.91</b>  |
| MCD-DCNN (dB)  | 24.93         | 24.87 | 24.82 | 24.32 | 24.96 | 24.90 | 24.96 | 24.93 | 24.95 | 24.65 | 24.83         |
| Noise level    | $\sigma = 30$ |       |       |       |       |       |       |       |       |       |               |
| DnCNN (dB)     | 18.85         | 18.84 | 18.83 | 18.79 | 18.80 | 18.80 | 18.85 | 18.84 | 18.85 | 18.83 | 18.83         |
| FFDNet (dB)    | 21.94         | 21.93 | 21.94 | 21.88 | 21.89 | 21.89 | 21.91 | 21.91 | 21.90 | 21.91 | 21.91         |
| MCD-DCNN (dB)  | 22.16         | 22.27 | 22.06 | 21.92 | 21.91 | 21.94 | 21.89 | 22.07 | 21.96 | 22.02 | <b>22.02</b>  |
| Noise level    | $\sigma = 40$ |       |       |       |       |       |       |       |       |       |               |
| DnCNN (dB)     | 17.87         | 17.87 | 17.87 | 17.84 | 17.85 | 17.83 | 17.86 | 17.86 | 17.87 | 17.86 | 17.86         |
| FFDNet (dB)    | 20.04         | 20.05 | 20.06 | 20.01 | 20.00 | 20.00 | 20.01 | 20.01 | 20.01 | 20.02 | 20.02         |
| MCD-DCNN (dB)  | 20.09         | 19.98 | 20.14 | 19.99 | 20.04 | 20.09 | 20.06 | 20.15 | 20.11 | 20.05 | <b>20.07</b>  |
| Noise level    | $\sigma = 50$ |       |       |       |       |       |       |       |       |       |               |
| DnCNN (dB)     | 13.87         | 13.87 | 13.87 | 13.84 | 13.86 | 13.84 | 13.85 | 13.86 | 13.86 | 13.86 | 13.86         |
| FFDNet (dB)    | 18.34         | 18.34 | 18.34 | 18.34 | 18.34 | 18.34 | 18.34 | 18.34 | 18.34 | 18.34 | 18.34         |
| MCD-DCNN (dB)  | 18.42         | 18.50 | 18.47 | 18.49 | 18.38 | 18.44 | 18.46 | 18.40 | 18.43 | 18.42 | <b>18.44</b>  |
| Noise level    | $\sigma = 60$ |       |       |       |       |       |       |       |       |       |               |
| DnCNN (dB)     | 12.41         | 12.40 | 12.41 | 12.38 | 12.39 | 12.40 | 12.37 | 12.38 | 12.37 | 12.38 | 12.39         |
| FFDNet (dB)    | 16.88         | 16.86 | 16.85 | 16.81 | 16.84 | 16.82 | 16.81 | 16.83 | 16.82 | 16.84 | 16.84         |
| MCD-DCNN (dB)  | 16.91         | 16.95 | 16.95 | 16.93 | 16.86 | 16.87 | 16.90 | 16.90 | 16.96 | 16.92 | <b>16.92</b>  |
| Noise level    | $\sigma = 70$ |       |       |       |       |       |       |       |       |       |               |
| DnCNN (dB)     | 9.24          | 9.23  | 9.24  | 9.21  | 9.20  | 9.19  | 9.19  | 9.21  | 9.19  | 9.21  | 9.21          |
| FFDNet (dB)    | 13.67         | 13.67 | 13.66 | 13.62 | 13.61 | 13.60 | 13.62 | 13.64 | 13.63 | 13.64 | 13.64         |
| MCD-DCNN (dB)  | 13.80         | 13.76 | 13.81 | 13.74 | 13.78 | 13.67 | 13.76 | 13.73 | 13.74 | 13.78 | <b>13.76</b>  |

that are extracted by multiscale convolutions are richer, and the dense connections can make more effective use of the extracted features. These measures can effectively improve the denoising performance of the model.

