ITC 4/53

Information Technology and Control
Vol. 53 / No. 4/ 2024
pp. 1279-1290
DOI 10.5755/j01.itc.53.4.38367

Image Denoising Using Adaptive Weighted Low-Rank Matrix Recovery

Received 2024/08/05 | Accepted after revision 2024/11/11

HOW TO CITE: Yujuan, W., Yun, G., Ping, W. (2024). Image Denoising Using Adaptive Weighted Low-Rank Matrix Recovery. *Information Technology and Control,* 53(4), 1279-1290. https://doi.org/10.5755/j01.itc.53.4.38367

Image Denoising Using Adaptive Weighted Low-Rank Matrix Recovery

Yujuan Wang, Yun Guo, Ping Wang

Department of Electrical Engineering, Shandong Huayu University of Technology, China;
e-mails: 283265306@qq.com; 924313524@qq.com; wangping67890@126.com

Corresponding author: 283265306@qq.com

This paper introduces a new image denoising method using adaptive weighted low-rank matrix recovery to tackle the challenges of separating low-rank information from noise and improving performance affected by empirical hyperparameters. We start by using image nonlocal similarity to build a low-rank denoising model, then apply the Gerschgorin theory to precisely determine the rank of the low -rank matrix. With this rank estimation, we use adaptive weighting along with singular value decomposition and weighted soft-thresholding to solve the denoising model, resulting in the denoised image. Experiments show our algorithm surpasses traditional denoising methods in average PSNR and SSIM. Specifically, for images contaminated with high-intensity noise (with a variance of 100), our algorithm achieves average PSNR and SSIM values of 24.66dB and 0.7267, respectively. Additionally, our algorithm exhibits superior performance in denoising images with real noise and is also applicable to color image denoising.

KEYWORDS: Denoising Algorithm; Low-Rank Matrix Recovery; Image Weighted Nuclear Norm Minimization; Image Denoising; Image Processing.

1. Low-rank Matrix Recovery Model

As one of the important carriers of human information acquisition, images are often affected by various disturbing factors in the real world, leading to a decrease in the quality of image acquisition and transmission, thus affecting people's information acquisition and transmission. The goal of image denoising is to recover a clear image from a noise-polluted image. How to effectively remove image noise has always been a popular scientific research problem, in view of this, this paper proposes a new adaptive weighted low-rank image denoising algorithm [11].

When some elements of a low-rank matrix or a matrix with low-rank properties are corrupted, the method of recovering the original matrix by automatically recognizing the corrupted elements is called low-rank matrix recovery [6]. In the low-rank matrix factorization method, we wish to find a matrix X which is as close as possible to the corrupted matrix Y with a certain data fidelity, and the matrix X can be decomposed into the product of two low-rank matrices [4, 12]. However, low-rank matrix decomposition is a class of nonconvex problems that are difficult to solve. To solve this problem, we can use the rank minimization method. This is a non-convex optimization problem, but can be approximated instead of rank minimization by nuclear norm minimization (NNM) [1] for low-rank matrix recovery. Kernel-paradigm minimization is widely used in low-rank matrix recovery algorithms. To summarize, low-rank matrix recovery is a method to obtain the original matrix by identifying and recovering the corrupted matrix elements. This is able to be achieved by low-rank matrix factorization or rank minimization, of which kernel-paradigm minimization is a commonly used method. The approximate solution problem of NNM can be expressed as:

$$\hat{Y} = arg_{Y}^{min}||X - Y||_{E}^{2}/2 + \tau||Y||_{*}, \tag{1}$$

where $\| Y \|_{*} = \sum_{i=1} \sigma_i(\mathrm{L})$ (L) is the kernel paradigm of matrix Y, $\sigma_i(Y)$ is the ith singular value, and $\|\cdot\|_E$ is the E-parameter. In the image denoising task, Y and X represent the denoised image matrix and the noise image matrix, respectively. It has been proved that the NuclearNorm Proximal (NNP) problem can be solved by taking a soft thresholding operation on the singular values to obtain a closed-loop solution:

$$\hat{Y} = U\mathcal{S}_{\tau}[\Sigma]V^{T}, \tag{2}$$

where Y= $U\Sigma V^{T}$ is the singular value decomposition of the matrix Y, is a parameter-based soft thresholding operator, where the degree of sparsity is controlled. In order to minimize the rank of the matrix, minimizing all singular values simultaneously and equally is a major limitation of the method, and does not take into account the physical significance of the singular values themselves, i.e., the image information is essentially retained in the larger singular values. In order to improve the flexibility of NNM, the Weighted Kernel Paradigm Minimization (WNNM) [5] method is proposed. The weighted kernel paradigm of matrix Y is defined as:

$$||Y||_{w}, *= \Sigma_{i=1}\omega_{i}\sigma_{i}(Y), \tag{3}$$

where ω_i is the non-negative weighting assigned to $\sigma(Y)$. The weighted kernel paradigm minimization model is as follows:

$$\hat{Y} = arg_{Y}^{min}||Y - \frac{Y||_{E}^{2}}{2} + \tau\big||\mathrm{Y}|\big|_{W,*}. \tag{4}$$

Compared with NNM, weighted kernel-paradigm minimization [14, 16] greatly improves the flexibility of the kernel paradigm, which makes the low-rank matrix recovery more accurate. Therefore, the weighted kernel-paradigm minimization model achieves better denoising performance in the image denoising task. The weighted kernel-paradigm minimization model performs a weighted soft-thresholding operation on the singular values of the optimal solution obtained is $\mathrm{Y} = U\mathcal{S}_{\tau w}[\Sigma]V^{T}$, where $S_{\tau w}[\Sigma] = \mathrm{sgn}(\sigma_i)\cdot\max(|\sigma_i| - \tau\omega_i, 0)$ is the weighted soft threshold operator [7, 13].

