IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 November 2022, accepted 30 November 2022, date of publication 2 December 2022,
date of current version 13 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226501

RESEARCH ARTICLE

Double-Norm Constrained Image Denoising Algorithm Based on Dictionary Learning Sparsity and FCM Structure Clustering

CHANGPENG JI[1], LINA HE[2], AND WEI DAI[1]
[1]School of Electronics and Information Engineering, Liaoning Technical University, Huludao 125105, China
[2]Graduate School, Liaoning Technical University, Huludao 125105, China

Corresponding author: Changpeng Ji (ccp@lntu.edu.cn)

ABSTRACT To solve the problem of image smoothness and fuzzy edge texture information after image denoising, proposed a new image denoising method based on dictionary learning. Firstly, the external cycling principal component analysis reduces the dimensions of image data while retaining the main data and constructing the learning dictionary. Secondly, used the fuzzy c-means structure clustering method internally to implement structural constraints on learning dictionary training, which considered the internal structure of image pixels. Then, the learning dictionary under the double constraints of sparse and structural clustering is obtained by internal and external iteration. Finally, the sparse representation coefficient and redundancy dictionary are obtained by the orthogonal matching pursuit method and alternate direction multiplier method, and the denoised images are estimated and updated according to sparse coding theory. Using the grayscale image from Set12 data set, color image from the CBSD68 data set, real noise from RENOIR data set, and texture image from USC-SIPI data set. The experimental results show that compared with the model-based algorithms (KSVD, ISKR, EPLL, NCSR, and LR-GSC) and the learning-based algorithms(DnCNN, IRCNN, and FFDNet), the proposed algorithm preserves the edge and texture information of the image better, and achieves better subjective visual effects and objective numerical results, especially for the image with complex structure and content, and the running time is much less than learning-based algorithms.

INDEX TERMS Dictionary learning, FCM clustering, image denoising, structural clustering, sparse representation.

I. INTRODUCTION

Images are an important way for humans to acquire and transmit information. In the real world, the quality of acquired images deteriorates due to the imperfection of image acquisition sensors, transmission media, and recording equipment [1]. In the noisy image, the noise will cover the critical information [2] and affect people's access to the vital information in the image. Image noise reduction model can be modeled as:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n} \tag{1}$$

$\boldsymbol{x}$ is the original image, $\boldsymbol{y}$ is the image with noise, and $\boldsymbol{n}$ is the noise matrix. Image noise reduction process is to obtain $\boldsymbol{x}$ through $\boldsymbol{y}$. Image denoising is the premise of other image signal processing. Improving the image denoising results is crucial for the subsequent use of the image. Image denoising algorithms can be generally divided into model-based methods [3], [4], [5], [6], [7], [8], [9], [10] and learning-based methods [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

The denoising method based on model mainly adopts the optimization strategy based on well-defined image prior or noise statistics. Elad proposed a dictionary learning algorithm based on a fixed dictionary with k times singular value decomposition (KSVD) and successively proposed a global based adaptive learning dictionary to solve the problem of ignoring image pixel information [5], [6], [7]. Zoran used maximizing the expected patch log likelihood (EPLL) of the reconstructed image, subject to constraining reconstructed

The associate editor coordinating the review of this manuscript and approving it for publication was Varuna De Silva.

image still close to the corrupted image [8]. Takeda extended the iterative steering kernel regression(ISKR) idea and proposed an ISKR framework for image denoising [9]. Zha proposed a low-rank regularized group sparse coding (LR-GSC) model [10], which utilized both group sparsity of the canonical coefficient and the low rank of the dictionary. The drawbacks of these algorithms are that they either need to spend a lot of time searching for similar blocks or they can't retain details such as the edge and texture of the image.

The denoising method based on deep learning mainly uses an effective network structure and data set for training to obtain the reconstructed image signal. Chen proposed a trainable nonlinear reaction-diffusion (TNRD) model [12]. Zhang used residual learning and batch normalization to accelerate the training process and proposed a feed-forward denoising convolutional neural networks (DnCNN) [14] to improve the denoising performance and trained a image restoration convolutional neural network(IRCNN) [15] to solve other inverse problems, and successively propose two algorithms to generalize the network. One is the fast and flexible denoising network (FFDNet) algorithm, a denoising network with adjustable noise level mapping as input [16]; the other is a convolutional blind denoising network (CBDNet), a convolutional blind denoising network with asymmetric learning to perform real image blind denoising [17]. At present, the image denoising algorithm based on deep learning has achieved the best effect, but it is difficult to explain the principle behind it, and the drawbacks of these methods are that the learning and inference time is extremely long.

In order to solve the problem of too long denoising time and loss of image edge, texture, and other details after denoising. Proposed an image denoising algorithm based on fuzzy c-means (FCM) clustering structure constraints and sparse dictionary learning (FCM-SDL). FCM-SDL algorithm is a compound denoising model which combines the FCM clustering algorithm and sparse representation. FCM clustering algorithm was used to make use of the structure of the image. Then alternate direction multiplier method (ADMM) [20] and orthogonal matching pursuit (OMP) [21] method were applied to solve the optimization. The fuzzy c-means clustering algorithm [22], [23] avoids the disadvantages of hard clustering and enhances the denoised image to retain more detailed information. Through the qualitative and quantitative comparison with similar excellent denoising algorithms, the effectiveness of the proposed algorithm is verified.

Khmag proposed two image denoising algorithms [24], [25], which used second-generation wavelet transform and principal component analysis (PCA). The influence of the second-generation wavelet transform on image denoising was that it eliminated the noise through one-generation wavelet transform, and second-generation wavelet improves the weak Gibbs effect. But our algorithm used the inner loop to make use of the internal structure information of the image through the clustering algorithm. On this basis, the outer circle utilized the sparsity of dictionary learning through the clustering constraint structure. PCA algorithm in our algorithm is mainly used to reduce the dimension of data during the construction of dictionary learning and speed up the calculation time. Dong proposed the non-local centralized sparse representation (NCSR) and used the combination of PCA and kmeans for image restoration [26]. Kmeans clustering constraints were not as fine as FCM clustering constraints, and FCM maintained finer image structure information at the cost of losing less time. We also adjusted the size of the image segmentation block and iteration threshold better in the experimental process according to many current studies.

The contributions of this paper are summarized as follows:

Firstly, we created a novel image-denoising method based on the picture's structure and the dictionary's sparsity.

Secondly, we used the FCM clustering algorithm to combine the constraint of the internal structure of the image with the sparse theory to obtain better the overall structure and detailed information of the denoised image.

Thirdly, we suggested adaptive modification parameters based on the category and noise level, primarily altering the size of the segmented picture block.

The diagram of the proposed algorithm is as Fig.1 shows:

The rest of the paper is organized as follows. Section II briefly introduces related works in image denoising and image clustering. Section III provides an overview of the FCM-SDL algorithm. Section IV shows the numerical results, and compares the proposed method with other image-denoising methods. Finally, Section V concludes the paper.

II. SPARSE REPRESENTATION AND STRUCTURAL CLUSTERING MODELS

A. SPARSE REPRESENTATION

In the sparse representation $\boldsymbol{D} \in \boldsymbol{R}^{(n\times k)}(n < k)$, the image $\boldsymbol{x} \in \boldsymbol{R}^{(n)}$ can be represented mathematically as $\boldsymbol{x} = \boldsymbol{D\alpha}$; $\boldsymbol{D}$ is a redundant dictionary, $\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_k$ is the column vector of the dictionary, k is the number of dictionary columns, and n is the number of dictionary rows. $\boldsymbol{x}$ is the original image and $\boldsymbol{\alpha}$ is the sparse representation coefficient.

Set a threshold to indicate that the accuracy of sparse representation approximately meets the requirements, that is

$$\|\boldsymbol{D\alpha} - \boldsymbol{x}\|_2 \leq \varepsilon \tag{2}$$

ε is the threshold value, followed by the requirement of sparsity L, that is

$$\left\|\hat{\boldsymbol{\alpha}}\right\|_0 \leq L \leq n \tag{3}$$

Eq.(3) ensures that $\boldsymbol{\alpha}$ is sparse enough. Parameter $(\varepsilon, L, \boldsymbol{D})$ defines a mathematical model as described above.

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad s.t. \quad \|\boldsymbol{D\alpha} - \boldsymbol{x}\| \leq T \tag{4}$$

$\hat{\boldsymbol{\alpha}}$ is the estimator of $\boldsymbol{\alpha}$. Sparse means that the sparse representation coefficient $\boldsymbol{\alpha}$ contains more zeros to ensure the sparse representation coefficient $\boldsymbol{\alpha}$ is sparse enough; sparse representation means using fewer column vectors of learning dictionary $\boldsymbol{D}$ when representing the reconstructed image by

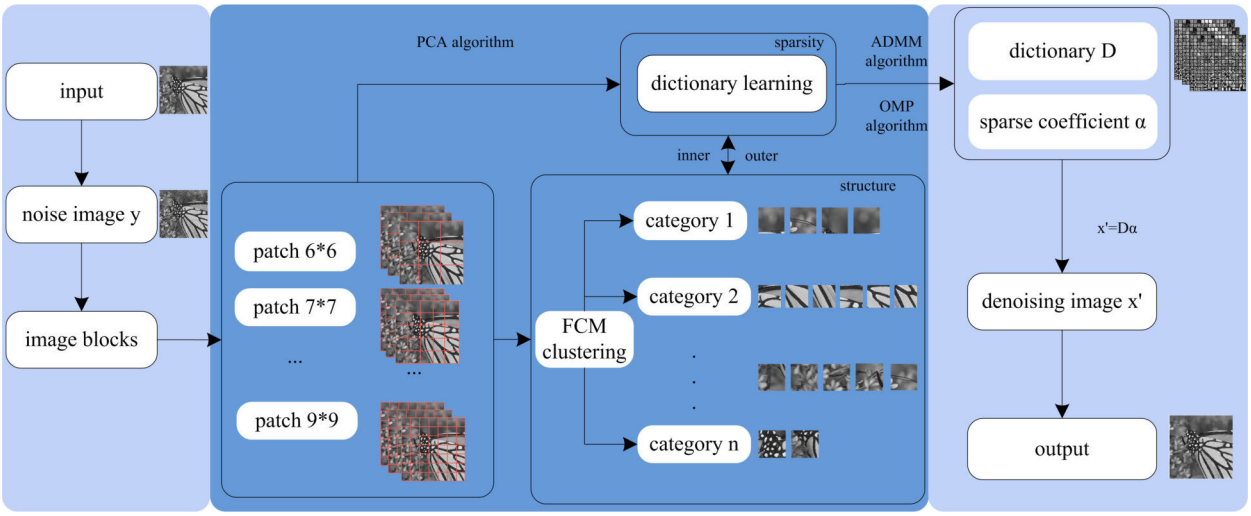


FIGURE 1. Diagram of the fuzzy c-means clustering structure constraints and sparse dictionary learning algorithm.

the learning dictionary $\boldsymbol{D}$ and sparse representation coefficients $\boldsymbol{\alpha}$; T is the iterative threshold [27], [28], which is determined by ε and σ and can solve T by the maximum a posteriori (MAP) algorithm [29].