To fully verify the denoising performance of the MCD-DCNN, we fix the noise level of the data in the test set at 40. To save time, we only compare with the FFDNet. In the experiment at this stage, we did not set the FFDNet noise level of the seismic data. Instead, we set the initial value of the FFDNet to 10, and then denoised the seismic data with a noise level of 40. Then, we gradually increased the initial value until 70. The results of the test are shown in Table 6.

From Table 6, we can see that when the initial noise level setting does not match the real noise level in the data, the denoising performance of the FFDNet decreases sharply. When the initial noise level and the actual noise level are the same, the denoising performance of the FFDNet will be most optimal. This shortcoming limits the practical application of the FFDNet in seismic data denoising. When denoising the actual seismic data, it is impossible for us to know the true noise level of the seismic data, and it is impossible to

**TABLE 6. Noise reduction effect without prior noise level.**

| Method   | Initial Noise Level | True Noise Level | PSNR (dB)    |
|----------|---------------------|------------------|--------------|
| FFDNet   | 10                  | 40               | 17.78        |
|          | 20                  |                  | 18.52        |
|          | 30                  |                  | 19.45        |
|          | 40                  |                  | <b>20.03</b> |
|          | 50                  |                  | 18.98        |
|          | 60                  |                  | 16.78        |
|          | 70                  |                  | 13.23        |
| MCD-DCNN | None                |                  | <b>20.07</b> |

set the initial noise level from low to high and test each profile. When the MCD-DCNN model proposed in this paper processes seismic data with a noise level of 40, the noise level is evaluated (the average evaluation value of 10 profiles is 40.17), and the flexibility of denoising is far greater than that of the FFDNet.

**C. EXPERIMENTAL DISCUSSION**

The purpose of this experiment is to restore effective seismic signals as much as possible according to the characteristics

of random noise in seismic data. Random noise has wide frequency bands and complex components and is widely distributed in effective seismic signals. In this regard, we proposed the MCD-DCNN model. The advantage of this model is that it can evaluate the noise in the seismic data and use multiscale convolutions to extract more data features. In addition, this model also uses dense connections to make full use of the extracted features, and improves the denoising performance of the model. On the basis of the above, we used single-trace, simple and complex data to simulate the seismic denoising process in the real environment to test the model. For seismic profile data, the denoising performance of the FFDNet and MCD-DCNN is comparable for a given noise level in the data. The DnCNN's denoising effect is slightly worse. The FFDNet requires the noise level in the given data, which is not feasible for actual seismic data denoising.

Whether it is the single-trace data denoising method or the noise level in the given data, these situations do not conform to the actual seismic data denoising. In this regard, we tested the FFDNets and DnCNNs again without specifying the noise level. The test results show that the denoising performance of the FFDNet is greatly reduced, while the denoising performance of the MCD-DCNN is not affected.

In general, the MCD-DCNN model proposed in this paper has achieved certain effects in denoising seismic data, but it also has shortcomings. First, the noise estimation and denoising modules in the model require a large amount of on-site seismic data for training. Therefore, data denoising can only be carried out after the field acquisition of seismic exploration is completed. The MCD-DCNN cannot achieve real-time denoising in the field. In addition, the general trend of seismic exploration data is three-dimensional, while the MCD-DCNN model proposed in this paper is two-dimensional. How to apply the model to 3D is our future research direction.

## VI. CONCLUSION

This paper proposes the MCD-DCNN model for seismic data denoising, which is divided into two modules: noise estimation and denoising. In the noise evaluation module, we use multiscale convolutions to enrich the features of the extracted seismic data. In addition, the noise estimation module uses dense connections to make full use of the extracted features. These methods improve the quasi-curvature of noise evaluation. In the denoising module, the noise evaluation solves the problem of prior noise level when the existing methods are used to denoise seismic data. On this basis, we combined multiscale convolutions and dense connections to greatly improve the flexibility and noise capabilities of the model.

