Semi-Supervised Learning-Based Image Denoising for Big Data

KUN ZHANG\textsuperscript{1} AND KAI CHEN\textsuperscript{2}

\textsuperscript{1}College of Intelligent Equipment, Shandong University of Science and Technology, Tai’an 271019, China
\textsuperscript{2}Taishan Institute of Technology, Shandong University of Science and Technology, Tai’an 271000, China

Corresponding author: Kun Zhang (skd993454@sdust.edu.cn)

\textbf{ABSTRACT} In this paper, the research of image noise reduction based on semi-supervised learning is carried out, and the neural network is used to reduce the noise of the image, so as to achieve more stable and good image display ability. Based on the convolutional neural network algorithm, the role of activation function optimization network is studied, combined with semi-supervised learning modes such as multi-feature extraction technology, to learn and extract the key features of the input image. Semi-supervised residual learning based on convolutional network is a good image denoising and denoising network model. Compared with other excellent denoising algorithms, it has very good results. At the same time, it greatly improves the image noise pollution and makes the image details clearer. At the same time, compared with other image denoising algorithms, this algorithm can show a good peak signal-to-noise ratio under various noise standard deviations. Through the research in this article, it is verified that the improved convolutional neural network denoising model and multi-feature extraction technology have strong advantages in image denoising.

\textbf{INDEX TERMS} Semi-supervised learning, big data, image denoising.

\section{I. INTRODUCTION}

With the development of science and technology, the use of computers to process images has become a common method and technique, such as denoising, enhancing, and restoring images [1]. Today, information communication is highly developed and connected across the globe. People usually collect different kinds of data, either passively or actively [2]. According to some statistics, most of the data people receive is visual information and it is of great relevance to study how to better remove image noise [3]. The problem of noise pollution of images occurs due to defects or imperfections in imaging equipment or information systems [4]. In the process of image information propagation, the existence of noise leads to low image clarity, which not only makes people’s vision deviations, but also leads to the loss of some of the detailed information in the image, and thus has a certain negative impact on the clarity and value of the image [5]. The appearance of noisy images will also adversely affect the subsequent image processing, including image recognition and transmission, extraction, and segmentation, etc. [6].

Therefore, with the help of scientific algorithms, the image processing of noisy images can be improved [7]. Therefore, it is necessary to reduce the noise with the help of scientific algorithms. For the human visual nerves, the de-noised images can show clearer and more accurate intent, which helps visual recognition and cognition [8]. Because of various internal and external factors, it is difficult to find a noise-free image that has varying degrees of contamination, which makes it difficult to distinguish relevant details [9]. Especially for medical and security images, noise reduction techniques are needed to preserve the details of the image [10]. If the noise in the image map is very serious, then it will also bring the loss of the original information, precisely because of the many hazards of noise, then noise reduction and denoising are very important [11]. For the image processing process, the image denoising process can get more high-definition image map, which also lays a good foundation for subsequent related applications [12]. Image noise reduction is often the first part of its information processing, with the help of noise reduction can further improve the quality of the image map, which also lays a solid foundation for the subsequent processing results [13]. Traditional image noise reduction methods have been proposed for a long time now, however,
most of those algorithms are not very satisfactory, especially when denoising, the image details are lost, absolutely most of them [14].

Currently, image noise reduction processing is an important part of image vision or image processing [15]. At the beginning of the last century, digital graphics were successfully compressed and transmitted based on submarine cables [16]. In the early stages of development, the purpose of the relevant image processing technology was to make it easier to apply images [17]. With the rapid development of computers, the Internet, and communication technologies, theoretical research, practical applications, and related technologies of image processing were gradually improved and eventually developed into a new discipline [18]. Scholars such as Kumar N, who have continuously optimized the algorithms of neural networks [19]. Professors Ren J, using relevant foreign research theories, combined with their own practical experience, began to carry out technical research in computer deep learning, and made great progress, with important influence in the relevant image denoising processing technology [20]. Ahmed MU and other scientists based on relevant research proposed a new form of technology: this adaptive multi-row deep BP, and these scholars state that the BP algorithm is tuned, which is also a learning method for self-coding, based on performing functional self-coding calculations on fuzzy and defective digital images, and then implementing noise reduction and de-noising of the images [21]. This denoising method is combined with a neural network structure to form a deep learning-based image denoising processing method [22]. Based on the theory, multiple trial and error and training of the denoised model can be carried out, and good recovery and denoising effects can be obtained, while the method also has image reconstruction algorithms, such as the nonlinear response diffusion model, which implements trainable algorithmic improvements, making very good results in image denoising and its recovery processing [23]. Kollem S introduced the concept of generative adversarial networks, a theory that has had a major impact on the entire image denoising process [24]. In Tao JH’s design, the generative adversarial network is functionally divided into two sub-networks, one for construction and the other for evaluation of construction quality, both of which further improve the quality of image data information through continuous feedback [25]. The algorithm provides a state of mind. It has two models, the generative and discriminative models [26]. The former can be used for training and learning, while the latter can discern whether the generated model is close to reality until it cannot be judged as true [27]. It is a zero-sum game [28]. Also, good results have been achieved in image recognition. In the last two years, some scholars have started to use this technique to deny images and have achieved amazing results [29].