2. Image Denoising Algorithm

2.1. Adaptive Weighted Low-rank Matrix Recovery Models

Weighted kernel-paradigm minimization models have been widely used in recent years due to their excellent performance; however, the weights in such methods depend on a regularization parameter chosen empirically and are correlated with the rank of the low-rank matrix, leading to the need to adjust the parameter iteratively in different tasks. In order to solve this problem, this paper proposes an adaptive weighting model that can adaptively weight the low rank matrix while passing through the observation data itself, and can accurately and efficiently recover the low rank matrix [17].

As mentioned earlier, the rank of the low-rank matrix is a very important parameter in the problem of recovering low-rank matrices. In some scenarios, this parameter is known, but in the vast majority of scenarios, it is unknown. In order to solve the problem that the rank of the low-rank matrix is unknown, this paper introduces the idea of Gerschgorin disk estimation to estimate the rank of the low-rank matrix. The signal N received by a sensor array in a noisy environment can be similarly expressed as the sum of the sparse noise signal matrix S [9, 10] and the low-rank source signal matrix Y [15]. The covariance matrix $_{NR}$ of matrix N of rank r can be defined as:

$$R_N = NN^T. \tag{5}$$

The eigenvalue decomposition of R_N yields:

$$R_N = U_{R_N}\Sigma_{R_N}U_{R_N}^H, \tag{6}$$

where $U_{R_N} = [u_1, u_2, \cdots, u_N]$, is a matrix consisting of eigenvectors and $\Sigma_{R_N} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ is a diagonal array consisting of eigenvalues. In a noise-free environment rank R_N is r, but in a real environment due to noise effects, rank R_N is n (n>> r). To accurately estimate the rank r of the low-rank matrix, Gerschgorin disk theory is introduced. First, the covariance matrix is divided:

$$R_N = \begin{pmatrix} R_{N1} & R \\ R^H & R_{nn} \end{pmatrix}. \tag{7}$$

$R_{N1} \in \mathbb{R}^{(n-1)\times(n-1)}$ in the above equation is obtained by deleting the last row and the last column of R_N. The eigenvalue decomposition of matrix R_{N1} can be derived:

$$R_{N1} = U_{N1}\Sigma_1 U_{N1}^H \tag{8}$$

$U_{N1} = [q_1', q_2', \cdots, q_{n-1}']$ is the matrix R_{N1} eigenvector matrix and $\mathbf{\Sigma}_1 = \left[\lambda_1', \lambda_2', \cdots, \lambda_{n-1}'\right]$ is the matrix-R_{N1} eigenvalue matrix. Define a you change matrix [5] $U \in \mathbb{R}^{n\times n}(UU^H = I)$ as follows:

$$U = \begin{pmatrix} U_{n1} & 0 \\ 0 & 1 \end{pmatrix}. \tag{9}$$

Then the covariance matrix after you change is:

$$R_T = U^H R_N U = \begin{pmatrix} \Sigma_1 & U_{N1}^H R \\ R^H U_{N1} & R_{nn} \end{pmatrix}$$
$$= \begin{pmatrix} \lambda_1 & 0 & 0 & & 0 & \rho_1 \\ 0 & \lambda_2 & 0 & \cdots & 0 & \rho_2 \\ 0 & 0 & \lambda_3 & & 0 & \rho_3 \\ & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{n-1} & \rho_{n-1} \\ \rho_1^* & \rho_2^* & \rho_3^* & & \rho_{m-1}^* & R_{mn} \end{pmatrix} \tag{10}$$

Eq. $\rho_i = q_i'^H R$. The eigenvalues of the Leigh Gerschgorin disk theory estimation matrix $_{RT}$ with the radius of the first (n- 1) Gerschgorin disks [2]:

$$r_i = |\rho_i| = |q_i'^H R|. \tag{11}$$

Therefore, the radius r_i of the it h Gerschgorin disk depends directly on the size of $q_i'^H R$. If q_i' is an eigenvector of a sparse space, the radius of the ith Gerschgorin disk, i.e., the radius of the sparse disk, will decrease significantly and tend to zero. If q_i' is an eigenvector of the low-rank space, the i-th Gerschgorin disk radius, i.e., the radius of the low-rank disk, will be non-zero and larger than the radius of the sparse disk. Therefore, the estimated rank by heuristic decision rule is:

$$GDE(k) = r_k - \frac{D(m)}{n-1}\Sigma_{i=1}^{n-1} r_i, \tag{12}$$

where k = 1, 2, n ... 2, and the adjustment factor D (m) is a constant with respect to n. The GDE (k) is computed from k = 1, 2, n ... 2. The GDE (k) is computed starting from k = 1, and when GDE (k) is negative for the first time, the rank of the low-rank matrix is r = k-1. As mentioned before, different singular values should be assigned different weights and it is desired that the weights do not need to be adjusted manually and repeatedly. Combined with the proposed low rank matrix rank estimation algorithm, this paper proposes an adaptive weighted low rank matrix recovery model based on the rank estimation, i.e:

