ATTENTIVE DEEP K-SVD NETWORK FOR PATCH CORRELATED IMAGE DENOISING

Yiwen Liang[1], *Lu Wang*[2,⋆], *Jianfei Wang*[1], *Ye Luo*[1,⋆]

[1] School of Software Engineering, Tongji University, China
[2] Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore

ABSTRACT

Techniques of dictionary learning and sparse representation are popular in recent study on image denoising, including classic K-SVD and its variants. The extension of K-SVD to its deep structure learned in an end-to-end way shows the state-of-the-art denoising performance with a great computation efficiency. However, we notice that the current learning framework takes images patches as independent samples, which ignores the inherent correlation among the patches. In this paper, we propose a deep K-SVD denoising network with attention mechanism to enhance the correlation within and among the patches. We impose the two-dimensional correlation on the intermediate parameters during the sparse representation procedure to achieve more smoothing and local-structure enhanced image features. Extensive numerical experiments using public data are conducted. The results on two datasets show that the proposed network achieves an average improvement of 0.81dB in peak signal-to-noise ratio (PSNR), 1.66% in the structural similarity (SSIM) and more than 90% in the convergence rate comparing to its counterpart, which demonstrate the efficiency and the competitive performance of our proposed network.

***Index Terms*—** Image denoising, channel and spacial attention, deep learning, sparse representation

1. INTRODUCTION

The restoration of image degraded by noise is an essential preprocessing step for various computer vision tasks. Over the decades, a mass of image denoising algorithms have been proposed to tackle the additive white Gaussian noise in image processing [2][3][17][21][13].

Decomposition denoising algorithm based on sparse theory is one of the most popular image denoising algorithms in recent years[8][14][15]. The rationale behind is that clean images can be parsimoniously represented in an over-completed dictionary. Image denosing can be achieved by retrieving only the significant presentation parts during the approximation to the noisy images. The over-completed dictionary can be fixed or a more representative one which can be learned from the images during the sparse representation [8]. K-SVD denoising algorithm [8] proposes to update the dictionary and image representation by Orthogonal Matching Pursuit (OMP) [7] in an iterative manner, where atoms in the dictionary are updated one by one by the Singular Value Decomposition (SVD) of a rank 1 matrix.

Most recently, a deep KSVD method (DKSVD) extends the original K-SVD method to a deep architecture [14], which redesigns both the steps of dictionary learning and the sparse presentation. The Learned Iterative Shrinkage and Thresholding Algorithm (LISTA)[10] is used in the sparse representation to replace OMP in order to make the dictionary learnable. Then the dictionary, i.e., a global image prior, is learned in a supervised manner. Similar to the original LISTA where an empirical scalar threshold is required to control the sparsity of the coefficients, DKSVD further proposes to incorporate a Multi-Layer Perceptron (MLP) network to adaptively learn that scalar. The proposed DKSVD connects deep neural network based solutions to the classical algorithms and achieves comparable denosing performance to other leading deep-learning based denoisers [19][20][12] with heavy computation complexities.

The main shortcoming of DKSVD is that the inherent correlation between and within the image patches are not fully considered in the denoising process. It is noted that only a scalar threshold is independently learned to control the sparsity of the sparse coefficient for each image patch in DKSVD[14]. In practice, image patches within an image share similar content or structure and their sparse representations tend to have similar patterns. The fact of correlation also applies to pixels within an image patch and the sparse coefficients at various positions commonly show additional structures, such as tree, cluster and group.