Then Eq.(4) can be optimized as:

$$\hat{\boldsymbol{\alpha}} = \underset{\alpha}{argmin} \|\boldsymbol{D\alpha} - \boldsymbol{y}\|_2^2 + \mu\|\boldsymbol{\alpha}\|_0 \tag{5}$$

μ is the penalty factor, and Eq.(4) and Eq.(5) are equivalent when μ is given a suitable one.

If we divide the unknown image $\boldsymbol{x}$ into overlapping small blocks for processing (prevent artifacts) and represent each small block by the constructed sparse model $(\varepsilon, L, \boldsymbol{D})$, then the MAP algorithm is used to estimate as:

$$\left\{\hat{\boldsymbol{D}}, \hat{\boldsymbol{\alpha}}_{ij}, \hat{\boldsymbol{x}}\right\} = \underset{D,\alpha_{ij},x}{argmin} \lambda \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum_{i,j} \mu_{ij} \left\|\boldsymbol{\alpha}_{ij}\right\|_0 + \sum_{i,j} \left\|\boldsymbol{D\alpha}_{ij} - \boldsymbol{R}_{ij}x\right\|_2^2 \tag{6}$$

The first part of Eq.(6) represents the global log-likelihood, which guarantees the closeness between $\boldsymbol{x}$ and $\boldsymbol{y}$, λ is the fidelity. The second part represents the reconstructed image still has sparsity under the condition of limited errors, μ_{ij} is the penalty factor of each item, which controls the weight. The third part indicates that the sub-image reconstructed by the dictionary is sufficiently similar to the original image, the sliding matrix window $\boldsymbol{R}_{ij}$ of size $n \times N$, which controls the size of the extracted image block, and $\boldsymbol{R}_{ij}\boldsymbol{x}$ is the sub-image of row i and column j, and $\boldsymbol{D\alpha}_{ij}$ is the approximate sub-image obtained by reconstruction.

The solution of $\hat{\boldsymbol{\alpha}}$ is a convex optimization problem, which the orthogonal matching pursuit algorithm can solve. The solution of $\hat{\boldsymbol{D}}$ is a problem of least absolute shrinkage and selection operator, use the ADMM algorithm to solve the problem. Initialize $\hat{\boldsymbol{D}}$ to an over-redundant discrete cosine transformation (DCT) dictionary, $\boldsymbol{x}$ is initialized to $\boldsymbol{y}$, and the PCA algorithm is used to update the dictionary [30], [31]. Finish updating the dictionary until it reaches the iteration threshold and sparsity. Then, after getting $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{D}}$, update the denoised image $\hat{\boldsymbol{x}}$ as:

$$\hat{\boldsymbol{x}} = \underset{x}{argmin} \lambda \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum_{i,j} \left\|\boldsymbol{D\alpha}_{ij} - \boldsymbol{R}_{ij}\boldsymbol{x}\right\|_2^2 \tag{7}$$

Solve Eq.(7) by the least square method:

$$\hat{\boldsymbol{x}} = \left(\lambda \boldsymbol{I} + \sum_{i,j} \boldsymbol{R}_{ij}^T \boldsymbol{R}_{ij}\right)^{-1} \left(\lambda \boldsymbol{y} + \sum_{i,j} \boldsymbol{R}_{ij}^T \boldsymbol{D}_{ij}\right) \tag{8}$$

$\boldsymbol{I}$ is the identity matrix, averaging the denoised blocks, and some relaxation is obtained by averaging the original noise image. The inverse matrix in the above expression is diagonal, so the calculation of Eq.(7) can also be carried out pixel by pixel according to the sliding window described in the sparse coding step.

B. FCM STRUCTURAL CLUSTERING

FCM clustering is a fuzzy clustering algorithm based on an objective function. The objective function of FCM and its constraint conditions are as follows:

$$\boldsymbol{J} = \sum_{i,j} \sum_{i,j} \mu_{ij}^m \left\|\boldsymbol{x}_j - \boldsymbol{c}_j\right\|^2 \tag{9}$$

$$\sum_{i=1}^{n} \mu_{ij} = 1, j = 1, 2, \ldots, n \tag{10}$$

$\boldsymbol{J}$ is the objective function. The purpose is to divide all the data points $\boldsymbol{x}_i (i = 1, 2, \ldots, n)$ into c categories under the condition that the value of the constructed objective function reaches the minimum. $\boldsymbol{c}_j$ is the contribution to class j; μ_{ij} is the membership degree of each sample belonging to a certain category i and represents the probability of each sample belonging to each category. m is a membership factor in ensuring that the clustering results can not only ensure the degree of similarity between samples in the same category but

also ensure the heterogeneity between samples in different categories.

The FCM update method is:

$$\boldsymbol{c}_i = \sum_{j=1}^{n} \frac{\mu_{ij}^m}{\sum_{j=1}^{n} \mu_{ij}^m} \boldsymbol{x}_j \tag{11}$$

Firstly, sum the membership degrees of all points to this category. Then divide the membership of each point by the sum, which is the proportion of each point in the whole, and multiply by that point's contribution $\boldsymbol{x}_j$ to this category. According to the number of $\boldsymbol{c}_i$, determine the class i. The evaluation of the clustering effect of the FCM algorithm mainly depends on the fuzzy average entropy. The closer the fuzzy average entropy is to zero, the better the clustering effect.

III. FCM-SDL MODEL AND DENOISING PRINCIPLE

A. FCM-SDL MODEL ESTABLISHMENT EQUATIONS

Based on Eq.(6), the model in this paper combines the image structure constraint term constituted by the FCM clustering algorithm, takes into account the prior information of natural image sparsity, to form a new FCM-SDL model, which can be expressed as

$$\hat{\boldsymbol{\alpha}} = argmin \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \mu_1\|\boldsymbol{\alpha}\|_1 + \mu_2\left\|\boldsymbol{\alpha}_i - \boldsymbol{\gamma}_i\right\|_2 \tag{12}$$

μ_1 and μ_2 are non-negative parameters, and $\boldsymbol{\gamma}_i$ is the clustering center of FCM clustering. Compared with the model in Eq.(6), the first item of this model describes the similarity between the estimated image and the original image, the second item describes the weighted relative TV constraint, which enhances the sparsity of the learning dictionary, and the third item describes the image structure constraint item composed of clustering.

FCM clustering can increase classification accuracy without taking too long and avoid the drawbacks of hard clustering algorithms. Increased classification accuracy allows for a better excavation of the image's core structural data as noise levels rise. When the noise level is high, FCM can effectively limit the general structure of the picture to prevent the loss of features like texture and edge, which avoids the over-smoothing phenomena of the image after noise removal.

B. FCM-SDL MODEL SOLVING

Solving the FCM-SDL model can be regarded as solving the following objective function:

$$f(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \mu_1\|\boldsymbol{\alpha}\|_1 + \mu_2\|\boldsymbol{\alpha} - \boldsymbol{\gamma}\|_1 \tag{13}$$

To ensure that the FCM-SDL model is always a convex optimization problem in the process of iterative solution, a surrogate function is introduced to ensure the shrinkage of the iterative solution [32]. The basic idea behind the surrogate function is to show that the simple process of iterative shrinkage in the scalar case also applicable in more general case (i.e., D is not unitary). In [32], the author introduces the following auxiliary functions.The surrogate function is as follows:

$$\boldsymbol{\Psi}(\boldsymbol{\alpha}, \boldsymbol{\alpha}_0) = \frac{c}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|_2^2 - \frac{1}{2}\|\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{D}\boldsymbol{\alpha}_0\|_2^2 \tag{14}$$

c is chosen to ensure that $\boldsymbol{\Psi}(\boldsymbol{\alpha}, \boldsymbol{\alpha}_0)$ is convex, because $\boldsymbol{\alpha}_0 = \boldsymbol{D}^T\boldsymbol{x}$, and the unitary matrix property $\|\boldsymbol{x}\|_2^2 = \|\boldsymbol{D}\boldsymbol{x}\|_2^2$, then the objective function of Eq.(13) becomes:

$$\begin{aligned} f(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\alpha}_0) = {} & \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \mu_1\|\boldsymbol{\alpha}\|_1 + \mu_2\|\boldsymbol{\alpha} - \boldsymbol{\gamma}\|_1 \\ & + \frac{c}{2}\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\right\|_2^2 - \frac{1}{2}\|\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{D}\boldsymbol{\alpha}_0\|_2^2 \end{aligned} \tag{15}$$

After some mathematical operations (see Appendix B), Eq.(15) can be simplified to:

$$\begin{aligned} f(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\alpha}_0) = {} & const + \mu_1\|\boldsymbol{\alpha}\|_1 + \mu_2\|\boldsymbol{\alpha} - \boldsymbol{\gamma}\|_1 \\ & + \frac{c}{2}\|\boldsymbol{\alpha} - v_0\|_2^2 \end{aligned} \tag{16}$$

where $\boldsymbol{v}_0$ is equal to $\boldsymbol{v}_0 = \frac{1}{c}\boldsymbol{D}^T(\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}_0) + \boldsymbol{\alpha}_0$.