$$\hat{Y} = arg_Y^{min} ||X - Y||_E^2/2 + \lambda||Y||_{W_Y^*}, \tag{13}$$

where, $W_Y = \mathrm{diag}\left(\{\omega_{Y,i}\}_{1\le i\le min(m,n)}\right)$. Unlike the previous methods of, this model uses a rank estimation algorithm to determine the weights and penalizes the singular values to varying degrees. In the iterative process, the low-rank matrix L, which is obtained by solving the soft-threshold operator, is expected to have its rth singular value greater than zero and its r+1st singular value less than zero. At the same time, in order to minimize the interference to the low-rank matrix, the weights are chosen with this factor in mind. Combined with the rank estimation algorithm proposed in the previous section, if the estimated rank of the low-rank matrix in the last iteration is r, the weight W_Y is set to:

$$\omega_{Y,i} = \frac{\sigma_{r+1}(X)^2}{\lambda \sigma_i(X)}. \quad (14)$$

Compared with the weighted kernel paradigm minimization algorithm [8], the model proposed in this paper is able to adaptively set the weights according to the observation matrix without manually adjusting the parameters. In addition, the model is able to ensure that the rank of the recovered low-rank matrix is equal to the target rank.

2.2. Adaptive Weighted Low-rank Matrix Recovery Image Denoising Algorithm

The detailed steps for applying the above adaptive weighted low-rank matrix recovery model to the image denoising task are as follows [16]:

(1) Divide the image X of size N × M into m image blocks xi of size Y × Y. A sliding window method can be used to partition the image into uniformly sized image blocks; (2) Search for the n image blocks that are most similar to the current image block xi in a search window of size Y × Y. The n image blocks that are most similar to the current image block xi can be searched for in the search window of size Y × Y. The n image blocks that are most similar to the current image block xi can be selected. Measures such as mean square deviation or correlation coefficient can be used to calculate the similarity and select the n most similar image blocks; (3) Stack each similar image block by column expansion into a similar block matrix X_j. Expand the pixel values of each image block by columns and then stack these column vectors into a similar block matrix; (4) Form a group of all the similar block matrices. Stack all the similar block matrices into one large matrix as the input matrix; (5) Solve the group of all similar matrices using the adaptive weighted low-rank matrix recovery model described above to recover the low-rank structure. A low-rank matrix recovery algorithm, such as the kernel-paradigm-based low-rank matrix recovery method, can be used to solve the input matrix to obtain the recovered low-rank structure; (6) After obtaining the recovered low-rank structure, convert it back to the image block form and stitch all the image blocks together to obtain the denoised image.

Through the above steps, the above adaptive weighted low-rank matrix recovery model can be applied to the image denoising task to realize the denoising process of the image. For each set of similar block matrix X_j solving then there is the following optimization problem:

$$\hat{Y}_j = arg_{Y_j}^{min} \|X_j - Y_j\|_E^2/2 + \lambda \|Y_j\|_{W_{Y,*}}. \quad (15)$$

For the minimization problem (15), the closed-form solution can be derived by the weighted soft-thresholding algorithm as:

$$\sigma_i(\hat{Y}_j) = \max(\sigma_i(X_j) - \lambda_{\omega Y,i}, 0). \quad (16)$$

From the weights given in Equation (14), combined with Eq. (16), the final singular value closed-form solution is given as:

$$\sigma_i(\hat{Y}_j) = \begin{cases} \sigma_i(X_j) - \dfrac{\sigma_{r+1}(X_j)^2}{\sigma_i(X_j)}, i \le r \\ 0, i > r \end{cases} \quad (17)$$

Equation (17) provides the solution to estimate the low rank matrix $\hat{Y}_j$ and finally the denoised image Y is obtained by combining all the group matrices $\{\hat{Y}_j\}_i^m$. In the actual denoising process, better image results can be obtained by performing multiple iterations of the above denoising process. The complete rank estimation based adaptive weighted low rank matrix recovery denoising algorithm is summarized in Algorithm 1.

Algorithm 1: Denoising algorithm based on adaptive weighted low-rank matrix recovery

Input: Noisy image Y

Output: Denoised image L^k

1. Initialize L^0 = Y, Y^0 = L
2. For k = 1 to Max-Iter
3. Iterative regularization: $Y^k=L^{k-1}+\delta(Y- L^{k-1})$
4. For each patch n in Y^k
5. Find similar patches to construct matrix Y_j
6. Estimate *r* using Eq. (12)
7. Estimate L_j using Eq. (17)
8. End loop
9. Aggregate L_j to form the denoised image L_k
10. End loop