In contrast to the traditional low-rank based denoising methods where correlation is directly modeled on image patches [11][5][18], we propose to learn adaptively each sparse coefficient a sparsity controller i.e., the threshold in LISTA, in one patch and impose a two-dimensional correlation on the learned thresholds globally for all patches of an image by an attention module. The proposed new network, named as Attentive deep K-SVD (AKSVD), can selectively emphasize informative features, smooth characteristic pa-

This work was partially supported by the General Program of National Natural Science Foundation of China (NSFC) under Grant 62276189, and the Fundamental Research Funds for the Central Universities No. 22120220583.

rameters and achieve better denoising effect. Experimental results on two datasets show that the proposed method far outperforms its original counterpart DKSVD with an average improvement of 0.81dB in peak signal-to-noise ratio (PSNR), 1.66% in the structural similarity (SSIM) and more than 90% in the rate of convergence.

2. ATTENTIVE K-SVD NETWORK

Our task is to learn an optimal dictionary from the noisy image patches by considering the correlation among and within image patches to enhance the structural and contextual information in feature learning. The overall architecture of our denoising network is given in Fig. 1. The noisy input images of size $\sqrt{N} \times \sqrt{N}$ are firstly sliced into small overlapping patches of size $\sqrt{p} \times \sqrt{p}$ with $p < N$ and then fed into our proposed AKSVD network for patch based denosing. Ultimately, the denoised image is restructured by weighting the obtained denoised image patches. In the following subsections, we describe in details each part of our proposed AKSVD network.

2.1. Sparse Coding based Image Denoising Formulation

Following the architecture of DKSVD in [14], we aim to solve the following optimization problem for image denoising:

$$\left\{\hat{\mathbf{X}}, \mathbf{D}, \alpha_k\right\} = \arg\min_{\alpha_k, \mathbf{D}, \mathbf{X}} \mu\|\mathbf{X}-\mathbf{Y}\|_2^2 + \sum_k \lambda_k \|\alpha_k\|_0 + \sum_k \|\mathbf{D}\alpha_k - \mathbf{R}_k\mathbf{X}\|_2^2 , \quad (1)$$

where $\mathbf{Y}$ is the noisy image and $\mathbf{X}$ is the output denoised image in each iteration; $\hat{\mathbf{X}}$ is the denoised image; $\mathbf{D} \in \mathbb{R}^{p\times m}$ is the over-complete dictionary; $\alpha_k \in \mathbb{R}^m$ is the sparse representation vector for the k-th patch; $\mathbf{R}_k \in \mathbb{R}^{p\times N}$ is the operator for patch extraction and $\|\alpha_k\|_0$ represents the number of nonzeros in α_k.

To solve the above problem in a supervised way, the unfolded LISTA net [4, 9, 1] is used to extract the sparse coefficient α_k for each patch successively by T steps with each step following:

$$\left\{ \begin{array}{l} \hat{\alpha_k}^{t+1} = \boldsymbol{soft}_{\lambda_k/c}\left[\hat{\alpha_k}^{t} - \frac{1}{c}\mathbf{D}^{\mathrm{T}}\left(\mathbf{D}\hat{\alpha_k}^{t} - \boldsymbol{y}\right)\right] \\ \hat{\alpha_k}^{0} = 0 \end{array} \right. , \quad (2)$$

where c is the square spectral norm of $\mathbf{D}$ and

$$\boldsymbol{soft}_P(x) = \mathrm{sign}\left(x_i\right)\left(|x_i| - P\right). \quad (3)$$

It is emphasized that regularization threshold λ_k controls the sparsity of the coefficients and the level of representation error, which depends both on the noise level and the structure of the input image patch. An MLP with three hidden layers is used to learn the scalar threshold for each patch in [14].

In summary, sparse presentation of the image patches will be achieved by the unfolded LISTA network defined by (2) with a scalar threshold λ_k learned by an MLP in the forward pass and the dictionary $\mathbf{D}$ is updated according to its gradient in the backward pass.