The image to be processed is input and the data is compressed by image pre-processing, then feature extraction is performed, and after image segmentation, the image is usually used for image recognition and other related applications. To complete the subsequent image processing process, the first part is the pre-processing of the image, and this is where the image is demised to obtain as many information features of the image as possible, which manifests itself in the visual perception that the image is clearer, which is the basis for the subsequent processing of the image. It is not difficult to see that the image denoising technology has a very important impact on the application of the image, and is a decisive factor in the clarity of the image and the application of the effect, but also the basic conditions for subsequent processing. With the advancement of computing, it is obvious that compared with traditional denoising algorithms, neural network-based image denoising methods have shown greater improvement in denoising image details and denoising image performance indicators. However, various new challenges will inevitably be encountered in the process of continuous improvement of the algorithm. For example, deep learning denoising algorithms have too many network layers, and the corresponding model training difficulty will increase, while gradients are more prone to discrete effects, so it is difficult to converge, the learning effect is poor, and the corresponding improvement of the image noise reduction effect will become difficult. It is these problems that arise, this study of the image map noise reduction algorithm based on deep learning, and explores a more optimal model for training the denoising model, and conducts a systematic study on the optimization techniques for the image denoising affect this aspect of the study is very interesting and practical.

II. IMAGE DENOISING DESIGN FOR SEMI-SUPERVISED LEARNING

A. IMPROVED SEMI-SUPERVISED LEARNING ALGORITHM DESIGN

General denoising algorithms can be divided into two categories according to their respective fields. Firstly, noise is removed by obtaining image feature information rules from the image perspective; secondly, the disturbing image map is recovered from an early evidence perspective with the help of image prior knowledge [30]. Thirdly, the image map is weighted and leveled employing space and frequency domain subdivision, and then the Gaussian noise is eliminated to achieve the corresponding Gaussian filter. This approach may result in a blurred image [31]. Non-linear filtering currently uses the median filter and other related logical operation techniques to implement, maximum filter, minimum filter, etc. These algorithms are based on the gray area of the comparison field. This kind of algorithm is based on the comparison of the grayscale values of the field, the overall design strategy algorithm is quite simple, and the effect is not very outstanding, but it is very convenient. The purpose of frequency, as opposed to spatial filtering, is to approximate the original function so that the corresponding coefficients can be obtained, and frequency domain filtering methods optimized for specific applications have been successfully used. For wavelet transform, many improved algorithms have
been applied to image denoising, but wavelet filtering is not effective in high-dimensional images. Currently, the denoising effect of this algorithm is different from other excellent algorithms. With the current rapid development of the cross-fusion of various disciplines based on the integration of mathematics, computer-related technology has been greatly enhanced, which has promoted the denoising-related algorithms in all aspects of a more substantial breakthrough, and some new algorithms have emerged, such as genetic algorithms, training dictionary algorithms, 3D block matching algorithms, ultra-complete terrain sparse coding, OTSC and other algorithms have made great progress in image denoising. As shown in Figure 1.

The algorithm is used to solve the following matrices [32].

$$Y = MX$$  \hspace{1cm} (1)

For the noise reduction model, from formula (2).

$$Y = X + D$$  \hspace{1cm} (2)

Our goal is to remove as much noise D as possible and recover X. How the process from a noisy image to a clear image is implemented in the model structure. In the next section, we will focus on image block-based denoising. For noise pollution, we can understand the process from a functional perspective.

$$D = \eta(Y)$$  \hspace{1cm} (3)

The noise added here and later is uniformly Gaussian white noise, the specific reason is also given explicitly in the previous article. The noise reduction logic is the inverse process of adding noise, which, if analyzed at the functional level, can be understood as.

$$G = \arg \min f E_y \|f(x) - y\|^2$$  \hspace{1cm} (4)

Next, the decomposed noisy picture block processing flow is used to explain how the image block denoising model works, and the output of the demised picture m*n is achieved through the input of the noisy picture m*n. One point of concern here is that if the algorithm is directly based on them n mapping to complete the corresponding mapping map composition, the input learning and operation costs required at this point are much higher. Therefore, it is possible to decompose the noisy soil block to achieve the corresponding purpose. In the above figure, the noisy zebra image map with noise is decomposed to form N number of noisy blocks of P*P windows, which are then processed separately, with specific algorithms including BM3D (expected patch log-likelihood, EPL) [33]. The strategy of decomposing the block image chunks is also used and the neural network is also decomposed into multiple patches for feature extraction. It is also this pattern of splitting and eventually aggregating. This can reduce the computational effort considerably and has better results.

The convolutional self-coding designed here is similar to the deep U-shaped convolutional denoising module of SMTR Net from Chapter 3, but unlike the denoising module, the self-coding structure in the semi-supervised convolutional
self-coding algorithm is meant to maximize the extraction of information that the intermediate layer can characterize the target, whereas DUBD Net is meant to reconstruct the final demised signal and is more concerned with preserving the structure of the source data from the input corrupted data information rather than focusing on whether the information in the middle layer is lost, and secondly [34], DUBD Net has two more fusion layers than the convolutional self-coding network in this section, which we consider this fusion layer is beneficial for reconstructing the information channel before and after the signal, but its cross-layer linking feature causes the middlemost hidden layer to be unable to obtain the complete target information. Therefore, the convolutional self-coding model designed in this section removes the fusion layer from DUBD Net and retains the other structural information.

In reconstructing the output, a weighted sum of the KL scatter and the minimum mean squared error function is designed as a loss function, where the KL scatter is used to constrain the HRRP distribution and the minimum mean squared error is used to locate the scatter point location. Its expression is shown in formula (5).

$$L = (1 - a) L_{\text{mse}} + a L_{\text{KL}}$$  \hspace{1cm} \text{(5)}$$

In the semi-supervised learning process, the convolutional self-coding network is first pre-trained using all the data, enabling the convolutional self-coding network to capture the global sample distribution features. The output of the intermediate hidden layer is input to the convolutional neural networks model for fine-tuning.