By converting the above minimization problem into its scalar form: $\boldsymbol{\alpha} \to t$, $\boldsymbol{v}_0 \to t_0$, and $\boldsymbol{\gamma} \to r$, the mathematical solution of the double-$\ell 1$ optimal solution can be achieved (see Appendix A).

$$g(t) = \frac{1}{2}(t - t_0)^2 + \tau_1|t| + \tau_2|t - r| \tag{17}$$

where $\tau_1 = \frac{\mu_1}{c}$ and $\tau_2 = \frac{\mu_2}{c}$ are the scaling relaxation parameters. Then the solution of Eq.(17) is as follows:

$$\boldsymbol{\alpha}_j^{(i+1)} = \begin{cases} \boldsymbol{S}_{\tau_1,\tau_2,\boldsymbol{\gamma}_j}(\boldsymbol{v}_j^{(i)}), \boldsymbol{\gamma}_j \geq 0 \\ -\boldsymbol{S}_{\tau_1,\tau_2,-\boldsymbol{\gamma}_j}(-\boldsymbol{v}_j^{(i)}), \boldsymbol{\gamma}_j < 0 \end{cases} \tag{18}$$

In Eq.(18), $\boldsymbol{v}^{(i)}$ is

$$\boldsymbol{v}^{(i)} = \frac{1}{c}\boldsymbol{D}^T(\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}^{(i)}) + \boldsymbol{\alpha}^{(i)} \tag{19}$$

The generalized shrink operator $\boldsymbol{S}_{\tau_1,\tau_2,r}(t)$ is defined as follows:

$$\boldsymbol{S}_{\tau_1,\tau_2,r}(t) = \begin{cases} t + \tau_1 + \tau_2 & t < -\tau_1 - \tau_2 \\ 0 & -\tau_1 - \tau_2 \leq t \leq \tau_1 - \tau_2 \\ t - \tau_1 + \tau_2 & \tau_1 - \tau_2 < t < \tau_1 - \tau_2 + b \\ b & \tau_1 - \tau_2 + b < t < \tau_1 + \tau_2 + b \\ t - \tau_1 - \tau_2 & t > \tau_1 + \tau_2 + b \end{cases} \tag{20}$$

C. FCM-SDL DENOISING PROCESS

In Eq.(17), τ_1, τ_2 refers to the situation where noise is not involved. When using τ_1, τ_2 when the image involves noise, the following strategies are adopted to adapt the two regularization parameters.

$$\tau_1 = c_1\frac{\sigma_w^2}{\sigma_\alpha}, \tau_2 = c_2\frac{\sigma_w^2}{\sigma_\gamma} \tag{21}$$

σ_w^2 is the noise variance, and c_1 and c_2 are auxiliary parameters to ensure the convexity of the surrogate function

(default is usually set to $c_1 < c_2$, emphasizing non-local terms).

$$\begin{cases} \hat{\boldsymbol{x}}^{(i+1)} = \tilde{\boldsymbol{S}}((1-\delta)\boldsymbol{x}^{(i)} + \delta \boldsymbol{y}) \\ (1-\delta)\hat{\boldsymbol{x}}^{(i)} + \delta \boldsymbol{y} = \boldsymbol{x}^{(i)} + \delta(\boldsymbol{y} - \boldsymbol{x}^{(i)}) \end{cases} \tag{22}$$

Eq.(22) is an operator that implements the idea of iterative regularization, where $\boldsymbol{S} = \tilde{\boldsymbol{D}} \circ \boldsymbol{S} \circ \boldsymbol{R}$ represents the projection onto the regularization constraint set. The right side can be regarded as a degraded *Landweber* operator (when the fuzzy kernel is reduced to an identity operator). δ is a small positive number that controls the amount of noise fed back to the iteration.

The steps of the algorithm FCM-SDL proposed are as Algorithm 1 shows:

Algorithm 1 Image-Denosing via FCM-SDL

1: **Input:** Noise image $\boldsymbol{y}$.
-initialization parameters: initialize the estimated image $\hat{\boldsymbol{x}}$: $\boldsymbol{x} \hat{=} \boldsymbol{y}$, and enter the denoising loop.
2: **Outer loop (dictionary learning):**loop i times, $i = 1, 2, \ldots, I$.
-Update Dictionary $\boldsymbol{D}$: via PCA algorithm.
3: **Inner loop (structural clustering):**loop j times, $j = 1, 2, \ldots, J$.
-iterative regularization:$\tilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} + \delta(\boldsymbol{y} - \hat{\boldsymbol{x}})$
-regularization coefficient update: estimating the new τ_1, τ_2 by Eq.(21).
-cluster center estimation update: new cluster centers are obtained $\boldsymbol{\gamma}_k$ by FCM algorithm estimation.
-image estimation update: through $\boldsymbol{x} = \boldsymbol{D} \circ \boldsymbol{S} \circ \boldsymbol{R}\tilde{\boldsymbol{x}}$ and get a new estimated image $\boldsymbol{x}$.
4: **Judgement:**
-if the iteration threshold T is reached, the iteration should be stopped; otherwise, it should be continued.
5: **Output:**restore image $\boldsymbol{x}$.
-using sparse coefficients and learning dictionary to reconstruct and restore the image.

IV. EXPERIMENT AND ANALYSIS

Experimental test environment: Matlab R2018b, the CPU is AMD Ryzen 5 4600H with Radeon Graphics 3.00GHz, the RAM is 16.0GB, NVIIDA GeForce GTX1650, and Python 3.10, PyTorch 1.12.1, and CUDA 11.3. The grayscale image from Set12 data set, color image from the CBSD68 data set, real noise from RENOIR data set [33], and texture image from USC-SIPI data set. The experiment was carried out under different types of noise with different intensities. The supervisor's visual effect and objective evaluation index obtained by different algorithms were compared through qualitative and quantitative analysis.

Setting experimental parameters: Adjust the size of the image block based on previous experience [34], [35], [36] and the intensity of noise [26]. When the noise intensity is $\sigma \leq 20$, $20 < \sigma \leq 40$, and $\sigma > 40$, the picture block size is set to 6×6, 7×7, and 8×8, and T is set to 2, 3, 4, and $\tau_1 = 0.1$, $\tau_2 = 0.2$, λ set to 0.23, 0.28, 0.36, and c_1 set to 0.56, 0.57, c_2 set to 0.59, 0.64, and clustering numbers set to 240, 240, 250, respectively.

Fig.2 is the test image used in this experiment. Fig.2(a) to Fig.2(f) are grayscale images from the Set12 data set. Fig.2(g) to Fig.2(i) are real noise images from the RENOIR data set. Fig.2(k) to Fig.2(l) are texture images from the USC-SIPI data set. Fig.2(m) is a color image from the CBSD68 dataset.

A. GRAYSCALE NOISY IMAGES

In this section, the paper focuses on the denoising case of grayscale image degradation contaminated by Gaussian white noise. Based on the previous discussion, the proposed new algorithm (FCM-SDL) is compared with model-based algorithms (ISKR, EPLL, LR-GSC, and NCSR) and learning-based algorithms (DnCNN, FFDNet, and IRCNN). To provide a fair comparison, high-quality images were combined with different levels of the same Gaussian white noise to create synthetic noise images with different noise levels of 10, 30, and 50. Then, we provide the visual denoising results of several algorithms under different noise levels and introduce PSRN and SSIM to evaluate the objective quality of denoised images. Fig.3, Fig.4, Fig.5, and Fig.6 show different models' overall and local results for restoring images for Gaussian white noise removal, with standard deviations of 15, 30, and 50, respectively.

When the noise level is low, the visual effect of each algorithm is shown in Fig.3. Fig.3 shows the overall and local results of each algorithm restoring the image when the noise level is 10 for the grayscale image. After the ISKR algorithm is processed, the image is too smooth, and the edge information of the "building" part is lost. In the image processing by KSVD and LR-GSC algorithm, the edge of the "shoulder" part is blurred. The "sky" in the smooth area processed by the DnCNN algorithm has a block effect. The FFDNet and IRCNN algorithms have a better effect on the processed image when the noise level is low, but the model trained 5000 times runs as long as 3 to 5 hours.

Fig.4 shows the overall and local results of each algorithm restoring the image when the noise level is 30 for the grayscale image. The image processed by the ISKR algorithm is severely deformed overall and locally, but the local result of the "eye" is not blurred. Both the "texture" local result and the "eye" local result after the KSVD algorithm is blurred. EPLL and NCSR algorithms have "texture" local result distortion, but "eye" local result details are preserved. The LR-GSC algorithm relatively completely preserves the local texture information of the image, but the local result of the "eye" is blurred. After DnCnn, FFDNet, and IRCNN algorithms remove noise, the image's texture details and eye details are lost and blurred to varying degrees. The FCM-SDL algorithm preserves the local result of "texture" and the local result of "eye."

When the noise level is high, the visual effects of each algorithm are shown in Fig.5 and Fig.6. Fig.5 and Fig.6 are

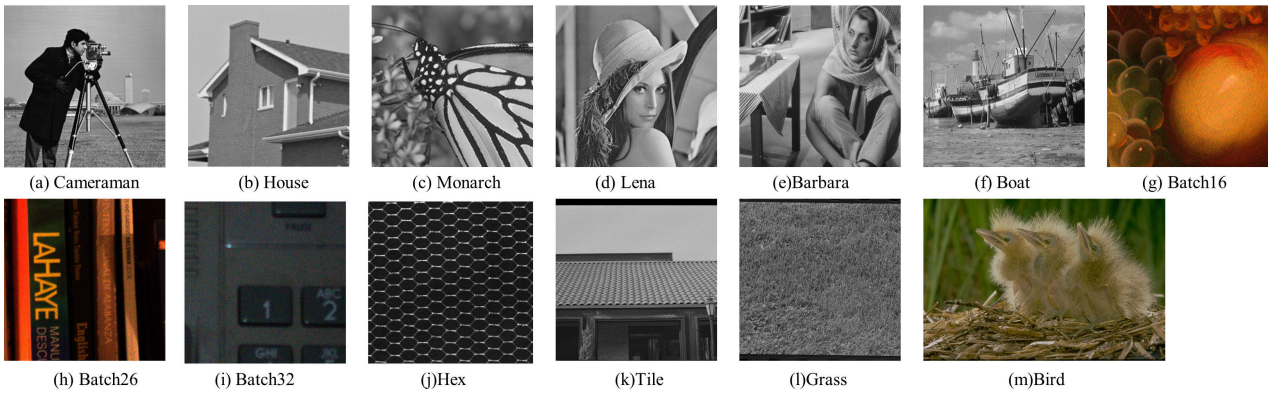


FIGURE 2. Different types of test images from different data sets used in the experiment.