2.2. Sparsity Controller Calculation by MLP

It is noted in (2) that only one scalar threshold λ_k is used for sparse coefficients update for the whole k-th patch. We argue that this is not sufficient to reflect the structural characteristics of the patch. We propose to extend the scalar threshold into a vector $\bar{\lambda}_k = [\lambda_{k,1}, \lambda_{k,2}, \ldots, \lambda_{k,m}]^T$ of the same size as α_k. In this way, we actually aim to solve the following reweighted l_1 minimization problem given dictionary $\mathbf{D}$:

$$\hat{\alpha_k} = \arg\min \|\mathbf{D}\alpha_k - \boldsymbol{y_k}\|_2^2 + \sum_i \lambda_{k,i}\|\alpha_{k,i}\|_1. \quad (4)$$

It is demonstrated in [6, 22] that employing weight on sparse coefficients allows for better estimation of the nonzero coefficient locations even with substantially fewer measurements.

We then propose to infer the vector $\bar{\lambda}_k$ by a MLP network from the noisy image patches according to $\bar{\lambda}_k = f_\gamma(\boldsymbol{y_k})$, where γ is the parameters of the MLP. Following [14], the MLP network has five layers, namely, the input layer, three hidden layers and output layer. The number of nodes of each layer is p^2,$8p^2$,$16p^2$,$8p^2$ and $4p^2$ respectively, where $m = 4p^2$ is the output size. The MLP function in this part learns the structural information of sparsity within the image patches. As $\bar{\lambda}_k$ is learned adaptively from the noisy image patches, it encourages different sparsity for each element in α_k.

2.3. Correlation-imposing by Channel-space Attention

Thus far, the network takes image patches as independent samples and the sparse coefficients are calculated according to (4) using LISTA independently without considering the correlation among and within the image patches. In this subsection, we incorporate a channel-space attention module between the MLP and the unfolded LISTA network to impose a two-dimensional correlation among and within the image patches.

The structure of channel-space attention module is given in Fig. 1, which is mainly composed of a channel attention block and a space attention block. Denote $\lambda \in \mathbb{R}^{q^2\times m}$ the sets of learned $\bar{\lambda}_k$ by the MLP from q^2 overlapped patches of a sub-image. We reshape λ sets to m feature images of size $q \times q$, denoted as λ^*, to establish the relationship between λ and the original image.

The correlation of sparse coefficients within patches is imposed by the channel attention block. We squeeze the spatial dimension of the input with average-pooling and max-pooling simultaneously to generate two different spatial context descriptors: λ^*_{avg1} and λ^*_{max1}. Both compressed features are fed into a shared three-layer MLP, respectively, to calculate

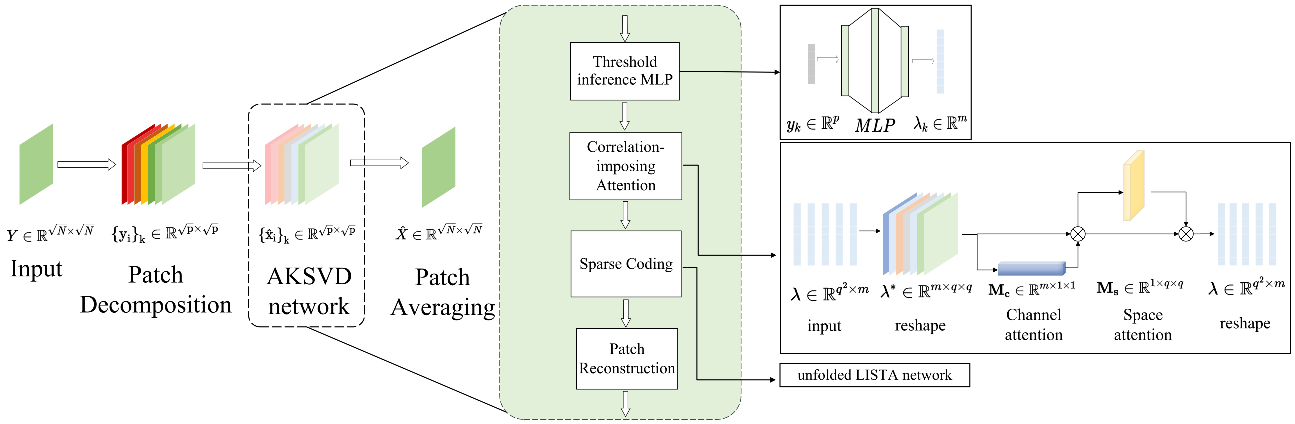
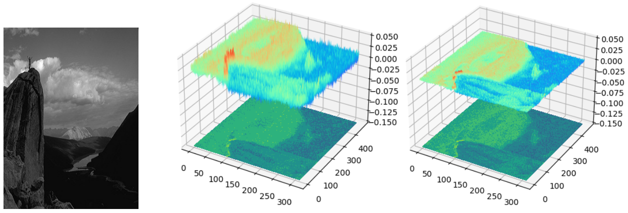