The t-SNE algorithm is an improvement on the Stochastic Neighbor Embedding (SNE) algorithm, where the goal of SNE is to maintain the spatially distributed characteristics of the samples in a low-dimensional space. T-SNE improves on the Stochastic Neighbor Embedding algorithm by applying an affine transformation to the data point mapping of the samples, by transforming the Euclidean distance into a conditional probability to explicitly express the similarity between sample points.

$$k_{ji} = \frac{\exp(-\|x_i - y_j\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|x_i - z_k\|^2/2\sigma^2)}$$  \hspace{1cm} \text{(6)}$$

When we map the data to the low-dimensional space, the similarity between the high-dimensional data points should also be reflected in the data points in the low-dimensional space. Here also described in the form of conditional probability, if the high-dimensional sample points $x_i$ and $x_j$; mapped points in the low-dimensional space are $y_i$ and $y_j$; respectively. Similarly, the conditional probability in the low-dimensional space is denoted by $q$, and the variance $\sigma$ of all Gaussian distributions is set to 0.5, mathematically described as in formula (7).

$$q_{ji} = \frac{\exp(-\|x_i - y_j\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|x_i - z_k\|^2/2\sigma^2)}$$  \hspace{1cm} \text{(7)}$$

To make the sample distribution after dimensionality reduction and in high-dimensional space can be close enough to each other, the $p$ dispersion is chosen as the loss function of SNE in the optimization process, as shown in formula (8).

$$\text{loss} = \sum_{i} \sum_{j} p \log \frac{p}{q}$$  \hspace{1cm} \text{(8)}$$

However, KL dispersion is an asymmetric measure and, therefore, is sensitive to the cost of loss at different distances, so that SNE is more concerned with the preservation of local features of the sample. Based on SNE, Matane proposes asymmetric SNE algorithm, the conditional probability of expressing distance similarity is replaced by a joint probability distribution [35]. The symmetric measures of similarity in high and low dimensional space are shown in formula (9) and formula (10).

$$k_{ji} = \frac{\exp(-\|x_i - y_j\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|x_i - z_k\|^2/2\sigma^2)}$$  \hspace{1cm} \text{(9)}$$

$$k_{ji} = \frac{\exp(-\|x_i - y_j\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|x_i - z_k\|^2/2\sigma^2)}$$  \hspace{1cm} \text{(10)}$$

Then the improved SNE makes each sample point have an impact on the overall loss function, which improves the global nature of the SNE algorithm. However, in practical experiments, symmetric SNE still suffers from the problem of sample crowding in low-dimensional space, making different classes of clusters that can be separated in high-dimensional space inseparable in low-dimensional space. To visually describe the problem of sample crowding in low-dimensional embedding.

Suppose that the sample distribution space is a hypersphere with radius 1 centered at the $n$-dimensional sample point $r$. Then the probability distribution of the distance $r$ from $x$; at any point within the sphere is as in formula (11) and the cumulative probability distribution is as in formula (12).

$$M(r) = \lim_{\Delta r \to 0} \frac{(r + \Delta r)^n - r^n}{\Delta r}$$  \hspace{1cm} \text{(11)}$$

$$F(r) = \int_0^r p(r)dr = r^n$$  \hspace{1cm} \text{(12)}$$

As the dimensionality increases, the probability density near the sampling point becomes lower and higher near the surface of the sphere, and if this same distribution is presented in a low-dimensional space, the data are squeezed together to form a sample crowding problem.

B. BIG DATA IMAGE DATA SYSTEM DESIGN

In image recognition, the main problem is not only the insufficient acquisition of labeled target data but also the incomplete sampling of the target pose. For example, the proposed SemiCAE algorithm carries out unsupervised pre-training on the pose of unlabeled samples to improve the generalization ability of deep neural networks, which effectively alleviates the
impact of pose sensitivity and inadequate pose sampling on the recognition system. As shown in Figure 2, the manifold embedding visualization of the Semi-CAE penultimate fully connected layer under 1 sample taken at 5° intervals. The embedding of the Semi-CAE fully connected layer shows that samples with proximity in the high-dimensional manifold space are distributed together, which makes it easy to associate that samples in high-density clusters are likely to have the same label, consistent with the popular assumption of semi-supervised learning.

Convolutional neural networks are simulations of the biological visual cortex with deep structure, and the following will explain how in a convolutional neural network we get the desired output from the propagation of each layer. As shown in Figure 2, this is a proposed model architecture for a convolutional neural network. Firstly, to understand its basic working principle, when an image with noise interference attached to it is input, this image is sampled by the combined convolutional and activation layers to get the first extracted image features. This image then continues with the input and loops the previous operation, and is processed through the convolution and activation layers, as well as the pooling layer, to obtain a second extraction of image feature information. After the two-image feature information is connected and integrated and segmented, they are fused with the complete image features to obtain the feature information of the final image. Convolutional neural networks are different from traditional neural networks mainly due to the sparse connection and weight sharing approach, which makes the algorithm widely used in imaging-related fields, as this network mechanism can well reduce the computational load in the network domain and is also very suitable for image feature learning.