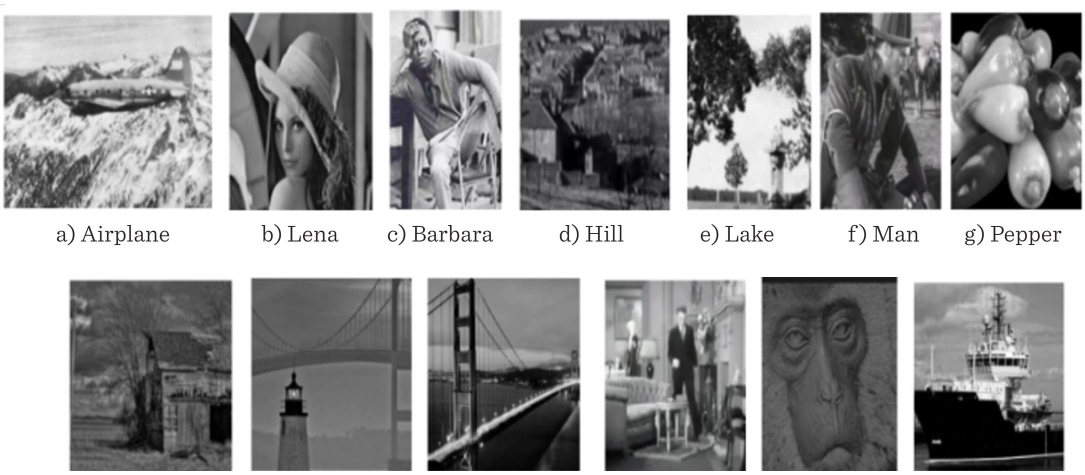
3. Experimental Results and Analysis

In order to verify the effectiveness of the adaptive weighted low-rank matrix recovery algorithm proposed in this paper on image denoising tasks, we selected images from standard image libraries and the Berkeley dataset for experimental testing. We used two metrics, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM), to quantitatively evaluate and analyze the algorithms against the same type of classical algorithms, BM3D, NNM, Weighted Kernel Paradigm Minimization, RRC, NLH and DNcnn. The codes of all compared algorithms are taken from those provided by the original authors. We first compare the performance of the adaptive weighted low-rank matrix recovery algorithm proposed in this paper with the BM3D algorithm on the denoising task. The experimental results show that our algorithm outperforms the BM3D algorithm in both PSNR and SSIM metrics. Next, we compare the algorithm proposed in this paper with the NNM algorithm. The experimental results show that our algorithm significantly outperforms the NNM algorithm in both PSNR and SSIM metrics. We also compared with classical algorithms such as weighted kernel paradigm minimization, RRC, NLH and DNcnn. The experimental results show that our algorithm achieves the best performance in both PSNR and SSIM metrics. In summary, our experimental results demonstrate the effectiveness of the adaptive weighted low-rank matrix recovery algorithm proposed in this paper on the task of image denoising. Compared with other classical algorithms, our algorithm has obvious advantages in both PSNR and SSIM metrics. The experimental parameters are set as follows, the search window L × L is 30 × 30. for the noise standard deviation. Noisy images with $\sigma n \leq 30$, $30 < \sigma\text{n} \leq 50$, $50 < \sigma\text{n} \leq 75$ and $\sigma\text{n} > 75$, the image y_i dimensions are set to 6×6, 7×7, 8×8, and 9×9, respectively, and the number of searching similar blocks m are 70, 90, 120 and 140, and the number of iterations of the algorithm is 8, 10, 12, and 14, respectively.

It is well known that the more pronounced the image noise is, the worse the training phase will make the network recognition, and to some extent the network is more demanding. Figure 1 displays the test images from a standard database. We added Gaussian noise (mean 0, standard deviations 30, 50, 75, and 100) to these images to create noisy versions. Figure 2 presents the average PSNR and