$Y \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$
Input
$\{\mathrm{y_i}\}_\mathrm{k} \in \mathbb{R}^{\sqrt{\mathrm{p}}\times\sqrt{\mathrm{p}}}$
Patch Decomposition
$\{\hat{\mathrm{x}}_\mathrm{i}\}_\mathrm{k} \in \mathbb{R}^{\sqrt{\mathrm{p}}\times\sqrt{\mathrm{p}}}$
AKSVD network
$\hat{X} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$
Patch Averaging
Threshold inference MLP
Correlation-imposing Attention
Sparse Coding
Patch Reconstruction
$y_k \in \mathbb{R}^p$
MLP
$\lambda_k \in \mathbb{R}^m$
$\lambda \in \mathbb{R}^{q^2\times m}$
input
$\lambda^* \in \mathbb{R}^{m\times q\times q}$
reshape
$\mathbf{M_c} \in \mathbb{R}^{m\times 1\times 1}$
Channel attention
$\mathbf{M_s} \in \mathbb{R}^{1\times q\times q}$
Space attention
$\lambda \in \mathbb{R}^{q^2\times m}$
reshape
unfolded LISTA network

Fig. 1: The overall architecture of image denoising using AKSVD network



(a) Original image (b) Before attention (c) After attention

Fig. 2: Heatmap of the sparsity controller λ in training.

the channel attention weights $\mathbf{M_c}$:

$$\begin{aligned}\mathbf{M_c} &= \mathrm{Sigmoid}(\mathrm{MLP}(\mathrm{AvgPool}(\lambda^*)) + \mathrm{MLP}(\mathrm{MaxPool}(\lambda^*))) \\ &= \mathrm{Sigmoid}\left(W_1\left(W_0\left(\lambda^*_{avg1}\right)\right) + W_1\left(W_0\left(\lambda^*_{max1}\right)\right)\right),\end{aligned} \tag{5}$$

where Sigmoid is the Sigmoid function, $W_0 \in \mathbb{R}^{m/r\times m}$ and $W_1 \in \mathbb{R}^{C\times C/r}$ are the MLP weights, r is the reduction ratio.

The output of channel attention is then fed as the input to the space attention, where the correlation among patches is encouraged. Similarly, through average-pooling and max-pooling in space dimension, we obtain two feature maps of size $q\times q$, which will then be concatenated and convoluted by a standard convolution layer followed by Sigmoid activation function. The spatial attention weights are finally computed as:

$$\begin{aligned}\mathbf{M_s} &= \mathrm{Sigmoid}\left(f([\mathrm{AvgPool}(\lambda^*_1); \mathrm{MaxPool}(\lambda^*_1)])\right) \\ &= \mathrm{Sigmoid}\left(f\left(\left[\lambda^*_{avg2}; \lambda^*_{max2}\right]\right)\right),\end{aligned} \tag{6}$$

where f denotes a convolution operation with the specified filter size, $\lambda^*_{avg2} \in \mathbb{R}^{1\times q\times q}$ and $\lambda^*_{max2} \in \mathbb{R}^{1\times p\times p}$ are two 2D feature maps. And the final λ^* is then updated by the output of the channel and space attention block as $\lambda^* = \mathbf{M_s}\otimes(\mathbf{M_c}\otimes\lambda^*)$, where $\otimes$ represents element-wise product. To calculate the spare coefficients, we need to reshape the correlation-imposed λ^* images into the original vectors before feeding them into the unfolded LISTA network.