In the recognition and processing session of the image, with the help of the convolutional neural networks model, which mainly mimics biological visual features and therefore has good adaptability, the biological neural network does not capture all image features and all image textures while viewing the image. Instead, it holds the main features, just like looking at a human face. It only targets the eyes, lips, chin, and other prominent areas and features. With these important features, you can quickly tell who the person is. It is the perception of local information that eventually leads to global information, and the human eye is the basis of a one-time model to see what is relevant. Traditional BP will use matrix multiplication to express the relationship between input and output. Current networks are usually deep networks, corresponding to more network layers overall, with more nodes in the input and intermediate implicit layers. No matter what happens, if two adjacent layers of neurons are connected one after the other using matrix in the same way as if conventional neural networks, there is a great computational cost for the corresponding neural networks, making it difficult to train the relevant networks. Sparse connections in convolutional neural networks, on the other hand, will have sparse interactions between the corresponding neurons, when there is the concentration in terms of local connections, rather than one-by-one interactions. This computation will reduce the cost by many orders of magnitude. This is precisely why
reinforcement network training is extremely important, and it is quite intuitive to look at the following three figures, which show this treatment.

This paper proposes a deep label reconstruction based algorithm on this manifold assumption, which uses a deep pre-trained recognition model to pre-label unlabeled samples also known as Pseudo Label (PL), and then uses a deep neural network model to simultaneously regularize and pre-train unlabeled samples with labeled data and label reconstructions. The flow of the deep label reconstruction algorithm is shown in Figure 3.

What should be understood about the components, where self-similarity is important, BM3D is also designed with the help of self-similarity of image blocks compared to traditional good algorithms? The present backbone statistics in the image has a basic similarity with the statistical features of the other links, the basic image is not related to the location of the features in the image. The same is true for convolutional neural networks to learn the basic features of an image. The actual process is to give the input image and scan the image using the convolutional kernel. The values in the convolutional kernel are called weights. For a single convolutional kernel, the weights do not occur in network iterations. The changes we described earlier will only be updated during backpropagation. In other words, differential positions in the image are primarily based on the same convolutional kernels that are scanned. In this process, their weights are consistent, so that the weights can be shared.

This is extremely important for convolutional neural networks, as this sharing of weights greatly reduces the computational parameters of the overall network. For example, for an image map of $m^*n$, if you use the connection network, you can get a characteristic image map of $XY$, and then you need to calculate the parameters of $m^n x y$. If you enter an image size of 102, then you can get an image map of $XY$. If the input image size is about 102 and the order of magnitude of $XY$ and $m^n$ is not too different, then the network parameters for this layer need to be 108 to 1010. For a multilayer deep network model, the layers are even more difficult to imagine.

Convolutional neural network weight sharing can greatly reduce the computational effort. The case is shown in Figure 3, where the convolutional kernels have values of $-1, 0$ and $1$. From bottom to top, the bottom half of the step is $1$, the connecting line that represents the weights, and red, blue, and black represent values $-1, 0$, and $1$. From left to right with the scan of step $1$, the convolution will form the second level of output. The first half of this step is $2$. Each move will be two units, rotated sequentially from left to right, and the corresponding $-3, 3$, 0-second level output will be obtained. The constant throughout the process is the value in the convolution kernel. This is the weight of the component.

Based on an image map of the same size as $m^n$, one can resort to the weight sharing property, if each pixel value in the output characteristic map connects only a square block $k^*k$ in the size of the input image, then the output results in an input image block subject only to the $k^*k$ pixel size. The output layer parameters are then reduced from $m^n$ to $k^*k$. For each box of size $k^*k$ in the original graph, the corresponding output value needs to be calculated, so a total of $m^*n k^*k$ is required, assuming the input image is of the same order of magnitude as 102, and if $k$ is less than 10, the 10-106 parameter is completed. Relatively speaking, weight sharing allows for a reduction in the order of magnitude with good impact and effect.

After the image has gone through the convolutional layer, the size of the output feature map tends to be smaller than the input image. After going through successive convolutional layers, the feature map keeps shrinking, and the requirement that the input and output have the same size cannot be satisfied for tasks such as image filtering and scene segmentation. The process of enlarging the size of the feature map to realize the change from low resolution to high resolution is generally called up-sampling, while the process of changing the image resolution from high resolution to low resolution is called down sampling. The feature map is enlarged by inverse convolution, making it a common method for up-sampling in deep learning. The encoder-decoder network structure is a common network structure model currently used in handling tasks such as image segmentation, sequence learning, and machine translation. The encoder analyzes and extracts the feature information of the input image, and the decoder fully utilizes the inverse convolution property to decode the high-dimensional feature map of the image extracted by the encoder to improve the resolution and size of the feature map, to complete tasks such as specified image segmentation.

After convolution in the encoder, the shape of the feature map increases as the network layers deepen and the height and

![FIGURE 3. Flowchart of the deep label reconstruction algorithm.](image-url)
width shrink; after deconvolution in the decoder, the height and width of the feature map increase and the number of channels shrink; an intermediate processing module is not necessary, but other network structures can be added or not according to the requirements of the task. An image of the same scale as the original input can be obtained through the network structure of the encoder-decoder structure.

C. IMAGE DATA SET AND EVALUATION METRICS
ANALYSIS DESIGN
If the convolutional deep learning network has 17 layers and there are many parameters. Then these parameters will have a large impact on many models. Many studies have shown that being able to tune the deep learning parameters to optimize the model has better performance, sometimes even better than modifying the Bunsen algorithm. For a deeper study of the relevant parameter settings, in this unit, the focus is on noise reduction according to the algorithm. For a neural network, it is possible to know the important properties of the weighted parameters, which is the bridge to the whole model, which can change the weights and optimize the whole network model. When we start training for the first time, then how should we initialize it at this point, it can evolve directly to zero, the answer is wrong. Therefore, if the initialization value is correct, it will save the training time, as shown in Figure 4.