We tested the model with single-trace, simple and complex data. In addition, we compared the MCD-DCNN model with the more commonly used DnCNN and FFDNet models. The comparison results show that, given the noise level in the

seismic data, the denoising performance of the MCD-DCNN and FFDNet are comparable. Finally, we simulated the real environment of field seismic exploration. The simulation results show that the denoising performance of the MCD-DCNN model is far greater than that of FFDNet.

## REFERENCES

- [1] X. Dong, T. Zhong, and Y. Li, "New suppression technology for low-frequency noise in desert region: The improved robust principal component analysis based on prediction of neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4680–4690, Jul. 2020, doi: [10.1109/TGRS.2020.2966054](https://doi.org/10.1109/TGRS.2020.2966054).
- [2] X. Dong, H. Jiang, S. Zheng, Y. Li, and B. Yang, "Signal-to-noise ratio enhancement for 3C downhole microseismic data based on the 3D shearlet transform and improved back-propagation neural networks," *Geophysics*, vol. 84, no. 4, pp. V245–V254, Jul. 2019, doi: [10.1190/GEO2018-0621.1](https://doi.org/10.1190/GEO2018-0621.1).
- [3] T. P. Banjade, S. Yu, and J. Ma, "Earthquake accelerogram denoising by wavelet-based variational mode decomposition," *J. Seismol.*, vol. 23, no. 4, pp. 649–663, Jul. 2019, doi: [10.1007/s10950-019-09827-0](https://doi.org/10.1007/s10950-019-09827-0).
- [4] W. Chen, H. Song, and X. Chuai, "Fully automatic random noise attenuation using empirical wavelet transform," *J. Seismic Explor.*, vol. 28, no. 2, pp. 147–162, Apr. 2019, doi: [10.1088/1742-2132/13/3/378](https://doi.org/10.1088/1742-2132/13/3/378).
- [5] R. Mortezaejad and A. Gholami, "Optimization of wavelet- and curvelet-based denoising algorithms by multivariate SURE and GCV," *J. Geophys. Eng.*, vol. 13, no. 3, pp. 378–390, Jun. 2016, doi: [10.1088/1742-2132/13/3/378](https://doi.org/10.1088/1742-2132/13/3/378).
- [6] Y. Jeong, B. Lee, J.-H. Han, and J. Oh, "Ocular axial length prediction based on visual interpretation of retinal fundus images via deep neural network," *IEEE J. Sel. Topics Quantum Electron.*, vol. 27, no. 4, pp. 1–7, Jul. 2021, doi: [10.1109/JSTQE.2020.3038845](https://doi.org/10.1109/JSTQE.2020.3038845).
- [7] Q. Cao, W. Zhang, and Y. Zhu, "Deep learning-based classification of the polar emotions of 'moe'-style cartoon pictures," *Tsinghua Sci. Technol.*, vol. 26, no. 3, pp. 275–286, Jun. 2021, doi: [10.26599/TST.2019.9010035](https://doi.org/10.26599/TST.2019.9010035).