To illustrate the impact of the attention module, we compare feature images of λ before and after the attention module in Fig. 2. It can be seen that after the attention module, the obtained feature is much smoother and more context information can be identified.

2.4. Patch Reconstruction

In the reconstruction module, the clean image is constructed by weighting the denoised image patches. Given the learned weight $w \in R^{\sqrt{n}\times\sqrt{n}}$ and the patch extraction operator $\mathbf{R}_k$, the clean image can be represented as:

$$\hat{\mathbf{X}} = \frac{\sum_k \mathbf{R}_k^{\mathrm{T}}\left(w \odot \hat{\boldsymbol{x}}_k\right)}{\sum_k \mathbf{R}_k^{\mathrm{T}} w}, \tag{7}$$

where $\odot$ is the Schur product and w is the weight that will be learned from the data during the network training.

3. EXPERIMENTAL RESULTS

In this section, extensive experiments are conducted using Berkeley segmentation dataset (BSDS) to verify the effectiveness of the proposed method. 432 out of 500 images are randomly selected as training data and the remaining 68 images are used for validation. White Normal-distributed noise with standard derivation of $\sigma = 15, 25, 50$ is added to the images. We employ Adaptive Moment Estimation (Adam) optimizer for network training with a learning rate of $1e-4$. All the experiments are conducted on the Ubuntu 16.04 using a desktop with an Intel Core E5-2673 CPU and four Nvidia GeForce GTX 1080 Ti GPUs. Figure 3 gives the convergence curve of training and validation losses when $\sigma = 25$. Comparing to the DKSVD in [14] where 10 hours are required for the network to fully converge, our AKSVD converges much faster and saves more than 90% of the training time.

For easy comparison with the results in [14], we evaluate the trained network on the Set12 and BSD68 data. Metrics

Table 1: Quantitive comparison between DKSVD [14] and AKSVD on Set12.

σ	Metrics	Model	Airplane	Barbara	C.man	Couple	House	Lena	Man	Monarch	Parrot	Pepper	Ship	Starfish	Avg.
15	PSNR	DKSVD	31.18	31.47	31.79	31.64	34.02	33.86	31.93	32.14	31.48	32.65	31.80	31.38	32.11
		AKSVD	**31.31**	**31.88**	**32.02**	**31.96**	**34.42**	**34.17**	**32.14**	**32.51**	**31.61**	**32.91**	**32.02**	**31.63**	**32.38**
	SSIM	DKSVD	0.90	0.90	0.86	0.86	0.88	0.89	0.87	0.93	0.90	0.90	0.85	0.91	0.89
		AKSVD	**0.90**	**0.91**	**0.90**	**0.87**	**0.88**	**0.89**	**0.88**	**0.94**	**0.90**	**0.91**	**0.85**	**0.91**	**0.90**
25	PSNR	DKSVD	26.52	27.27	27.05	27.69	29.83	29.60	28.07	27.38	26.70	28.04	27.94	26.77	27.74
		AKSVD	**28.90**	**29.45**	**29.57**	**29.62**	**32.57**	**32.03**	**29.83**	**30.21**	**29.21**	**30.46**	**29.86**	**29.19**	**30.08**
	SSIM	DKSVD	0.84	0.81	0.81	0.77	0.83	0.84	0.78	0.89	0.82	0.85	0.76	0.83	0.82
		AKSVD	**0.86**	**0.86**	**0.85**	**0.81**	**0.85**	**0.86**	**0.81**	**0.91**	**0.86**	**0.87**	**0.80**	**0.87**	**0.85**
50	PSNR	DKSVD	23.45	23.56	24.19	24.80	27.08	26.66	25.39	24.22	23.91	24.85	25.12	23.62	24.74
		AKSVD	**25.24**	**24.91**	**26.13**	**26.04**	**28.73**	**28.46**	**26.73**	**26.06**	**25.96**	**26.57**	**26.52**	**25.15**	**26.38**
	SSIM	DKSVD	0.75	0.67	0.71	0.64	0.77	0.76	0.67	0.79	0.74	0.76	0.65	0.71	0.72
		AKSVD	**0.77**	**0.70**	**0.75**	**0.67**	**0.77**	**0.77**	**0.69**	**0.81**	**0.77**	**0.78**	**0.68**	**0.75**	**0.74**