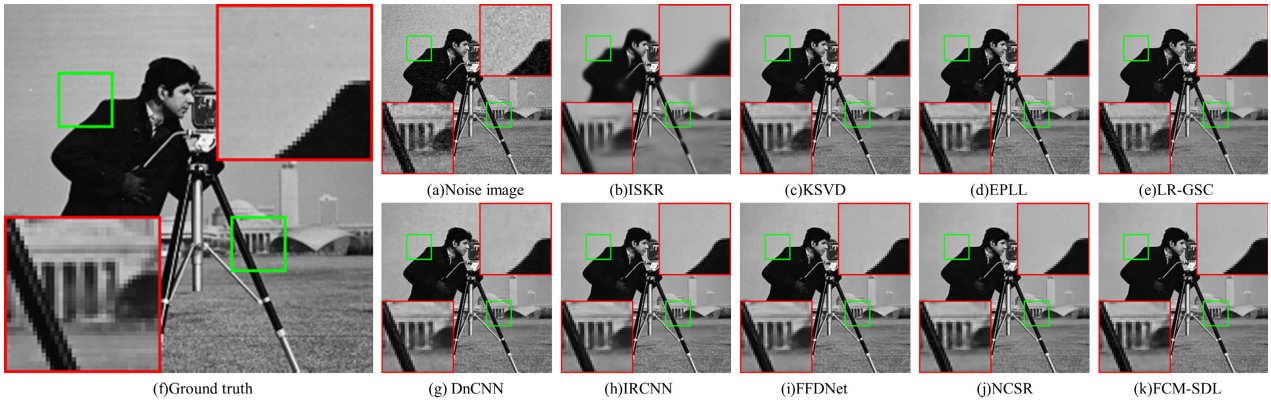


FIGURE 3. Cameraman image denoising results (the noise images with $\sigma = 10$).

FIGURE 4. Barbara image denoising results (the noise images with $\sigma = 30$).

the overall and local results of the restored images of each algorithm when the noise level is 50 for the grayscale image. The simple image (''house'' image) results are shown in Fig.5. The ISKR algorithm is not clean for noise removal, which seriously affects the visual effect. The image processed by KSVD and LR-GSC algorithm has a blurring phenomenon in local results. On the whole, there are plaque effects in KSVD, EPLL, DnCNN, FFDNet, IRCNN, and NCSR, and the partial image of the ''eaves'' on the left side of the image processed by KSVD, DnCNN, and FFDNet algorithms has a noticeable edge loss, and the edges on both sides of EPLL are preserved. Better. The overall image visual effect of the LR-GSC and FCM-SDL algorithm is good, and the edge information is preserved. The complex image (''monarch'' image) results are shown in Fig.6. When the noise level is high for images with complex structures, the ISKR algorithm does not remove the noise. The edges of the left part of the KSVD and LR-GSC algorithms are blurred. EPLL algorithm has a

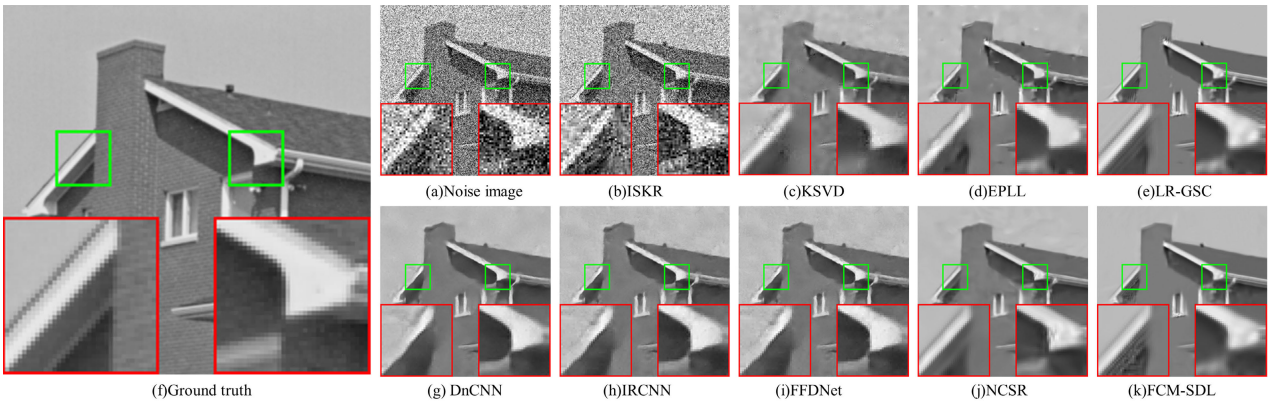


FIGURE 5. House image denoising results (the noise images with $\sigma = 50$).

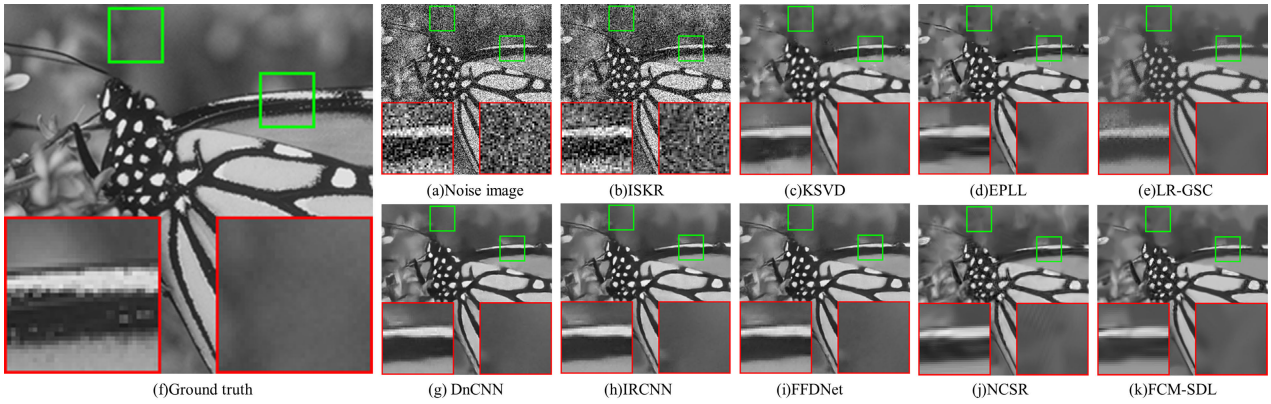


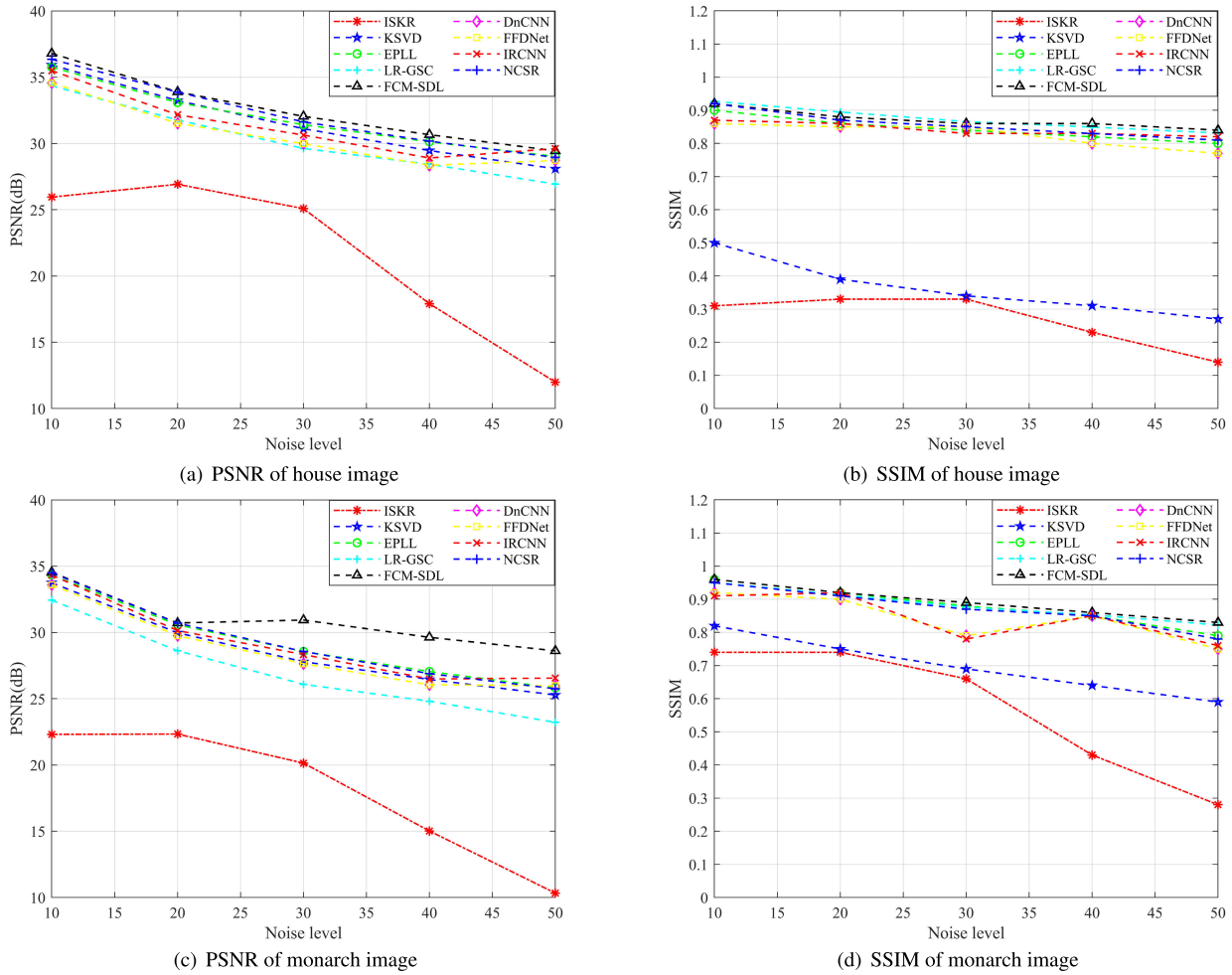
FIGURE 6. Monarch image denoising results (the noise images with $\sigma = 50$).

blocking effect. On the right side of the NCSR algorithm, the enlarged part can see the strip deformation in the smooth area. DnCNN, FFDNet, IRCNN, and FCM-SDL maintain the overall image structure, but DnCNN, FFDNet, and IRCNN can see that the pixels at the ''petals'' in the upper left area of the ''monarch'' image are blurred.

Fig.7 shows the PSNR and SSIM curves of images with simple structures (''house'' images) and images with complex structures (''monarch'' images) at different noise levels. Fig.7(a) and Fig.7(b) are the PSNR and SSIM curves of the house image, and Fig.7(c) and Fig.7(d) are the PSNR and SSIM curves of the monarch image. As the noise level increases, the PSNR and SSIM of the proposed algorithm show better performance in both simple and complex images, indicating that it has certain robustness.