The structure of the system, shown in Figure 4, is a neural network based on a scale recursive structure, which is divided into three scales for recursion, and each scale consists of E-block and D-Block to form an encoder-decoder structure, where E-block is the encoder and D-Block is the decoder, and a recurrent neural network unit is added as an intermediate processing module based on the encoder-decoder network structure. In the forward propagation phase of the system, a noisy input image of size 256*256*1 is referred to as the original image, wherein the interpolation method used is a bilinear interpolation, and the interpolated image of size 1/4 of the original image is referred to as the first scale image, the interpolated image of size 1/2 of the original image is referred to as the second scale image, and the interpolated image of size 1/1 of the original image is referred to as the third scale image. Thus, the size of the first scale image is 64*64*1, the size of the second scale image is 128*128*1, and the size of the third scale image is 256*256*1.

The first scale image is overlapped in the last dimension to obtain a network input of 64*64*2, and the system outputs a feature map of 64*64*1, called the first scale feature map. The first scale feature map is interpolated to twice the original size and overlapped with the second scale image in the last dimension to obtain a 128*28*2 network input, and the system outputs a 128*128*1 feature map, referred to as the second scale feature map. The second-scale feature map is interpolated to twice the original size and overlapped with the third-scale image in the last dimension to obtain a network input of 256*256*2 and a system output of 256*256*1, referred to as the third-scale feature map. The resulting third-
The scale feature map is the demised image of the interferometric phase map corresponding to the real and imaginary parts. The structure of the system in each scale is shown in Figure 5.

The problem of noise reduction for image block-based models is to keep the input and output sizes consistent. However, when extracting features from convolutional kernels, for pixels in the region near the center of the block, although a lot of feature extraction is applied, the convolutional kernels also do a good job of extracting feature information from the edges. This reduces the amount of information used, which equates to a more accurate approximation to the middle, and the closer to the edge, the greater the error. The size of the noisy image block is very important and can determine the algorithm result as well as the algorithm performance. For example, the 49 * 49, MPL used in BM3D uses 47 * 47. In this study, the size of the target image block is controlled at 45 * 45 with a block number of m_2000, where the table is the size in batch normalization. To remove some of the effects, the noise was entered in the MLP and symmetric padding was performed. The same method as in the text is used in this paper. Before the convolution, find a simple zero-fill strategy without boundary effects with simple and effective properties. This time, by adjusting and experimenting with the m parameter in batch standardization, and carrying out experiments on several cases of 32, 64, 128, it is concluded that different batch sizes have different effects on the corresponding results, where the best processing can be achieved when m is 64.

In each scale of the system, the parameters in E-block, D-block, and recurrent neural network units are shared, where E-block consists of a series of convolutional layers, residual blocks, and residual blocks are composed of two layers of convolutional layers with jump joints. According to the change in the shape of the feature map in E-block, E-block can be divided into five modules, after each module, the height and width of the feature map in E-block become half of the original, the number of channels becomes twice the original, in the five modules, the number of feature map channels are 8, 16, 32, 64 and 128. Recurrent neural network unit optional LSTM, gated recurrent unit for comparison experiments. D-block is composed of an anti-curl layer and a convolutional layer, according to the shape change in D-block. D-block can also be divided into five modules, after each module, the height and width of the feature map in D-block become twice the original, the number of channels becomes half the original, through the anti-curl layer of D-block will E-block in the convolutional layer extracted feature map information to recombination. The simultaneous feature fusion of feature map information of multiple depths through the hopping structure not only facilitates the gradient propagation during backpropagation but also speeds up the convergence of the loss function.

A convolution-based implementation of the recurrent neural network unit is added to the encoder and decoder network structure as an intermediate module, and the key information of the filter-related noise distribution model is preserved in the neuron state in the recurrent neural network. The advantage of recurrent neural network units based on the full-connection layer is that they process sequence data and, like fully-connected neural networks, cannot extract local features of the image when processing data types in the form of a grid similar to feature maps or images, ignoring the spatial correlation of the image, containing excessive data redundancy, which not only can learn long-term dependencies of temporal data by recurrent neural networks but also combines the
advantages of convolutional neural networks for extracting spatial characteristics of images.

III. ANALYSIS OF RESULTS
A. SIMULATION RESULTS AND ANALYSIS
Based on semi-supervised learning, we abandon the traditional gradient descent algorithm and use the Adam algorithm to perform the replacement computation, choosing the multi-feature extraction technique to extract image feature information at the first layer, and then using the improved existing function to complete the activation of the relevant functions afterward. The salient features of the whole algorithm in terms of training and its convergence speed are further analyzed. It can be found that the adoption of the algorithm model has a great advantage in terms of convergence speed and velocity, showing superiority over other algorithms, as shown in Figure 6.

As shown in Figure 6, 14,000 iterations are performed for both the high and low noise criteria. The base of X is 10 logarithms, where the Y-axis will represent the value of the loss function that exists. At the curve level, the low noise criterion blue line corresponds to a final degree of convergence loss that is less than the high noise criterion blue line. This is a good fit with the standard rule because the higher the noise, the more disturbing the corresponding pixel will be and the more difficult it is to optimize the model. Moreover, when the noise is 35, we find that it tends to converge after 102.5, 316 iterations. At high noise of 75, the other curve tends to converge, and it converges almost 1000 times. The second time there was only a slight decrease. The corresponding loss function curves converge and remain stable through more than 4,000 comparisons and queries on other convolutional neural networks. This time, the optimized DNNA is used as the basis to construct the corresponding convolutional neural networks model, which has a better overall training speed.