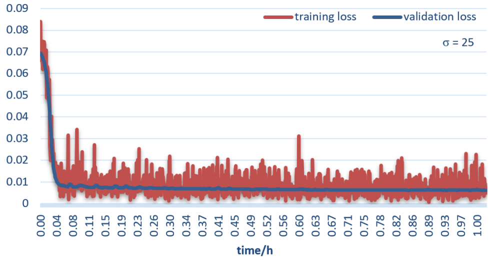


Fig. 3: Convergence of training and validation losses.

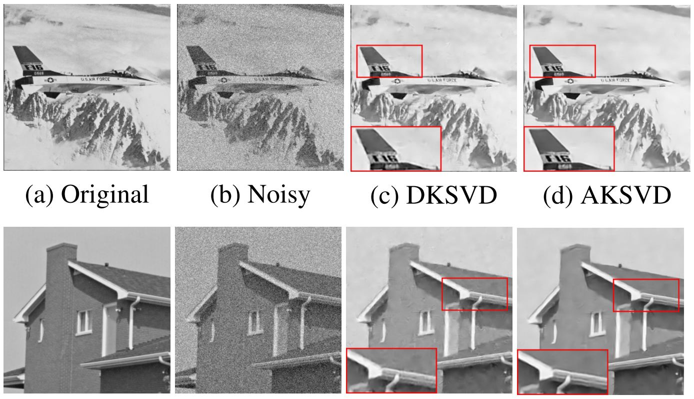
of PSNR and SSIM are used to quantitatively measure the quality of the reconstructed images. In [14], which is computationally intensive due to its reliance on a higher number of iterations, we established a more streamlined experimental framework to optimize efficiency in both time and resource utilization. We applies $T = 5$ iterations of LISTA and $K = 1$ EPLL-like denoising rounds [16] in Table 1 and Table 2. Results in both Table 1 and Table 2 show that the proposed AKSVD outperforms its counterpart DKSVD [14] for image denoising. The proposed method achieves an average improvement over DKSVD of 0.81dB in PSNR and 1.66% in SSIM. Comparing with two leading classic denoising algorithms, BM3D [3] and WNNM [11], our network has an improvement on PSNR by 0.24dB and 0.02dB on average for BSD68. Table 3 studies the impact of different structures of AKSVD when training with EPLL using Set12 dataset with $\sigma = 25$. Depending on how many round the EPLL will be used and where the attention module is incorporated, we evaluate four different structures of our AKSVD. Their denoising performances are summarized in Table 3, where AKSVD$_{t,p}$ refers that we have t EPLL iterations and the attention module is incorporated at p-th round of EPLL. It is demonstrated in Table 3 that different structures of AKSVD show slight performance differences and it achieves its best when 3 rounds are performed in EPLL with the attention module incorporated in the last round.

4. CONCLUSIONS

In this paper, we propose an attention-guided deep K-SVD network for image denoising. By extending the threshold parameter in the unfolded LISTA network, we reformulate the sparse presentation problem into a weighted l_1 minimization convex optimization. Furthermore, we employ a channel-space attention module to impose a two-dimensional correlation within and among the image patches globally on the threshold parameter to achieve more accurate parameter estimation and sparse representation. The experimental results demonstrate the effectiveness of our proposed network in perspective of reconstructed image quality and computational efficiency.