We also compared the average running time of each algorithm for images in the Set12 dataset under different noise levels, as shown in Fig.8. Fig.8 shows that the running time of the FCM-SDL algorithm is at the middle level, almost the same as the running time of the LR-GSC and NCSR algorithms. However, the FCM-SDL algorithm can retain more texture and detailed information when removing noise and still maintains a good performance at a high noise level. The training time is significantly reduced compared to learning-based algorithms, and the FCM-SDL algorithm is more interpretable compared with the learning-based algorithms.

In the subjective results, we can see that other denoising methods often remove details such as edges and textures of the image when removing noise, which makes the denoising result too smooth. When the noise level is low, the ISKR algorithm can effectively remove the noise, but the reconstructed image will be deformed; but for a high level of noise, the ISKR algorithm cannot remove the noise completely. The image processed by the LR-GSC algorithm maintains some image structure but still loses the edge and texture information of the image. The edge of the image after denoising by DnCNN and IRCNN algorithm is too smooth. FFDNet is slightly better than DnCNN and IRCNN and preserves the structure of the image to a certain extent, but the edges of the image are still too smooth. The main reason is that these learning-based methods are limited to extracting high-frequency features. On the contrary, our algorithm can reconstruct local details because adding FCM clustering constraints in our method can better capture the global structural information of the image, and the learned dictionary under



(a) PSNR of house image

(b) SSIM of house image

(c) PSNR of monarch image

(d) SSIM of monarch image

FIGURE 7. **Comparison of the denoising performances of different algorithms (the noise images with σ = 10, 20, 30, 40, 50).**

clustering constraints can integrate prior external information and the internal structure features of the image to help the reconstruction.

To compare the performance more specifically, we compared each algorithm's PSNR and SSIM indicators after denoising the grayscale image with synthetic noise. Table 1 lists the PSNR and SSIM values of images restored by different algorithms (the bold ones are the maximum values of PSNR and SSIM for each row). As can be seen from Table 1, the proposed model is numerically superior in terms of PSNR and SSIM. Compared with the ISKR algorithm, KSVD algorithm, EPLL algorithm, LR-GSC algorithm, and NCSR algorithm, the PSNR has increased by 10.915dB, 0.992dB, 1.137dB, 2.096dB, and 0.3755dB on average, and the SSIM has increased by 0.5065, 0.992, 0.401, 1.137, 0.0765 on average, 2.096, 0.0195, 0.3755, 0.0655. Compared with the learning-based algorithms (DnCNN, IRCNN, FFDNet), the PSNR has an average increase of 1.237dB, and the SSIM has an average increase of 0.072. Numerical results demonstrate that our proposed denoising method improves image evaluation metrics over model-based and learning-based algorithms. At various noise levels, there are both learning-based and model-based algorithms with excellent PSNR and SSIM values.

By comparing the visual effect, average running time, and numerical results of the restored images, it can be seen that the proposed algorithm achieves the best result in evaluating the three aspects. The proposed algorithm has a short running time and interpretability. It has the best visual effect at all noise levels, and excellent PSNR and SSIM values in both learning- and model-based algorithms.

B. COLOR SCALE NOISY IMAGES

In this section, the paper focuses on the denoising case of color image degradation contaminated by Gaussian white noise, comparing the proposed FCM-SDL algorithm with model-based algorithms (ISKR and BM3D) and learning-based algorithms (DnCNN, FFDNet, and IRCNN). A comparison is made to verify the effectiveness of the proposed algorithm in color image denoising. Based on the denoising results of grayscale images at various noise levels, this section combines the high-quality image with the same

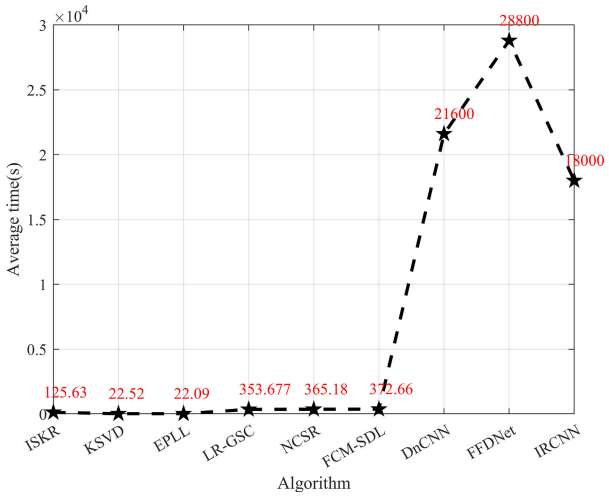


FIGURE 8. Average time of each algorithm to remove Gaussian white noise.

Gaussian white noise at level 50 to create a synthetic color noise image. Then, we provide the visual denoising results of several algorithms and introduce two evaluation indicators, PSNR and SSIM, to evaluate the objective quality of denoised images.

The denoising results of different methods for color images in the case of $\sigma = 50$ are shown in Fig.9. In the subjective results, we can find that other denoising methods tend to remove edge details along with the noise, which makes the result too smooth. ISKR can neither preserve image structure nor remove noise deeply. BM3D [4] preserves the structure of the image to a certain extent but fails to remove noise deeply. DnCNN [14] and IRCNN [15] algorithms will over-smooth the edges of the image, and FFDNet [16] is slightly better than the first two methods. Our algorithm can integrate prior external information and internal structural features of the image under the constraints of FCM clustering to help reconstruct the restored image. In the objective numerical comparison, compared with other algorithms, the FCM-SDL algorithm achieved the best PSNR and SSIM values among the algorithms. FCM-SDL algorithm achieved at least 0.57dB and 0.0072 improvements in PSNR and SSIM values. Comparing the visual effects and numerical results of the restored color image comprehensively, it can be seen that the proposed algorithm has achieved the best results in the comprehensive evaluation. The experiments in this section show that the FCM-SDL algorithm proposed in this paper can also retain the details and texture of the image while removing the noise and obtaining a better visual effect than the grayscale image.

C. REAL WHITE NOISE REMOVAL

In this section, the paper focuses on the denoising case of the real noisy image. We compared the FCM-SDL algorithm with the ISKR algorithm, NCSR algorithm, DnCNN algorithm, FFDNet algorithm, and IRCNN algorithm, aiming to verify the proposed algorithm's effectiveness in denoising real noisy images. We provide the visual denoising results of several algorithms and introduce PSRN, SSIM, SNR, and figure of merit (FOM) to evaluate the objective quality of denoised images.

Fig.10 shows the overall and local results of different models for the restored image with real noise removed. The first line of Fig.10 is the denoising result of the image "batch16", the second line of Fig.10 is the denoising result of the image "batch26", and the third line of Fig.10 is the denoising result of the image "batch32". It can be seen that the DnCNN[14] algorithm and the IRCNN[15] algorithm make the left side of the "batch16" image blurred, the character '2008' in the image "batch26" is blurred, the edges of the buttons in the image "batch32" are blurred, and the overall image is blurred. Texture details are lost, block effects exist, and noise removal is not clean. FFDNet [14] is slightly better than the first two methods, but the overall effect is not good. The edge and texture information of the ISKR algorithm is well preserved, but there are local deformations. The edge and texture information of the NCSR algorithm is well preserved, but the local noise removal is not clean. The noise of the FCM-SDL algorithm is removed cleanly, and the edge and texture information of the image is kept well.

In order to compare the performance more objectively, we compared the PSNR, SSIM, SNR, and FOM of each algorithm after denoising the real noise image. Table 2 lists the PSNR, SSIM, SNR, and FOM values of images recovered by different algorithms. In the objective numerical comparison, the FCM-SDL algorithm obtained the best PSNR, SSIM, SNR, and FOM values among all algorithms. The average increase of PSNR was 3.158dB, SSIM was 0.067, SNR was 5.539dB, and FOM was 0.064.

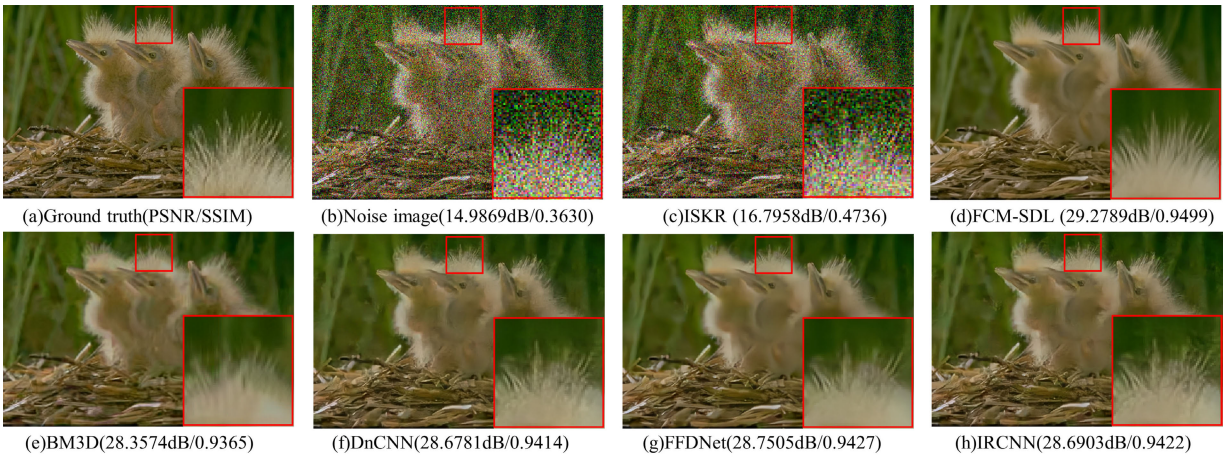
D. THE FEASIBILITY OF THE PROPOSED ALGORITHM

In this section, this article focuses on the feasibility of the proposed algorithm. The proposed new model (FCM-SDL) is compared with the NFCM-SDL algorithm and NCSR algorithm without FCM constraints, aiming to verify the effectiveness of the proposed algorithm for image structure preservation during image denoising and the novelty of the FCM-SDL algorithm. This experiment was performed using one edge structure image, "Hex," and two texture images, "Tile" and "Grass," all of which were destroyed by Gaussian white noise $\sigma = 50$.