Figure 1

Test images used for comparison of denoising algorithms

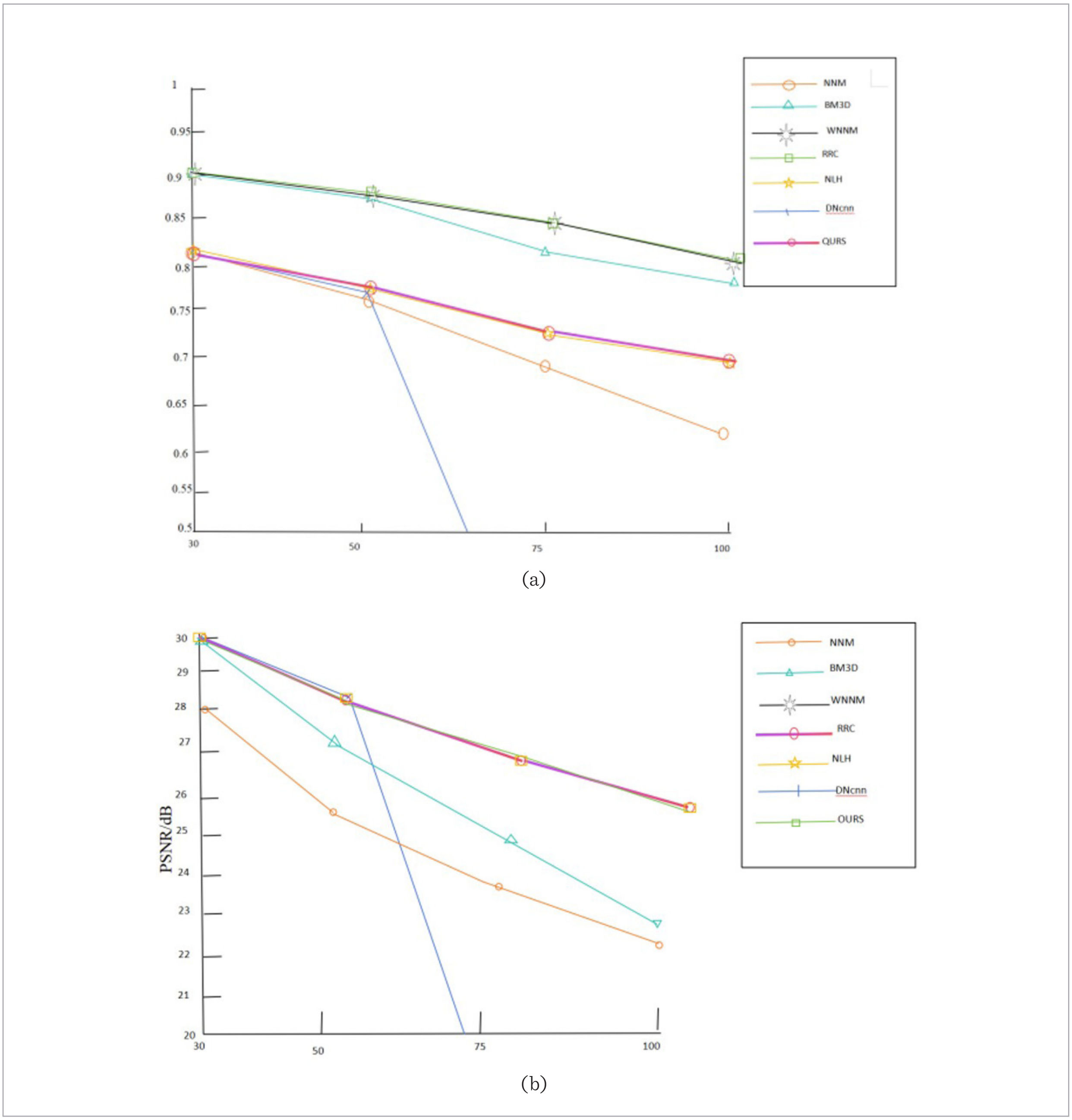


a) Airplane b) Lena c) Barbara d) Hill e) Lake f) Man g) Pepper

h) Cabin i) Beacon j) Bridge k) Couple l) Gorilla m) Boat

Figure 2

Variation curves of the mean PSNR and SSIM values obtained from various algorithms with respect to the noise variance. (a) Comparison plot of PSNR values; (b) Comparison plot of SSIM values



SSIM values for different noise levels, comparing various algorithms with the one presented in this paper. The exact PSNR and SSIM values are in Tables 1 and 2, with the best results in bold and the second best underlined.

Based on the results presented in Tables 1 and 2, the following observations can be made: The NNM algorithm exhibits non-significant performance in denoising due to its imposition of uniform penalties on singular values [3], leading to inaccurate recovery of low-rank matrices. The BM3D algorithm performs relatively averagely in denoising, with a noticeable decline in effectiveness as noise levels increase. The RRC algorithm demonstrates good denoising performance in certain scenarios, but its effectiveness also decreases significantly with increasing noise levels. This is because the algorithm requires prior estimation of the low-rank matrix, which becomes less accurate when images are corrupted by substantial noise, rendering the denoising performance of the algorithm ineffective. The weighted nuclear norm minimization algorithm performs better than other methods in denoising, but it requires adjustment of empirical parameters to adapt to different scenes. The NLH algorithm exhibits good PSNR values in most test images, but its SSIM performance decreases significantly

Table 1

Comparison of PSNR of different algorithms under different noise intensities

algorithm	σn	Lena/dB	Boat/dB	Beacon/dB	Cabin/dB	Gorilla/dB	Lake/dB	Average
NNM	30	30.25	27.92	31.33	26.87	21.37	26.26	27.33
	50	27.84	25.49	28.34	24.62	20.84	23.36	25.08
	75	26.03	23.95	25.26	22.32	19.59	20.64	22.96
	100	24.51	22.58	23.05	20.68	19.04	19.34	21.53
BM3D	30	31.36	29.22	32.83	29.23	24.76	28.27	29.28
	50	29.15	26.88	30.23	26.35	22.58	25.71	26.81
	75	27.36	25.22	26.98	23.62	21.28	23.13	24.60
	100	26.05	24.07	23.90	21.30	19.88	20.63	22.64
WNNM	30	31.53	29.34	32.92	29.64	24.85	28.44	29.45
	50	29.22	27.07	30.62	27.68	22.70	26.23	27.25
	75	27.72	25.83	29.09	26.34	21.48	24.52	25.83
	100	26.30	24.20	27.82	25.24	20.59	23.29	24.57
RRC	30	31.77	29.79	32.68	29.59	24.69	28.38	29.48
	50	29.48	27.30	30.62	27.54	22.63	26.27	27.30
	75	27.62	25.71	29.09	26.04	21.11	24.44	25.65
	100	26.41	24.60	27.82	25.02	20.27	23.20	24.55
NLH	30	31.55	29.55	32.84	29.60	24.80	28.32	29.44
	50	28.38	27.45	30.65	27.64	22.26	25.98	27.23
	75	27.73	25.85	27.22	26.34	20.96	24.42	25.42
	100	26.52	24.77	27.98	25.32	20.38	23.36	24.72
DNcnm	30	31.83	30.08	32.93	29.78	24.92	28.45	29.66
	50	29.35	27.33	30.69	27.63	22.79	26.18	27.32
	75	18.83	18.62	19.41	19.48	17.46	18.86	18.78
	100	14.05	14.12	14.47	14.71	13.74	14.44	14.25
Ours	30	31.74	29.47	32.85	29.61	24.83	28.41	29.45
	50	29.33	27.35	30.71	27.66	22.70	26.21	27.32
	75	27.74	25.83	29.24	26.30	21.49	24.60	25.87
	100	26.53	24.71	28.02	25.26	20.64	23.38	24.76