Table 2: Comparison of AKSVD and classic methods on BSD68.

σ	DKSVD		AKSVD		BM3D		WNNM	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
15	31.18	0.88	31.35	0.88	31.07	0.87	31.37	0.88
25	28.61	0.81	28.94	0.82	28.57	0.80	28.83	0.81
50	25.60	0.68	25.70	0.69	25.70	0.69	25.87	0.69


(a) Original (b) Noisy (c) DKSVD (d) AKSVD

(e) Original (f) Noisy (g) DKSVD (h) AKSVD

Fig. 4: Denoising result comparisons between DKSVD and AKSVD by example images: Airplane and House.

Table 3: Different attention structure of AKSVD versus classic methods on Set12.

Metrics	AKSVD$_{(1,1)}$	AKSVD$_{(3,1)}$	AKSVD$_{(3,2)}$	AKSVD$_{(3,3)}$	BM3D	WNNM
PSNR	30.08	30.12	30.12	30.21	29.97	30.26
SSIM	0.85	0.85	0.86	0.86	0.85	0.85

5. REFERENCES

[1] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *Siam J. Imaging Sciences*, 2(1):183–202, 2009.

[2] Brownrigg and R. K. D. The weighted median filter. *Communications of the Acm*, 27(8):807–818, 1984.

[3] Kostadin Dabov, Alessandro Foi, and Karen Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *2007 15th European Signal Processing Conference*, pages 145–149, 2007.

[4] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

[5] Weisheng Dong, Guangming Shi, and Xin Li. Nonlocal image restoration with bilateral variance estimation: A low-rank approach. *IEEE Transactions on Image Processing*, 22(2):700–711, 2013.

[6] M. B. Wakin E. J. Candes and S. P. Boyed. Enhancing sparsity by reweighted l_1 minimization. *SIAM Journal on Optimization*, 14(5-6):1065–1088, 2008.

[7] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010.

[8] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[9] M.A.T. Figueiredo and R.D. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8):906–916, 2003.

[10] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 399–406. Omnipress, 2010.

[11] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014.

[12] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.

[13] S. Liu, M. Shi, S. Hu, and X. Yang. Synthetic aperture radar image de-noising based on shearlet transform using the context-based model. *Physical Communication*, 13, 2014.

[14] Meyer Scetbon, Michael Elad, and Peyman Milanfar. Deep k-svd denoising. *IEEE Transactions on Image Processing*, 30:5944–5955, 2021.

[15] Miaowen Shi, Fan Zhang, Suwei Wang, Caiming Zhang, and Xuemei Li. Detail preserving image denoising with patch-based structure similarity via sparse representation and svd. *Computer Vision and Image Understanding*, 206:103173, 2021.

[16] Jeremias Sulam and Michael Elad. Expected patch log likelihood with a sparse prior. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 99–111. Springer, 2015.

[17] Q Xia, S Xing, D Ma, D Mo, P Li, and Z Ge. An improved k-svd-based denoising method for remote sensing satellite images. *Journal of Remote Sensing*, 20(3):441–449, 2016.

[18] Ting Xie, Shutao Li, and Bin Sun. Hyperspectral images denoising via nonconvex regularized low-rank and sparse matrix decomposition. *IEEE Transactions on Image Processing*, 29:44–56, 2020.

[19] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[20] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

[21] X. Zhang and Y. Xiong. Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter. *IEEE Signal Processing Letters*, 16(4):295–298, 2009.

[22] YUN-BIN ZHAO and DUAN LI. Reweighted l_1-minimization for sparse solutions to underdetermined linear systems. *SIAM Journal on Optimization*, 22(3):1065–1088, 2012.