- [8] R. Elshawi, A. Wahab, A. Barnawi, and S. Sakr, "DLBench: A comprehensive experimental evaluation of deep learning frameworks," *Cluster Comput.*, vol. 24, no. 3, pp. 2017–2038, Sep. 2021, doi: [10.1007/s10586-021-03240-4](https://doi.org/10.1007/s10586-021-03240-4).
- [9] S. Yu, J. Ma, and W. Wang, "Deep learning for denoising," *Geophysics*, vol. 84, no. 6, pp. V333–V350, Nov. 2019, doi: [10.1190/GEO2018-0668.1](https://doi.org/10.1190/GEO2018-0668.1).
- [10] X. He, F. Qian, Y. Fan, L. Feng, and G. Hu, "3-D seismic noise attenuation via tensor sparse coding with spatially adaptive coherence constraint," *IEEE Access*, vol. 9, pp. 12217–12229, 2021, doi: [10.1109/ACCESS.2021.3051069](https://doi.org/10.1109/ACCESS.2021.3051069).
- [11] W. Li and J. Wang, "Residual learning of cycle-GAN for seismic data denoising," *IEEE Access*, vol. 9, pp. 11585–11597, 2021, doi: [10.1109/ACCESS.2021.3049479](https://doi.org/10.1109/ACCESS.2021.3049479).
- [12] L. Yang, W. Chen, W. Liu, B. Zha, and L. Zhu, "Random noise attenuation based on residual convolutional neural network in seismic datasets," *IEEE Access*, vol. 8, pp. 30271–30286, 2020, doi: [10.1109/ACCESS.2020.2972464](https://doi.org/10.1109/ACCESS.2020.2972464).
- [13] F. Wang and S. Chen, "Residual learning of deep convolutional neural network for seismic random noise attenuation," *IEEE Geoscience Remote Sens. Lett.*, vol. 16, no. 8, pp. 1314–1318, Jun. 2019, doi: [10.1109/LGRS.2019.2895702](https://doi.org/10.1109/LGRS.2019.2895702).
- [14] W. Sang, S. Yuan, X. Yong, X. Jiao, and S. Wang, "DCNNs-based denoising with a novel data generation for multidimensional geological structures learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 10, pp. 1861–1865, Oct. 2021, doi: [10.1109/LGRS.2020.3007819](https://doi.org/10.1109/LGRS.2020.3007819).
- [15] S. Wang, Y. Li, N. Wu, Y. Zhao, and H. Yao, "Attribute-based double constraint denoising network for seismic data," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5304–5316, Jun. 2021, doi: [10.1109/TGRS.2020.3021492](https://doi.org/10.1109/TGRS.2020.3021492).
- [16] G. H. Li and Y. Li, "Random noise of seismic exploration in desert modeling and its applying in noise attenuation," *Chin. J. Geophys.*, vol. 59, no. 2, pp. 682–692, Feb. 2016, doi: [10.6038/cjg20160225](https://doi.org/10.6038/cjg20160225).
- [17] M. Li, Y. Li, N. Wu, and Y. Tian, "Desert seismic data denoising based on energy spectrum analysis in empirical curvelet domain," *Studia Geophysica et Geodaetica*, vol. 64, no. 3, pp. 373–390, Jul. 2020, doi: [10.1007/s11200-019-0476-4](https://doi.org/10.1007/s11200-019-0476-4).