Table 2

Comparison of SSIM of different algorithms under different noise intensity

algorithm	σn	Lena	Boat	Beaco	Cabin	Gorilla	Lake	Average
NNM	30	0.8060	0.8260	0.8919	0.8024	0.6374	0.8640	0.8046
	50	0.7593	0.6988	0.8276	0.7035	0.5399	0.7552	0.7141
	75	0.7527	0.6335	0.7656	0.6343	0.4129	0.6365	0.6392
	100	0.6890	0.5984	0.7292	0.5937	0.3491	0.5859	0.5909
BM3D	30	0.9101	0.8871	0.8145	0.8610	0.8411	0.9033	0.8695
	50	0.8651	0.8176	0.8674	0.7791	0.7064	0.8409	0.8127
	75	0.8140	0.7466	0.8138	0.7026	0.5622	0.7664	0.7342
	100	0.7669	0.6903	0.7645	0.6416	0.4566	0.6962	0.6693
WNNM	30	0.9172	0.9137	0.9174	0.8813	0.8463	0.9065	0.8970
	50	0.8652	0.8494	0.8658	0.8080	0.7212	0.8490	0.8264
	75	0.8321	0.7991	0.8330	0.7652	0.6303	0.7975	0.7762
	100	0.7881	0.7519	0.7971	0.7173	0.5251	0.7437	0.7205
RRC	30	0.8749	0.8650	0.8618	0.7935	0.6877	0.7835	0.8110
	50	0.8331	0.7730	0.8255	0.7284	0.5677	0.7289	0.7427
	75	0.7922	0.7185	0.7972	0.6679	0.4261	0.6697	0.6786
	100	0.7649	0.6841	0.7769	0.6355	0.3402	0.6289	0.6384
NLH	30	0.8678	0.8398	0.8628	0.8059	0.7080	0.7841	0.8114
	50	0.8288	0.7734	0.8311	0.7377	0.5241	0.7158	0.7351
	75	0.7926	0.7241	0.7994	0.6977	0.4047	0.6664	0.6808
	100	0.7642	0.6875	0.7773	0.6519	0.3535	0.6310	0.6442
DNcnm	30	0.8770	0.8542	0.8703	0.8101	0.7371	0.7942	0.8238
	50	0.8301	0.7894	0.8289	0.7402	0.6122	0.7363	0.7562
	75	0.2151	0.3346	0.1802	0.2549	0.3346	0.2926	0.2686
	100	0.0960	0.1261	0.0739	0.1198	0.2004	0.1520	0.1280
Ours	30	0.9133	0.9051	0.9159	0.8808	0.8474	0.9042	0.8944
	50	0.8681	0.8695	0.8706	0.8152	0.7350	0.8480	0.8344
	75	0.8302	0.7983	0.8364	0.7676	0.6371	0.7952	0.7775
	100	0.7918	0.7545	0.8053	0.7235	0.5386	0.7472	0.7268

with increasing noise. Additionally, in some images, the NLH algorithm performs poorly and requires continuous adjustment of preset hyperparameters for optimization. The deep learning-based DNcnn algorithm achieves high PSNR scores when noise intensity is low. However, as noise intensity increases, both PSNR and SSIM metrics rapidly decline, falling short of other comparison methods, including the proposed algorithm in this paper. Furthermore, deep learning algorithms have a high dependence on training data, which is not a limitation of the algorithm presented in this paper. In most cases, the algorithm proposed in this paper achieves optimal evaluation metrics, and even when it does not reach the highest value, it often attains the second-best result. Additionally, as shown in Figure 2, as noise intensity increases, the performance of other algorithms declines significantly, while the proposed algorithm maintains good denoising effectiveness. At a noise level of σn=100, the PSNR of the proposed algorithm is improved by an average of 3.22dB, 2.12dB, 0.18dB, 0.21dB, 0.04dB, and 10.51dB compared to the comparison algorithms, respectively. Therefore, it can be concluded that the algorithm proposed in this paper demonstrates good denoising performance across different noise levels.

Figures 3 and 4 display the denoising outcomes for the Starfish and House images [18] from the Berkeley dataset, noisy with a variance of σn=50, using various algorithms. The zoomed boxes highlight local details. The weighted kernel paradigm minimization and RRC combined with this paper's algorithm perform well in noise removal. However, WNNM introduces detail and texture artifacts, and the weighted kernel method causes edge artifacts in the starfish and misses details in the house image. RRC mitigates these issues but over-smooths some areas, losing detail. The NLH algorithm generally denoises well but blurs some local areas. In contrast, this paper's algorithm effectively removes noise while preserving edges, textures, and other details.

In addition, we consider applying the proposed model to the task of image denoising containing real noise and verify the applicability of the algorithm of this paper to different types of noise through experiments. Since the noise level of the image in this case is unknown, we use a method to estimate the noise level of the image in our experiments by estimating the noise standard deviation σn that to determine the relevant parameters of the proposed algorithm in this paper, including the image patch size of $\sqrt{d} \times \sqrt{d}$, the number of similar patches searched (m), and the number of iterations. Figure 5 demonstrates the denoising comparison effects for two typical images with real noise (Eyes and Plate). It can be observed that the classical BM3D algorithm still leaves some residual noise during the denoising process. On the other