Fig.11 shows the overall and local results of the recovered image of the edge structure image at $\sigma = 50$. It can be seen from the figure that the denoising image processed by the algorithm without adding FCM constraints has a block effect as a whole, and the structure is blurred at the magnification part. However, the overall structure of the denoising image processed by the NCSR algorithm and FCM-SDL algorithm is clear, but the FCM-SDL is clearer at the local enlargement. The values of the FCM-SDL algorithm PSNR, SSIM, and SNR are improved by 0.1054dB, 0.0011, and 0.2389dB compared to the NCSR algorithm and by 0.6018dB, 0.0401, and 0.6019dB compared to the NFCM-SDL algorithm.

TABLE 1. PSNR and SSIM of each algorithm under Gaussian white noise.

σ	Image	model-based												learning-based					
		ISKR		KSVD		EPLL		LR-GSC		NCSR		FCM-SDL		DnCNN		FFDNet		IRCNN	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
10	Cameraman	21.19	0.31	33.65	0.55	34.05	0.93	32.85	0.92	33.94	0.93	34.05	**0.94**	33.40	0.90	**34.22**	0.91	33.98	0.89
	Lena	25.89	0.47	35.49	0.62	35.61	0.91	34.18	0.93	35.86	0.91	**35.89**	**0.97**	34.73	0.87	35.64	0.88	35.40	0.87
	Barbara	22.98	0.67	34.34	0.84	33.77	0.95	32.85	0.95	35.02	0.94	**35.11**	**0.97**	32.61	0.91	33.61	0.93	33.23	0.92
	Boat	23.01	0.41	33.63	0.68	33.67	0.87	32.19	0.97	33.65	0.89	**33.85**	**0.97**	33.13	0.88	33.72	0.90	33.55	0.87
20	Cameraman	22.18	0.34	29.94	0.44	30.38	0.88	28.52	0.87	30.29	0.87	**30.45**	**0.89**	29.51	0.85	30.17	0.87	29.94	0.86
	Lena	26.28	0.49	32.42	0.51	32.61	0.87	31.23	0.95	32.91	0.87	**32.96**	**0.95**	31.65	0.82	32.24	0.86	31.98	0.85
	Barbara	23.25	0.70	30.42	0.75	29.95	0.89	29.57	0.91	31.24	0.90	**31.81**	**0.95**	28.41	0.85	29.15	0.86	28.76	0.83
	Boat	23.72	0.45	30.33	0.53	30.70	0.82	28.93	0.92	30.75	0.82	**30.75**	**0.93**	29.83	0.83	30.35	0.85	30.16	0.83
30	Cameraman	21.14	0.36	27.97	0.37	28.40	0.84	26.46	0.81	28.52	0.84	**28.63**	**0.85**	27.59	0.79	28.32	0.80	28.21	0.78
	Lena	24.48	0.46	30.12	0.45	28.45	0.84	29.19	0.89	31.12	0.84	**31.14**	**0.92**	29.68	0.76	30.29	0.78	30.31	0.77
	Barbara	21.42	0.66	27.89	0.65	27.67	0.82	27.37	0.86	29.74	0.87	**30.75**	**0.93**	26.08	0.80	26.47	0.82	26.52	0.80
	Boat	22.58	0.49	28.42	0.44	28.86	0.77	26.85	0.88	28.91	0.77	**28.95**	**0.88**	27.92	0.77	28.26	0.78	28.34	0.78
40	Cameraman	16.21	0.28	26.73	0.33	27.09	0.80	25.30	0.79	27.12	0.80	**27.19**	**0.80**	26.12	0.80	26.92	0.82	26.43	0.83
	Lena	17.36	0.27	28.83	0.41	27.00	0.81	28.09	0.91	29.91	0.85	**29.94**	**0.91**	28.07	0.88	29.03	0.90	28.56	0.87
	Barbara	15.90	0.44	26.20	0.57	26.05	0.76	26.11	0.91	27.56	0.83	**28.37**	**0.91**	24.45	0.86	24.93	0.87	24.77	0.87
	Boat	16.72	0.33	27.07	0.38	27.55	0.73	25.73	0.86	27.59	0.73	**27.71**	**0.86**	26.33	0.79	27.01	0.82	26.74	0.81
50	Cameraman	11.35	0.20	25.76	0.29	26.09	0.78	23.87	0.76	26.13	0.78	**26.18**	**0.78**	26.26	0.78	27.00	0.76	26.95	0.77
	Lena	11.28	0.13	27.84	0.37	25.90	0.82	26.92	0.88	28.90	0.80	28.93	**0.89**	28.40	0.76	**29.12**	0.79	29.11	0.80
	Barbara	11.12	0.25	24.84	0.49	24.90	0.70	24.88	0.87	24.34	0.80	**27.22**	**0.89**	24.11	0.81	25.40	0.83	25.26	0.82
	Boat	11.25	0.18	25.88	0.33	26.17	0.70	24.60	0.79	26.60	0.70	**27.73**	**0.83**	26.42	0.74	26.95	0.73	26.90	0.71



(a)Ground truth(PSNR/SSIM) (b)Noise image(14.9869dB/0.3630) (c)ISKR (16.7958dB/0.4736) (d)FCM-SDL (29.2789dB/0.9499)

(e)BM3D(28.3574dB/0.9365) (f)DnCNN(28.6781dB/0.9414) (g)FFDNet(28.7505dB/0.9427) (h)IRCNN(28.6903dB/0.9422)

FIGURE 9. Color scale image denoising results (the noise images with $\sigma = 50$).

TABLE 2. PSNR, SSIM and SNR of each algorithm under real noise.

Image	ISKR				NCSR				FCM-SDL				DnCNN				FFDNet				IRCNN			
	PSNR	SSIM	SNR	FOM	PSNR	SSIM	SNR	FOM	PSNR	SSIM	SNR	FOM	PSNR	SSIM	SNR	FOM	PSNR	SSIM	SNR	FOM	PSNR	SSIM	SNR	FOM
Batch16	29.24	0.97	20.46	0.98	33.62	0.96	23.85	0.98	**33.85**	**0.99**	**25.07**	**0.99**	30.42	0.97	21.64	0.98	30.93	0.98	22.15	0.98	30.85	0.98	22.07	0.98
Batch26	30.22	0.84	18.44	0.95	31.22	0.88	20.37	0.97	**34.39**	**0.96**	**22.61**	**0.99**	29.51	0.83	17.73	0.96	30.30	0.87	18.52	0.95	30.24	0.86	18.47	0.94
Batch32	28.39	0.46	14.03	0.99	32.07	0.79	18.88	0.99	**34.49**	**0.80**	**20.14**	**1.00**	32.77	0.77	18.41	0.95	33.53	0.80	19.17	0.99	32.97	0.78	18.62	0.99

Fig.12 shows the overall and local results of the restored image of the texture image at $\sigma = 50$. It can be seen from the figure that the denoising image processed by the NFCM-SDL algorithm and NCSR algorithm has a block effect in places with many smooth components, the edges at the magnified local "tiles" are blurred, the edges at the

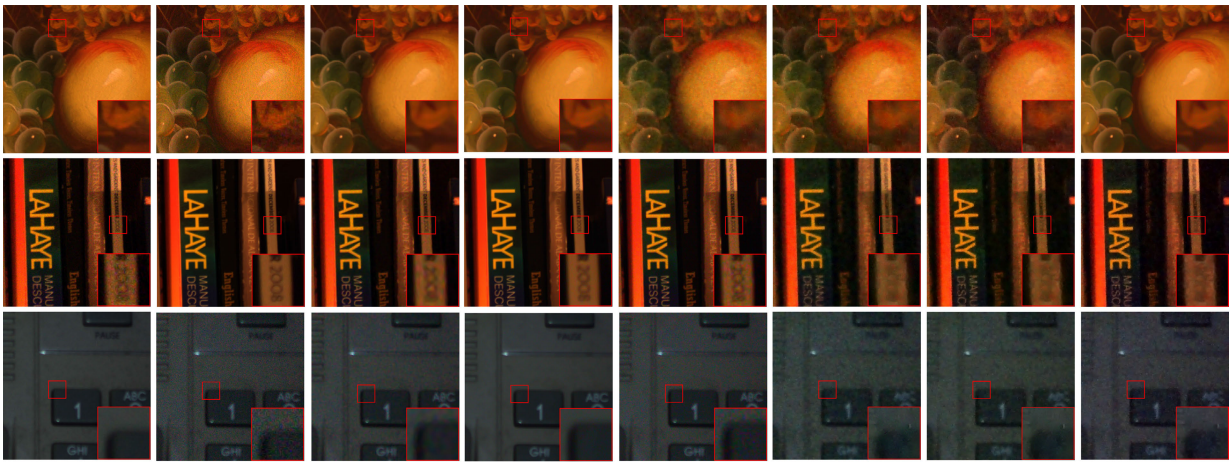


FIGURE 10. Visual comparison of different algorithms for removing real noise.

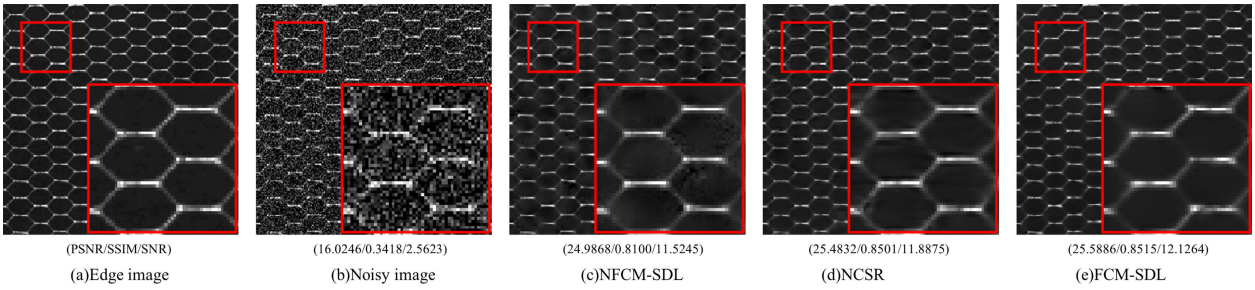


FIGURE 11. Edge image denoising results(the noise images with $\sigma = 50$).