The training set of the gated recurrent unit consists of simulation data, which is generated by (1) generating an initialization matrix of specified size by Gaussian random matrix; (2) scaling up the initialization matrix to 256*256 by double-triple interpolation and scaling up the matrix according to the maximum true phase; (3) keeping the noiseless true phase as a label and adding The complexity of the generated winding phase diagram can be controlled by adjusting the size of the random initialization matrix as shown in Figure 7, and the more complex the winding phase diagram is, the denser the interference pattern is and the more difficult it is to filter the noise.

By adjusting the size of the random initialization matrix, the complexity of the winding phase diagram can be controlled, and the more complex the winding phase diagram is the denser the interference pattern is the more affected by the noise, and the more difficult it is to filter. The random initialization matrix is initialized to 3*3 as shown in Figure 7. By expanding the initialization matrix to 7*7, the more complex winding phase diagram is generated.

20,000 simulated band-noise winding phase maps of size 256*256 are generated by simulation as a training set, with 50% of each of the real and imaginary images. The true phase signal-to-noise ratio after adding Gaussian noise is 21dB and the average number of residual differences is 3520. The size of the input image during training is 256*256, the initial learning rate is 1e-4, the size of each batch is 10 and the optimization method used is Adam. The gated recurrent unit using different recurrent neural network units was evaluated and the results are shown in Figure 8.

The recurrent neural network has the highest SSIM and the lowest MSE, which shows that the recurrent neural network can filter out the noise to the maximum extent while maintaining the structural characteristics between pixels. Therefore, the filtering quality of gated recurrent units using the gated recurrent unit is the best among the two types of gated recurrent units, and the following experiments will mainly compare the traditional method with the recurrent neural network. The SPD of Goldstein filter is similar to that of the recurrent neural network, but the MSE is the largest, which indicates that the Goldstein filter is too strong, resulting in a large correlation between the residual phase and the original noise phase, and thus losing the detail of the image during the filtering process. The recurrent neural network has the same noise rejection capability, but the phase detail retention is significantly better than the Goldstein filter, and the recurrent neural network algorithm runs much faster than the Goldstein filter. We can see from Figure 8 that the image error after recurrent neural network filtering is smaller than the two traditional interferometric filtering methods.

Figure 9 shows the SNR variation of the three filtering methods according to different evaluation indexes, and we can see that recurrent neural network is better than the Goldstein filter except for SPD, and recurrent neural network is better than Lee filter except for MSE.

By analyzing Figure 9, the performance of the gated recurrent unit network proposed in this paper is comparable to or better than other methods within a certain range of SNRs, despite the use of fixed SNRs for the training samples, which...
indicates that the proposed network has some generalization capability over a certain range of SNRs. For the best performance of the gated recurrent unit in real scenarios, the training sample and the measured direct should have similar feature distributions. Denoising results are given for Countryside (noise intensity of 20), House (noise intensity of 40), Barbara (noise intensity of 60), and Monarch (noise intensity of 80) algorithms. For the House image, the algorithm in this chapter does a better job in the sky part, with less noise interference than K-SVD, and shows its good denoising ability in detail. For the Monarch image, the algorithm is more prominent, comparing the details in the image with the outlines of the flowers. The dictionary update and atom-optimized image denoising algorithm in this chapter have outstanding detail at the presentation level and low noise interference, while the other three comparison algorithms are too smooth in image detail. We can see that WSDL is overall better than these three comparison algorithms, and the effect of prod is somewhat better than K-SVD, but not obvious. The algorithm in this chapter guarantees the image details to a certain extent, which is mainly due to the sparser representation dictionary in the sequential dictionary update part and the adaptive exclusion of the noisy dictionary atoms that interfere with the image representation in the dictionary atom optimization part, and thus the better denoising results can be obtained.

B. ANALYSIS OF IMAGE DENOISING RESULTS
In the bottom right corner of Figure 10, we extracted the eye pixels of the parrot in the green box and plotted them as blocks with the help of MATLAB. This allows us to
FIGURE 10. Denoising comparison under different noise standard deviations.

The first step in Figure 10 compares the image clarity of the algorithm after optimization with the differences between the different noise markers. We tested the differences for noise standard deviations $\sigma = 15, 25, 35, 50, 65, 75$. In this case, parrot images from the Set12 standard test set are analyzed. Although in this study, the processing results of the corresponding algorithm are plotted based on the noise environment $\sigma = 50$ only, however, this does not mean that the optimized convolutional neural networks noise reduction algorithm performs worse than the other optimized algorithms in other noise environments, in fact, Figure 10 above shows that the noise reduction effect of this test image is consistent in other noisy environments. For example, the optimized noise reduction algorithm is not as good as the noise on the two images of House and Barbara. The wavelet neural network algorithm is good and shows that wavelet neural networks are superior in processing these images. Under other noise conditions, these images would be better processed. Whereas the improved denoising algorithm has the best overall performance on all 10 test images, it is usually consistent over a range of noise under different noise criteria.