Figure 3

Comparison of Starfish image denoising effect

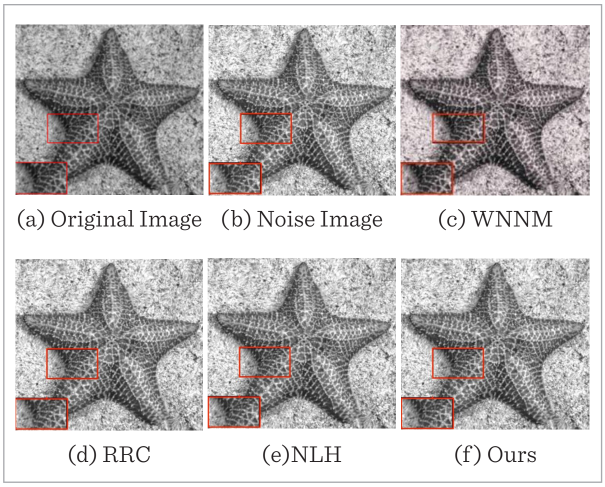


Figure 4

Comparison of House image denoising effect

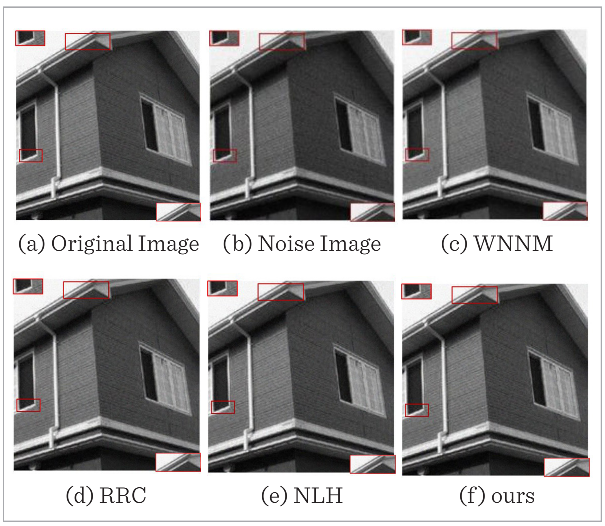
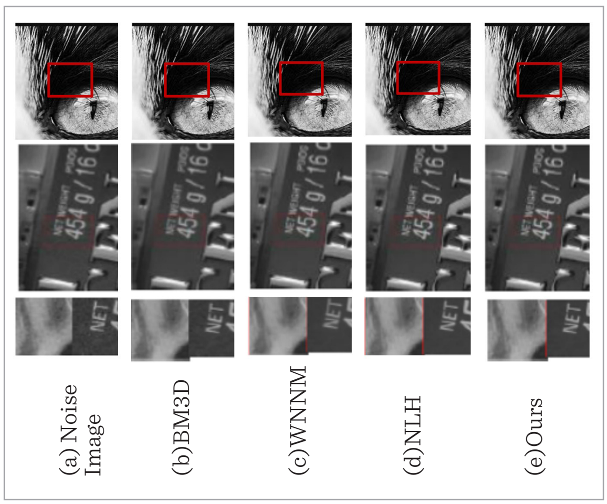


Figure 5

Comparison of denoising result images containing real noise images



hand, the WNNM algorithm produces over-smoothed results, leading to blurred images and the loss of some detailed information. In contrast, the NLH algorithm, specifically designed for denoising real noise images, exhibits good denoising performance. The proposed algorithm in this paper achieves comparable visual effects to the NLH algorithm.

To provide a more objective comparison, Table 3 lists the PSNR and SSIM metrics of the denoised images obtained by different algorithms for these two images. It can be seen that the NLH algorithm performs the best for denoising images with real noise, while the PSNR and SSIM metrics of the proposed algorithm are second only to the NLH algorithm and are quite close to it. In addition, in terms of inference time for a single image, the algorithm in this paper has the least inference time compared to the other algorithms, and the BM3D algorithm has the longest inference time. In summary, the proposed algorithm in this paper can effectively remove unknown real noise from images while preserving texture details well, achieving good visual effects.

Table 3

PSNR/SSIM comparison of denoising results for images containing real noise

Model	Eyes	Plate	Time(s)
BM3D	36.62dB/0.9765	37.80dB/0.9534	5.6
WNNM	40.38dB/0.9861	34.66dB/0.9758	4.3
NLH	40.57dB/0.9925	40.69dB/0.9912	3.6
Ours	40.48dB/0.9872	40.66dB/0.9886	3.0

The algorithm of the paper can simply be extended to color images. This involves extracting and separately denoising the red, green, and blue channels from the image, then combining the denoised channels back into a color image.

In order to further validate the complexity comparison of this paper's algorithm with others, we quantitatively compare the computation, parameter count and memory of all the algorithms and produce Table 4. As can be seen from Table 4, the overall network complexity of this paper's algorithm is lower, except in Memory compared to NLH, the rest of the evaluation indexes are the lowest and the best performance,

Table 4

Comparison of network complexity

Model	FLOPs	Params	Memory
BM3D	325.60G	153.07M	120.91M
WNNM	457.02	98.33M	269.12M
NLH	112.13G	101.39M	89.63M
Ours	109.11G	78.99M	100.20M

which further indicates that this paper's algorithm has the best performance.

In order to further verify the performance of this paper's algorithm, we will experiment this paper's algorithm on randomly selected color images of two chapters, and the results are shown in Figure 6. As can be seen from Figure 6, the left side of the image is the image with noise, and the right side is the image after denoising, this paper's algorithm in the image with a variety of noise to a certain extent can improve the clarity of the image, denoising effect is better.

Figure 6

Color image denoising analysis

4. Conclusion

This paper introduces an adaptive weighted low-rank matrix recovery algorithm for image denoising, addressing the limitations of traditional methods that struggle with weak rank constraints and poor recovery, leading to ineffective noise removal. The new algorithm adapts its weighting based on the data, effectively recovering the low-rank matrix. Tests on synthetic noisy images demonstrate that it surpasses established algorithms like NNM, BM3D, WNNM, RRC, NLH, and DNcnn. Notably, under high noise variance (100), it achieves an average PSNR of 24.66dB and SSIM of 0.7267, effectively removing noise while preserving original image details. The algorithm also performs well on real-world noisy images and is applicable to color images.