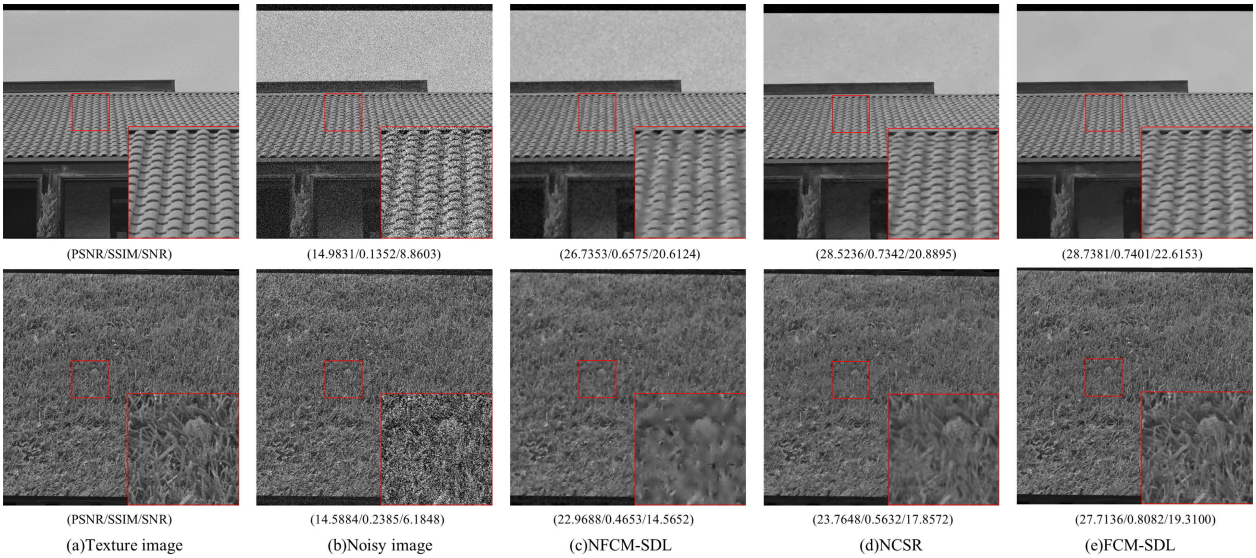


FIGURE 12. Texture image denoising results(the noise images with $\sigma = 50$).

zoomed local ''grass'' are lost. The ''grass'' image is blurred as a whole. However, the overall structure of the denoising image processed by the algorithm adding FCM constraint is clear, the edge at the zoomed ''tile'' is clear, the edge at the zoomed ''grass'' is visible, and the overall structure of the image is maintained. Compared with the NFCM-SDL algorithm PSNR, SSIM, and SNR, the FCM-SDL algorithm improves the values of 0.6018dB, 0.0401, and 0.6019dB in

Tile images, respectively. In Grass, the image is improved by 2.0028dB, 0.3429, and 4.7448dB, respectively. Compared with the NCSR algorithm PSNR, SSIM, and SNR, the FCM-SDL algorithm improves the values of 0.6018dB, 0.0401, and 0.6019dB in Tile images, respectively. In Grass, the image is improved by 2.0028dB, 0.3429, and 4.7448dB, respectively. The visual and numerical results of denoising images with complex texture structures show that FCM clustering maintains the effectiveness of image structure for image denoising. The experimental results also show that although both FCM-SDL and NCSR are based on clustering algorithms, their denoising effects are different, and the FCM-SDL algorithm is novel.

V. CONCLUSION

Because the traditional dictionary learning algorithm does not consider the inherent structural features of image pixels, resulting in image over-smoothing, image texture, edge information loss, and other problems, a novel FCM-SDL denoising algorithm combining dictionary learning and FCM structure clustering is proposed. Firstly, the FCM-SDL algorithm can effectively shorten the time of generating the learning dictionary by using the PCA method through the external loop. Secondly, the FCM-SDL algorithm uses the FCM clustering method to constrain the learning dictionary through the inner loop, which makes up for the defect that the existing sparse representation-based image denoising methods do not consider the internal structure of the image. Thirdly, the learning dictionary under the constraint of inner and outer loop iteration can improve the efficiency of image denoising while maintaining the texture and edge details. Finally, sparse representation coefficients and redundant dictionaries are obtained by using the OMP method and alternating direction multiplication. According to the sparse coding theory, the denoised image is obtained. This algorithm is superior to KSVD, ISKR, EPLL, LR-GSC, DnCNN, IRCNN, and FFDNet in both subjective effect and objective evaluation. This method can effectively remove white Gaussian noise in gray images and color image and successfully remove real noise. At the same time, the edge and texture information of the image is retained, which enhances the subjective effect and increases the objective evaluation value. Richly textured images can provide better visuals, but images with smoother components will be worse than richly textured images. That is what we are going to deal with in the future. Through the research of plug and play algorithm and expansion algorithm, it is feasible to combine the learning-based algorithm with the model-based algorithm. For the texture part, the model-based algorithm can give more constraints. The smooth region can rely on the learning-based algorithm to process and synthesize different types of image blocks to get better image processing results.

APPENDIX A
THE SIMPLIFICATION PROCESS OF FCM-SDL MODEL

To comprehend how the double-$\ell 1$ optimization problem is solved, first explain how the single-$\ell 1$ optimization problem was solved. The single-$\ell 1$ optimization problem is described as follows:

$$\boldsymbol{\alpha} = \underset{\alpha}{argmin}\ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{D\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{A-1}$$

Assuming $\boldsymbol{D}$ is a unitary matrix, in terms of $\boldsymbol{DD}^T = \boldsymbol{I}$, the objective function is:

$$\begin{aligned} f(\boldsymbol{\alpha}) &= \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{D\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \\ &= \frac{1}{2}\left\|\boldsymbol{D}(\boldsymbol{D}^T x - \boldsymbol{\alpha})\right\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \\ &= \frac{1}{2}\|\boldsymbol{D}(\boldsymbol{\alpha}_0 - \boldsymbol{\alpha})\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \\ &= \frac{1}{2}\|\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \end{aligned} \tag{A-2}$$

where $\boldsymbol{\alpha}_0 = \boldsymbol{D}^T x$ and the unitary matrix's characteristic $\|\boldsymbol{x}\|_2^2 = \|\boldsymbol{Dx}\|_2^2$. The objective function is diagonalized as a result of the process described above, the objective function becomes,

$$f(\boldsymbol{\alpha}) = \sum_i \left[\frac{1}{2}(\boldsymbol{\alpha}_0(i) - \boldsymbol{\alpha}(i))^2 + \lambda\,|\boldsymbol{\alpha}(i)|\right] \tag{A-3}$$

The scalar minimization issue is made more accessible by the scalar form, which directly replaces the vector form:

$$g(t) = \frac{1}{2}(t - t_0)^2 + \lambda\,|t| \tag{A-4}$$

By using a mathematical approach to solve the issue of the minimal value of one variable function, it is possible to derive the generalized contraction operator of the optimization problem with a single-$\ell 1$ constraint,

$$\boldsymbol{S}_\lambda(t) = \begin{cases} 0, |t_0| \le \lambda \\ t_0 - sgn(t_0)\lambda, |t_0| > \lambda \end{cases} \tag{A-5}$$

Apply the same method to obtain a generalized contraction operator for the double-$\ell 1$ optimization problem above.

APPENDIX B
ILLUSTRATION OF THE MATHEMATICAL SIMPLIFICATION OPERATIONL

The specific simplification process of Eq.(15) to Eq.(16) is explained as follows:

$$\begin{aligned} f(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\alpha}_0) = &\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{D\alpha}\|_2^2 + \mu_1\|\boldsymbol{\alpha}\|_1 + \mu_2\|\boldsymbol{\alpha} - \boldsymbol{\gamma}\|_1 \\ &+\frac{c}{2}\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\right\|_2^2 - \frac{1}{2}\|\boldsymbol{D\alpha} - \boldsymbol{D\alpha}_0\|_2^2 \end{aligned} \tag{B-1}$$

Because of $\boldsymbol{\alpha}_0 = \boldsymbol{D}^T x$, the last term in Eq.(15) can be written as,

$$\frac{1}{2}\|\boldsymbol{D\alpha} - \boldsymbol{D\alpha}_0\|_2^2 = \frac{1}{2}\left\|\boldsymbol{D\alpha} - \boldsymbol{DD}^T x\right\|_2^2 = \frac{1}{2}\|\boldsymbol{D\alpha} - \boldsymbol{x}\|_2^2 \tag{B-2}$$

Since $\boldsymbol{y}$ is $\boldsymbol{x}$ with noise, $\|\boldsymbol{y} - \boldsymbol{D\alpha}\|_2^2$ is the difference between $\boldsymbol{y}$ and $\boldsymbol{x}$. $\|\boldsymbol{D\alpha} - \boldsymbol{x}\|_2^2$ is $\boldsymbol{D\alpha} \approx \boldsymbol{x}$.Eq.(14) has too many

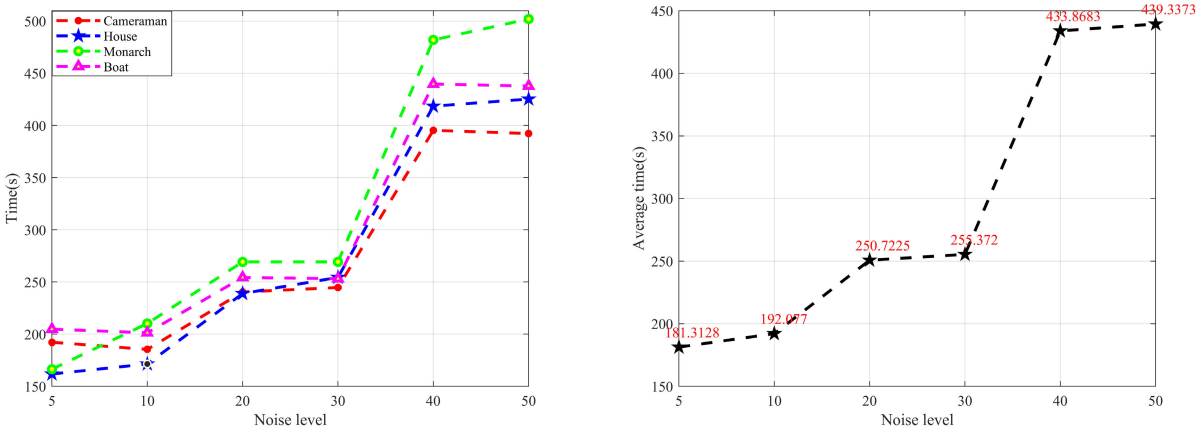


FIGURE 13. Comparison of image denoising running times in Set12 data set (σ = 5, 10, 20, 30, 40, 50 noisy images).