Also, the denoising model itself has better data performance. For this reason, a subjective evaluation method can be applied to the denoised images. We observe the demised image from a subjective point of view. The seven images are placed in the order of their peak SNR from low to high. The worst performer among the denoising algorithms is BM3D, while the improved convolutional denoising network has the best denoising effect, with an overall peak SNR of 30.00dB, which is 0.65dB higher than that of convolutional neural networks. One is because the image is rich in detail in the eye, and the other is that convolutional neural networks were also used so that it could be compared with the pairing algorithm. In terms of vision, the parrot’s eye is clear from the front to the back row. When MLP was adopted as the algorithm, the eye position of the image was still more blurred. The overall effect of its performance shows that the algorithm used in the study is more advantageous than convolutional neural networks. In comparing the effect of the algorithm in terms of specific detailed features. It can be found that the convolutional neural networks as well as the improved algorithm are the most reasonable choice.

Although it can be concluded from the above analysis that the algorithm has good denoising performance under the differential noise criterion. However, it is not convincing to look at the de-noising performance of the model itself. Therefore, in this study, the statistical results under the differential noise criterion are constructed for the above two test sets so that the different algorithm models can be easily compared with other relevant algorithm models to enable objectivity. The test results of the dataset are presented in Figure 11.

To comprehensively compare the convolutional neural network models with better performance in the field of noise reduction, the existing data in the literature are applied in this study. In this paper, we show how the noise standard deviation at $\sigma = 15, \sigma = 25, \sigma = 50$ compares with other excellent denoising algorithms. To make the comparison clearer, we use it in all algorithms. In this noisy background, the bold
characters show that the algorithm model is at the maximum PSNR value of the detection graph among the listed models, and thus Figure 12 can be obtained.

It can be seen from Figure 12 that the optimized noise reduction algorithm, for this Set12 training set, is better than the other three algorithms. Comparing BM3D with WNNM has a big advantage on House and Barbara, which is about 1db higher than the improved algorithm, but from some other tests, the algorithm adopted in this study performs better. The algorithm is worse, and except for the better PSNR on the House chart, BM3D performs significantly different than the other test chart algorithms. As shown in Figure 13, you can see the good performance demonstrated in each test using the improved algorithm. The performance of the image denoising algorithm on Set12 can be seen in Figure 13 when taking $\sigma = 50$. By averaging the algorithms on the 12 test sets, the improved denoising algorithm is effective in improving the peak signal-to-noise ratio.

Compared to set12 in the standard data set, the analysis of the corresponding noise reduction results shown significant improvement over convolutional neural networks. To improve the image learning algorithm and further enhance the corresponding learning capabilities, significantly reduce the relevant training model time, etc., the first layer multi-feature extraction technique was selected in this study, while three convolutional kernels of different sizes were used according to different locations. The features are extracted constantly and synchronously on the input image block. This method can extract rich image features, which enables the convolutional neural networks model to better learn the noisy image blocks and thus significantly remove the corresponding noise effects. To make the fuzzy noise image clear and image details obvious, image denoising is achieved by using a neural network, to achieve more stable and good image display capability. Based on the convolutional neural network (convolutional neural networks) algorithm, the role of the network is studied by activation function optimization. Multi-feature extraction techniques and deep learning are fused so that they can be used to learn the rich features of the input image as well as better leverage the adaptive algorithm. The algorithm can optimize the backward propagation of the convolutional neural networks and speed up the training of the model, which can significantly improve the convergence of the algorithm. Deep residual learning based on the convolutional network is used to denoise the image. A better network model for image denoising and noise reduction. Finally, a comparison with other excellent denoising algorithms is made. From the perspective of the comparison results, the optimized denoising algorithm can support the improvement of noise pollution and clearer image details without compromising image clarity. At the same time, the algorithm can exhibit good peak signal-to-noise ratio under a variety of noise standard deviations compared to other image denoising algorithms. In this paper, it is verified that the convolutional neural network noise reduction model, which is improved by the multi-feature extraction technique, has strong advantages for image denoising.

**IV. CONCLUSION**

This paper mainly studies a denoising algorithm based on improved semi-supervised learning. The algorithm first maps the semi-supervised learning tensor to the Riemann component, then uses the sparse Bayesian learning method to sparsely represent it, and finally obtains the denoised image by reconstructing the image. In the experiment, the algorithm is combined with several existing denoising algorithms to denoise simulated images and semi-supervised learning images. From the results of qualitative and quantitative analysis, the algorithm can preserve the non-uniformity of tensors. Under the premise of linear structure, the semi-supervised learning image is denoised. In this paper, a semi-supervised learning denoising algorithm based on Riemann’s non-local similarity is proposed based on the common characteristic of non-local self-similarity of natural images and the application of non-local similarity theory to semi-supervised learning denoising in combination with Riemann geometric theory.
The algorithm first maps the semi-supervised learning tensor onto Riemann manifolds and searches for nonlocal similarity blocks to form similar block groups using a Riemann similarity measure, then uses a Gaussian mixture model to learn the a priori distribution of the block groups, and finally demises the block groups by Bayesian inference, reorganizing the demised block groups to obtain the final demised images. Experiments on denoising simulated and semi-supervised learning images show that the algorithm not only removes the noise in semi-supervised learning but also better preserves the edge texture information of the images. By comparing the denoising experiments of the two algorithms proposed in this paper, some theoretical limitations of the two algorithms are found, which need to be further investigated in the future. Then the image block characteristic structural properties of the trained dictionary atoms are used, and the effects of the structural complexity and noise intensity of the original image on the dictionary atoms are fully considered to adaptively perform atom detection and remove the noisy atoms; finally, the optimized dictionary is used to reconstruct the image. The experimental results show that the algorithm can achieve a better denoising effect compared with the classical denoising algorithm.