In fact, the algorithm proposed in this paper is not specifically designed for a particular type of noise, but rather relies on the image data itself and incorporates the NSS prior information of the image for denoising. Therefore, in theory, this algorithm should still achieve good denoising results when dealing with images containing a mixture of multiple types of noise. Future research can further expand experimental studies and validations. Currently, denoising algorithms based on deep learning are very popular, and they exhibit good denoising performance due to their extensive data-driven approach. However, a significant drawback of these algorithms is their excessive dependence on training data. Obtaining sufficient training data containing different types of noise is often challenging, limiting their generalization capabilities. Future research can explore combining low-rank priors with deep learning algorithms to reduce the algorithm's reliance on training data through few-shot learning and improve its generalization capabilities.

Conflict of Interest

"I hereby solemnly declare that in the course of the research for this dissertation and in the course of the research for this dissertation, I do not have any conflict of interest that may affect the impartiality of the research results."

Acknowledgement

This work is supposed by Dezhou Intelligent Equipment Research and Development Center.

References

1. Huang, C., Li, Z., Liu, Y. Quaternion-Based Weighted Nuclear Norm Minimization for Color Image Restoration. Pattern Recognition, 2022, 128, 108665. https://doi.org/10.1016/j.patcog.2022.108665
2. Ke, S., Lin, T. Sub/Super-Synchronous Oscillation Oriented Dominant Controller Parameters Stability Region Based on Gerschgorin Disk Theorem. Applied Sciences, 2021, 11(3), 1205. https://doi.org/10.3390/app11031205
3. Li, H., Liu, T., Wu, X., Chen, Q. A Bearing Fault Diagnosis Method Based on Enhanced Singular Value Decomposition. IEEE Transactions on Industrial Informatics, 2020, 17(5), 3220-3230. https://doi.org/10.1109/TII.2020.3001376
4. Li, M., Ding, D., Heldring, A., Hu, J., Chen, R., Vechhi, G. Low-Rank Matrix Factorization Method for Multiscale Simulations: A Review. IEEE Open Journal of Antennas and Propagation, 2021, 2, 286-301. https://doi.org/10.1109/OJAP.2021.3061936
5. Li, S., Li, Q., Zhu, Z. The Global Geometry of Centralized and Distributed Low-Rank Matrix Recovery Without Regularization. IEEE Signal Processing Letters, 2020, 27, 1400-1404. https://doi.org/10.1109/LSP.2020.3008876
6. Li, X., Zhu, Z., Man-Cho So, A. Nonconvex Robust Low-Rank Matrix Recovery. SIAM Journal on Optimization, 2020, 30(1), 660-686. https://doi.org/10.1137/18M1224738
7. Liu, H., Foygel Barber, R. Between Hard and Soft Thresholding: Optimal Iterative Thresholding Algorithms. Information and Inference: A Journal of the IMA, 2020, 9(4), 899-933. https://doi.org/10.1093/imaiai/iaz027
8. Liu, X. Hyperparameter-Free Localized Simple Multiple Kernel K-Means with Global Optimum. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(7), 8566-8576. https://doi.org/10.1109/TPAMI.2022.3233635
9. Ma, J., Huang, W. Sparse Low-Rank Matrix Estimation with Nonconvex Enhancement for Fault Diagnosis of Rolling Bearings. IEEE Transactions on Instrumentation and Measurement, 2023, 72, 1-10. https://doi.org/10.1109/TIM.2023.3269103
10. Macris, N., Rush, C. All-or-Nothing Statistical and Computational Phase Transitions in Sparse Spiked Matrix Estimation. Advances in Neural Information Processing Systems, 2020, 33, 14915-14926.
11. Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W., Lin, C. W. Deep Learning on Image Denoising: An Overview. Neural Networks, 2020, 131, 251-275. https://doi.org/10.1016/j.neunet.2020.07.025

12. Tong, T., Ma, C., Chi, Y. Accelerating Ill-Conditioned Low-Rank Matrix Estimation via Scaled Gradient Descent. Journal of Machine Learning Research, 2021, 22(150), 1-63.
13. Wang, Y., Deng, L. Temporal Convolutional Network with Soft Thresholding and Attention Mechanism for Machinery Prognostics. Journal of Manufacturing Systems, 2021, 60, 512-526. https://doi.org/10.1016/j.jmsy.2021.07.008
14. Wang, Y., Zheng, Y. Time-Weighted Kernel-Sparse-Representation-Based Real-Time Nonlinear Multimode Process Monitoring. IEEE Transactions on Industrial Informatics, 2021, 18(4), 2411-2421. https://doi.org/10.1109/TII.2021.3104111
15. Wu, X., Yan, J. Gridless Mixed Sources Localization Based on Low-Rank Matrix Reconstruction. IEEE Wireless Communications Letters, 2020, 9(10), 1748-1752. https://doi.org/10.1109/LWC.2020.3003446
16. Yin, S., Li, Y., Sun, B. Mixed Kernel Principal Component Weighted Regression Based on Just-in-Time Learning for Soft Sensor Modeling. Measurement Science and Technology, 2021, 33(1), 15102. https://doi.org/10.1088/1361-6501/ac2ca4
17. Yu, H., Wang, X., Wang, G., Zeng, X. An Active Three-Way Clustering Method via Low-Rank Matrices for Multi-View Data. Information Sciences, 2020, 507, 823-839. https://doi.org/10.1016/j.ins.2018.03.009
18. Zeng, Y., Guo, Y. Recognition and Extraction of High-Resolution Satellite Remote Sensing Image Buildings Based on Deep Learning. Neural Computing and Applications, 2022, 34(4), 2691-2706. https://doi.org/10.1007/s00521-021-06027-1