- [18] Y. Gao, P. Zhao, G. Li, and H. Li, "Seismic noise attenuation by signal reconstruction: An unsupervised machine learning approach," *Geophys. Prospecting*, vol. 69, no. 5, pp. 984–1002, Jun. 2021, doi: [10.1111/1365-2478.13070](https://doi.org/10.1111/1365-2478.13070).
- [19] J. Ward, M. Thorne, A. Nowacki, and S. Rost, "Automatic slowness vector measurements of seismic arrivals with uncertainty estimates using bootstrap sampling, array methods and unsupervised learning," *Geophys. J. Int.*, vol. 226, no. 3, pp. 1847–1857, Jun. 2021, doi: [10.1093/gji/ggab196](https://doi.org/10.1093/gji/ggab196).
- [20] O. M. Saad, K. Inoue, A. Shalaby, L. Samy, and M. S. Sayed, "Automatic arrival time detection for earthquakes based on stacked denoising autoencoder," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1687–1691, Nov. 2018, doi: [10.1109/LGRS.2018.2861218](https://doi.org/10.1109/LGRS.2018.2861218).
- [21] M. Titos, A. Bueno, L. García, and C. Benítez, "A deep neural networks approach to automatic recognition systems for volcano-seismic events," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 5, pp. 1533–1544, May 2018, doi: [10.1109/JSTARS.2018.2803198](https://doi.org/10.1109/JSTARS.2018.2803198).
- [22] Q. Feng, Y. Li, and H. Wang, "Intelligent random noise modeling by the improved variational autoencoding method and its application to data augmentation," *Geophysics*, vol. 86, no. 1, pp. T19–T31, Jan. 2021, doi: [10.1190/Geo2019-0815.1](https://doi.org/10.1190/Geo2019-0815.1).
- [23] Y. X. Zhao, Y. Li, and B. J. Yang, "Denoising of seismic data in desert environment based on a variational mode decomposition and a convolutional neural network," *Geophys. J. Int.*, vol. 221, no. 2, pp. 1211–1225, May 2020, doi: [10.1093/gji/ggaa071](https://doi.org/10.1093/gji/ggaa071).
- [24] X. Si, Y. Yuan, T. Si, and S. Gao, "Attenuation of random noise using denoising convolutional neural networks," *Interpretation*, vol. 7, no. 3, pp. SE269–SE280, Aug. 2019, doi: [10.1190/INT-2018-0220.1](https://doi.org/10.1190/INT-2018-0220.1).
- [25] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017, doi: [10.1109/TIP.2017.2662206](https://doi.org/10.1109/TIP.2017.2662206).
- [26] H. Jun, H.-T. Jou, C.-H. Kim, S. H. Lee, and H.-J. Kim, "Random noise attenuation of sparker seismic oceanography data with machine learning," *Ocean Sci.*, vol. 16, no. 6, pp. 1367–1383, Nov. 2020, doi: [10.5194/os-16-1367-2020](https://doi.org/10.5194/os-16-1367-2020).
- [27] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018, doi: [10.1109/TIP.2018.2839891](https://doi.org/10.1109/TIP.2018.2839891).
- [28] Y. Li, Z. Zhao, Y. Tian, and Q. Feng, "Multi-scale DCF network: A new desert low-frequency noise suppression method," *Acta Geodaetica et Geophysica*, pp. 357–371, Apr. 2021, doi: [10.1007/s40328-021-00339-3](https://doi.org/10.1007/s40328-021-00339-3).
- [29] X. Tian, B. Zheng, S. Li, C. Yan, J. Zhang, Y. Sun, T. Shen, and M. Xiao, "Hard parameter sharing for compressing dense-connection-based image restoration network," *J. Electron. Imag.*, vol. 30, no. 5, pp. 1–13, Oct. 2021, doi: [10.1117/1.JEI.30.5.053025](https://doi.org/10.1117/1.JEI.30.5.053025).
- [30] Y. Zhou, M. Zhang, J. Zhu, R. Zheng, and Q. Wu, "A randomized block-coordinate Adam online learning optimization algorithm," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12671–12684, Aug. 2020, doi: [10.1007/s00521-020-04718-9](https://doi.org/10.1007/s00521-020-04718-9).
- [31] Y. Zhang, H. Lin, Y. Li, and H. Ma, "A patch based denoising method using deep convolutional neural network for seismic image," *IEEE Access*, vol. 7, pp. 156883–156894, 2019, doi: [10.1109/ACCESS.2019.2949774](https://doi.org/10.1109/ACCESS.2019.2949774).
- [32] Z. Wen, H. Wang, Y. Gong, and J. Wang, "Denoising convolutional neural network inspired via multi-layer convolutional sparse coding," *J. Electron. Imag.*, vol. 30, no. 2, pp. 1–14, Mar. 2021, doi: [10.1117/1.JEI.30.2.023007](https://doi.org/10.1117/1.JEI.30.2.023007).
- [33] Z. Li and J. Wu, "Learning deep CNN denoiser priors for depth image inpainting," *Appl. Sci.*, vol. 9, no. 6, p. 1103, Mar. 2019, doi: [10.3390/app9061103](https://doi.org/10.3390/app9061103).
- [34] Z. Gao, Y. Wang, X. Liu, F. Zhou, and K.-K. Wong, "FFDNet-based channel estimation for massive MIMO visible light communication systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 3, pp. 340–343, Mar. 2020, doi: [10.1109/LWC.2019.2954511](https://doi.org/10.1109/LWC.2019.2954511).

...