variables, making it challenging to solve mathematically. The number of variables needs to be reduced. Because $\boldsymbol{\alpha}_0 = \boldsymbol{D}^T\boldsymbol{x}$, $\left\|\boldsymbol{D}^T\boldsymbol{x} - \boldsymbol{\alpha}_0\right\|_2^2 = 0$, plug $\left\|\boldsymbol{D}^T\boldsymbol{x} - \boldsymbol{\alpha}_0\right\|_2^2 = 0$ in the fourth term, and the variable is reduced; as a result, Eq.(B–2) becomes,

$$\frac{c}{2}\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\right\|_2^2 = \frac{c}{2}\left\|\boldsymbol{\alpha} - \frac{1}{c}(\boldsymbol{D}^T\boldsymbol{x} - \boldsymbol{\alpha}_0) - \boldsymbol{\alpha}_0\right\|_2^2 \quad \text{(B-3)}$$

Based on this, Eq.(B-4) can be mathematically reduced to (B-5), and the mathematical solution of Eq.(16) can be obtained.

$$f(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\alpha}_0) = const + \mu_1\|\boldsymbol{\alpha}\|_1 + \mu_2\|\boldsymbol{\alpha} - \boldsymbol{\gamma}\|_1 + \frac{c}{2}\left\|\boldsymbol{\alpha} - \boldsymbol{v}_0\right\|_2^2 \quad \text{(B-4)}$$

where $\boldsymbol{v}_0$ is equal to $\boldsymbol{v}_0 = \frac{1}{c}\boldsymbol{D}^T(\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}_0) + \boldsymbol{\alpha}_0$.

$$\boldsymbol{\alpha}_j^{(i+1)} = \begin{cases} \boldsymbol{S}_{\tau_1,\tau_2,\boldsymbol{\gamma}_j}(\boldsymbol{v}_j^{(i)}), \boldsymbol{\gamma}_j \geq 0 \\ -\boldsymbol{S}_{\tau_1,\tau_2,-\boldsymbol{\gamma}_j}(-\boldsymbol{v}_j^{(i)}), \boldsymbol{\gamma}_j < 0 \end{cases} \quad \text{(B-5)}$$

Eq.(B-1) is the same as Eq.(15), and Eq.(B-4) is the same as Eq.(16). Therefore,, the variable is reduced, and the mathematical operation from Eq.(15) to Eq.(15) is described above.

APPENDIX C
ALGORITHMIC COMPLEXITY OF THE FCM-SDL ALGORITHM

The algorithm complexity can be described as time complexity and space complexity. We will analyze our proposed algorithm from two aspects using the big O notation. The big O notation is a mathematical notation used to describe the asymptotic behavior of functions.

First, we analyze the time complexity of this algorithm. In the running process of the FCM-SDL algorithm, there are binary search statements with ''if'' and ''else'' in pairs whose time complexity is $O(logn)$. FCM-SDL algorithm also has loop nesting, there is quadratic loop nesting, and its time complexity is $O(n^2)$; The maximum cycle nesting of the whole algorithm is three times; that is, when classifying the image, it is necessary to calculate the image blocks of row i and column j and arrange the classified ones, so the time complexity of the cycle nesting of three times is $O(n^3)$. So, sort by time complexity.

Order the time complexity from smallest to largest: $O(1) < O(logn) < O(n) < O(nlogn) < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n^n)$.

Therefore, the time complexity of the FCM-SDL algorithm is $O(n^3)$.

We also drew the time graph of the FCM-SDL algorithm based on the Set12 data set. Fig.13(a) shows the time used by the proposed algorithm to make noise in the Set12 data set with different noise levels, and Fig.13(b) shows the average time used by the proposed algorithm to make noise in the Set12 data set with different noise levels. With the increase in image noise, the algorithm's running time will be longer. However, the complexity of the proposed algorithm is low, the maximum nesting in the algorithm operation is three cycles, and the algorithm's complexity is $O(n^3)$.

Space complexity measures the amount of storage space temporarily occupied by an algorithm during its run. Space complexity is the number of variables. The number of variables in the FCM-SDL algorithm is 105. Therefore, its space complexity is $O(105)$.

In summary, the time complexity of the FCM-SDL algorithm is $O(n^3)$, and the space complexity is $O(105)$.

REFERENCES

[1] C. A. Micchelli, L. Shen, Y. Xu, and X. Zeng, ''Proximity algorithms for the L1/TV image denoising model,'' *Adv. Comput. Math.*, vol. 38, no. 2, pp. 401–426, Feb. 2013.
[2] J. Tian, L. Chen, and L. Ma, ''A wavelet-domain non-parametric statistical approach for image denoising,'' *IEICE Electron. Exp.*, vol. 7, no. 18, pp. 1409–1415, Sep. 2010.
[3] A. Buades, B. Coll, and J.-M. Morel, ''A non-local algorithm for image denoising,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 60–65.
[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, ''Image denoising by sparse 3-D transform-domain collaborative filtering,'' *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.
[5] M. Elad and M. Aharon, ''Image denoising via sparse and redundant representations over learned dictionaries,'' *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[7] M. Scetbon, M. Elad, and P. Milanfar, "Deep K-SVD denoising," *IEEE Trans. Image Process.*, vol. 30, pp. 5944–5955, 2021.
[8] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. ICCV*, Barcelona, Spain, 2011, pp. 479–486.
[9] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.
[10] Z. Zha, B. Wen, X. Yuan, J. Zhou, and C. Zhu, "Reconciliation of group sparsity and low-rank models for image restoration," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.
[11] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2392–2399.
[12] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.
[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[14] 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.
[15] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2808–2817.
[16] 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.
[17] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1712–1722.
[18] Y. Chang, L. Yan, M. Chen, H. Fang, and S. Zhong, "Two-stage convolutional neural network for medical noise removal via image decomposition," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 2707–2721, Jun. 2020.
[19] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 3155–3164.
[20] J. Yang, L. Luo, J. Qian, Y. Tai, F. Zhang, and Y. Xu, "Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 156–171, Jan. 2017.
[21] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6202–6216, Dec. 2012.
[22] S. Zhang, Y. Li, D. Cheng, Z. Deng, and L. Yang, "Efficient subspace clustering based on self-representation and grouping effect," *Neural Comput. Appl.*, vol. 29, no. 1, pp. 51–59, Jan. 2018.
[23] H. Wang, S. Zhou, L. Yu, and J. Zhao, "Adaptive filtering fuzzy C-means image segmentation with inclusion degree," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Tianjin, China, Aug. 2019, pp. 1637–1641.
[24] A. Khmag, A. R. Ramli, and N. Kamarudin, "Clustering-based natural image denoising using dictionary learning approach in wavelet domain," *Soft Comput.*, vol. 23, no. 17, pp. 8013–8027, Sep. 2019.
[25] A. Khmag, A. R. Ramli, S. A. R. Al-Haddad, S. J. Hashim, Z. M. Noh, and A. A. M. Najih, "Design of natural image denoising filter based on second-generation wavelet transformation and principle component analysis," *J. Med. Imag. Health Informat.*, vol. 5, no. 6, pp. 1261–1266, Nov. 2015.
[26] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.
[27] J. Bobin, J.-L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho, "Morphological component analysis: An adaptive thresholding strategy," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2675–2681, Nov. 2007.
[28] S. Voronina and H. J. Woerdeman, "A new iterative firm-thresholding algorithm for inverse problems with sparsity constraints," *Appl. Comput. Harmon. Anal.*, vol. 35, no. 1, pp. 151–164, Jul. 2007.
[29] H. He, W.-J. Lee, D. Luo, and Y. Cao, "Insulator infrared image denoising method based on wavelet generic Gaussian distribution and MAP estimation," *IEEE Trans. Ind. Appl.*, vol. 53, no. 4, pp. 3279–3284, Jul. 2017.
[30] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, no. 4, pp. 1531–1549, 2010.
[31] H. Shen, L. Peng, L. Yue, Q. Yuan, and L. Zhang, "Adaptive norm selection for regularized image restoration and super-resolution," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1388–1399, Jun. 2016.
[32] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.
[33] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1586–1595.
[34] S. Kumar Sahoo and W. Lu, "Image denoising using sparse approximation with adaptive window selection," in *Proc. 8th Int. Conf. Inf., Commun. Signal Process.*, Singapore, Dec. 2011, pp. 1–5.
[35] W. Sun and Q. Du, "Graph-regularized fast and robust principal component analysis for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3185–3195, Jun. 2018.
[36] L. Jia, S. Song, L. Yao, H. Li, Q. Zhang, Y. Bai, and Z. Gui, "Image denoising via sparse representation over grouped dictionaries with adaptive atom size," *IEEE Access*, vol. 5, pp. 22514–22529, 2017.

CHANGPENG JI received the master's degree in computer application technology from Liaoning Technical University, in 2005. He is currently a Full Professor and a Master's Supervisor at Liaoning Technical University. He is also the Codesys Senior Application Engineer, the Senior Artificial Intelligence Designer, the Academic Leader of Information and Communication Engineering, the Key Member of the Outstanding Young Teacher in Liaoning Province, in 2006, the Expert in Discipline Assessment and Dissertation Assessment of the Ministry of Education, and the Distinguished Visiting Professor at the VIT, India. He presided over more than 60 research projects. He has gained six science and technology advancement medals, published more than 110 academic articles, and published three academic works. He has obtained more than 30 state patents. He is the Editor-in-Chief of *IJConvC*. He is also on the Editorial Committee of the *Journal of Computers*, *IJES*, *IJHPCN*, and *IJVICS*, and a Specialist Reviewer, such as *Telecommunications Technology*.

LINA HE was born in Handan, Hebei, China, in 1997. She received the bachelor's degree in communication engineering from Shijiazhuang University, in 2019. She is currently pursuing the master's degree with the School of Electronics and Information Engineering, Liaoning Technical University.

WEI DAI received the Ph.D. degree in safety administration from Liaoning Technical University, in 2018. She is currently a Full Lecturer at Liaoning Technical University. Her representative work has been published in high-level international/national journals/conferences, such as *China Safety Science Journal*, *China Journal of Sensors and Actuators*, *Journal of Liaoning Technical University (Natural Science)*, and *Journal of Nanoelectronics and Optoelectronics*. She has gained one science and technology advancement medal and obtained a state patent.