REFERENCES

[1] H. Samih, “Smart cities and Internet of Things,” J. Inf. Technol. Case Appl. Res., vol. 21, no. 1, pp. 5–12, Jan. 2019.
[2] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, “A semi-supervised convolutional neural network for hyperspectral image classification,” Remote Sens. Lett., vol. 8, no. 9, pp. 839–848, Sep. 2017.
[3] H. Wu and S. Prasad, “Semi-supervised deep learning using pseudo labels for hyperspectral image classification,” IEEE Trans. Image Process., vol. 27, no. 3, pp. 1259–1270, Mar. 2018.
[4] Z.-H. Zhou, “A brief introduction to weakly supervised learning,” Nat. Sci. Rev., vol. 5, no. 1, pp. 53–53, Jan. 2018.
[5] C. Sakaridis, D. Dai, and L. Van Gool, “Semantic foggy scene understanding with synthetic data,” Int. J. Comput. Vis., vol. 126, no. 9, pp. 973–992, Sep. 2018.
[6] T. Kong and K. Lee, “Semi-supervised Bayesian adaptive multiscale shrinkage for wavelet-based denoising,” Commun. Statist. Simul. Comput., vol. 46, no. 6, pp. 4399–4418, Jul. 2017.
[7] P. Xu, W. Lu, and B. Wang, “A semi-supervised learning framework for gas chimney detection based on sparse autoencoder and TSVM,” J. Geophys. Eng., vol. 16, no. 1, pp. 52–61, Feb. 2019.
[8] Z. He, F. Nan, X. Li, S.-J. Lee, and Y. Yang, “Traffic sign recognition by combining global and local features based on semi-supervised classification,” IET Intell. Transp. Syst., vol. 14, no. 5, pp. 323–330, May 2020.
[9] Z. Zhang, D. Zhang, X. Yin, and Y. Liu, “A novel semi-supervised learning approach in artificial olfaction for E-Nose application,” IEEE Sensors J., vol. 16, no. 12, pp. 4895–4911, Jun. 2016.
[10] Z. Wang, L. Qu, J. Xin, H. Yang, and X. Gao, “A unified distributed ELM framework with supervised, semi-supervised and unsupervised big data learning,” Memetic Comput., vol. 11, no. 3, pp. 305–315, Sep. 2019.
[11] A. Gogna, A. Majumdar, and R. Ward, “Semi-supervised stacked label consistent autoencoder for reconstruction and analysis of biomedical signals,” IEEE Trans. Biomed. Eng., vol. 64, no. 9, pp. 2196–2205, Sep. 2017.
[12] E. Markurjev, A. L. Bertozzi, and F. Chung, “A semi-supervised heat kernel pagerank MBO algorithm for data classification,” Commun. Math. Sci., vol. 16, no. 5, pp. 1241–1265, 2018.
[13] Z. Shi, S. Osher, and W. Zhu, “Weighted nonlocal Laplacian on interpolation from sparse data,” J. Sci. Comput., vol. 73, nos. 2–3, pp. 1164–1177, Dec. 2017.
[14] S. Ren, S. Wan, X. Gu, P. Yuan, and H. Xu, “Semi-supervised hyperspectral image classification using local low-rank representation,” Remote Sens. Lett., vol. 10, no. 2, pp. 195–204, Feb. 2019.
[15] D. A. Alaswad and Y. F. Hassan, “Using semi-supervised learning in cellular automata for edge detection of image segmentation,” Int. J. Eng. Comput. Sci., vol. 7, no. 02, pp. 23613–23619, Feb. 2018.
[16] J.-L. Yin, B.-H. Chen, and Y. Li, “Highly accurate image reconstruction for multimodal noise suppression using semi-supervised learning on big data,” IEEE Trans. Multimed., vol. 20, no. 11, pp. 3045–3056, Nov. 2018.
[17] C. You, W. Cong, M. W. Vanniekerk, P. K. Saha, E. A. Hoffman, G. Wang, G. Li, Y. Zhang, X. Zhang, H. Shan, M. Li, S. Ju, Z. Zhao, and Z. Zhang, “CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE),” IEEE Trans. Med. Imag., vol. 39, no. 1, pp. 188–203, Jan. 2020.
[18] Y. Zhang, J. Wen, Z. Liu, and C. Zha, “A semi-supervised approach of graph-based with local and global consistency,” Int. J. Inf. Technol. Manage., vol. 18, nos. 2–3, pp. 243–255, Jun. 2019.
[19] N. Kumar, P. Uppala, K. Duddu, H. Sreedhar, V. Varma, G. Guzman, M. Walsh, and A. Sethi, “Hyperspectral tissue image segmentation using semi-supervised NMF and hierarchical clustering,” IEEE Trans. Med. Imag., vol. 38, no. 5, pp. 1304–1313, May 2019.

K. Zhang, K. Chen: Semi-Supervised Learning-Based Image Denoising for Big Data
KUN ZHANG was born in Shandong, China, in 1981. She received the bachelor’s and master’s degrees from the Shandong University of Science and Technology, in 2003 and 2008, respectively. From 2003 to 2005, she was with the Shandong University of Science and Technology, where she currently works. She has published one article, which has been indexed by SCI. Her research interests include computer software and digital image processing.

KAI CHEN was born in Shandong, China, in 1982. He received the bachelor’s and master’s degrees from the Shandong University of Science and Technology, in 2005 and 2010, respectively. He is currently with the Taishan Institute of Technology, Shandong University of Science and Technology. His research interests include big data and computer networks.

* * *