Deep Content-Dependent 3-D Convolutional Sparse Coding for Hyperspectral Image Denoising

Haitao Yin, *Member, IEEE*, and Hao Chen

***Abstract*—Despite the significant successes in hyperspectral image (HSI) denoising, pure data-driven HSI denoising networks still suffer from limited understanding of inference. Deep unfolding (DU) is a feasible way to improve the interpretability of deep network. However, the specialized spatial-spectral DU methods are seldom studied, and the simple spatial-spectral extension leads to unpleasant spectral distortion. To tackle these issues, we first propose a content-dependent 3-D convolutional sparse coding (CD-CSC) to jointly represent spatial-spectral feature. Specifically, the 3-D filters used in CD-CSC for each HSI are unique, which are determined by linear combination of base 3-D filters. Then, we develop a novel CD-CSC-inspired DU network for HSI denoising, called CD-CSCNet. Furthermore, by exploiting the lightweight of separable convolution and the adaptability of hypernetwork, we design a separable content-dependent 3D Convolution (SCD-Conv) to carry out CD-CSCNet. SCD-Conv not only reduces computational complexity, but also can be viewed as the convolutional sparse coding with spatial and spectral dictionaries. Extensive experimental results on the ICVL, Zhuhai-1 OHS-3C, and GaoFen-5 datasets demonstrate that CD-CSCNet outperforms several recent pure data-driven and DU-based networks quantitatively and visually.**

***Index Terms*—Hyperspectral image denoising, deep network, convolutional sparse coding, deep unfolding, 3-D convolution, separable convolution.**

I. Introduction

HYPERSPECTRAL image (HSI) is acquired by a spectrometer with many narrow spectral bands ranging from ultraviolet to infrared wavelength, and different imaging spectral band can capture specific information of observed scenes. Due to abundant spatial–spectral descriptions, HSI has a broad range of applications in Earth observation, environmental protection, agriculture monitoring, and mineral exploration [1]. However, HSI acquisition system is influenced inevitably by atmospheric environment, imaging technology limitation, and lighting condition, which result in various noise degradations, including Gaussian noise, stripe, impluse, and deadline. These noises severely degrade the spatial-spectral structure of HSI, and cause huge challenges in image classification [2], [3], [4], [5] and target detection [6], [7], [8]. HSI denoising is an effective tool to remove noise from corrupted HSI, and can dramatically improve the image quality.

Owing to its paramount importance in various high-level vision interpretations, HSI denoising is always research hotspot. During the past few decades, numerous HSI denoising methods have been developed [9]. The existing methods typically revolve around the spatial-spectral model between noise-free HSI and noisy HSI. According to the types of spatial-spectral model, the existing HSI denoising methods can be roughly divided into the filtering-based, optimization model-based, and deep learning (DL)-based.

The filtering-based methods adopt the viewpoint that image structure and noise have different distributions, and the noise component can be separated through filtering. Benefiting from the compact representation of wavelet basis, the wavelet with thresholding is popularly used. Othman et al. [10] used the 2-D wavelet and 1-D wavelet for the spatial denoising and spectral denoising, respectively. Chen et al. [11] executed the 2-D bivariate wavelet and 1-D dual-tree complex wavelet sequentially in the low-energy part of principal component analysis domain. To exploit the 3-D structure of HSI, multidimensional filters based methods have been developed, such as the 3-D wavelet [12], multidimensional wavelet packet [13], and block-matching 4-D (BM4D) [14].

The optimization model-based methods restore the noise-free HSI through variational optimization model that derives from the theory of Bayesian maximum a posteriori estimation. The regularization term is the core of this category, which commonly corresponds to the prior assumption and statistical property of HSI. The total variation (TV) and low rank (LR) priors are two representative models. The classical TV model considers the local spatial variation of image, which can preserve spatial smoothness and edge structure. As to HSI, simple band-by-band implementation of TV may destroy spectral information. Thus, some spatio-spectral extensions of TV are developed by taking account of both spatial and spectral variations, such as the Cubic TV [15], spatio-spectral TV (SSTV) [16], anisotropic SSTV [17], l_0-l_{1-2} SSTV [18], and graph SSTV [19]. The LR models exploit the spectral correlation and the nonlocal self-similarity in HSI, which are mostly realized through LR matrix recovery (LRMR) and tensor decomposition. The popular implementations of LRMR based approaches include the matrix nuclear norm minimization [20] and the weighted Schatten p-norm [21], [22]. To avoid the calculation of singular value decomposition (SVD) in nuclear norm minimization, lots of matrix factorization-based LRMR methods [23], [24], [25] are

Manuscript received 21 November 2023; revised 2 January 2024; accepted 19 January 2024. Date of publication 23 January 2024; date of current version 8 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61971237, in part by the China Computer Federation-Baidu Open Fund under Grant 202215, and in part by the Jiangsu Qing Lan Project. *(Corresponding author: Haitao Yin.)*

The authors are with the College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing 210023, China (e-mail: haitaoyin@njupt.edu.cn; ch1263152934@163.com).

Digital Object Identifier 10.1109/JSTARS.2024.3357732

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

proposed. To jointly use the spatial low rankness and spectral low rankness, Xue et el. [26] developed the joint spatial and spectral LR based method. Alternatively, the tensor decomposition based LR methods attempt to preserve the spectral–spatial correlation using tensor operations, such as Tucker decomposition [27], [28], [29], CANDECOMP/PARAFAC decomposition [30], [31], t-SVD [32], [33], [34], tensor-ring decomposition [35], multigraph-based LR tensor approximation [36], spectral-spatial transform based sparse and LR model [37], and optimal LR tensor model [38]. In addition, multiple priors have also been suggested, such as hybrid LR and TV [39], [40], [41], [42], hyper-Laplacian regularized LR [43], and hybrid LR and sparsity [44].

Over the past decade, lots of works employ DL to solve the HSI denoising task, and achieve dominating performance. Up to now, numerous ingenious HSI denoising networks have been designed, such as plain convolutional neural network (CNN) [45], residual network [46], global reasoning network [47], partial densenet [48], attention network [49], [50], [51], recurrent network [52], [53], multiscale adaptive fusion network [54], 3-D CNN [55], and transformer [56]. In general, most HSI denoising networks are handcrafted and pure data-driven, which ignore the domain knowledge of HSI denoising and make deep network inexplicable.

Recently, deep unfolding (DU) is utilized to construct deep network targeting at an explainable architecture. The concept of DU was originally proposed in the learned iterative shrinkage-thresholding algorithm (LISTA) [57], that is, the structure of deep network strictly follows the iterative solution of optimization problem. Due to its white-box operation, LISTA has attracted continuous attentions. DU has also been extended into HSI denoising, such as the LR model with DRUNet [58], LR model with U-Net [59], nonnegative matrix factorization model with dilated deep residual network [60], Gaussian mixture model with FFDNet [61], and deep sparse coding [62]. DU provides a sound strategy to integrate the interpretability of optimization model with DL. However, the existing DU-based approaches still have several drawbacks:

1) The network parameters are shared for all HSIs, ignoring the specificity and diversity of each HSI.
2) The simple extension from 2-D deep network to its 3-D version suffers from limited flexibility of spatial-spectral representation. Intuitively, the 3-D convolution is feasible to capture spatial-spectral features. However, plain 3-D deep network extracts the coupled spatial-spectral features, which has low scalability in different spectral bands.
3) The current implementations of DU cannot balance the performance and transparency of network effectively.

To address the above issues, we first propose a novel content-dependent 3-D convolutional sparse coding (CD-CSC). Specifically, CD-CSC is the convolutional sparse coding (CSC) with dynamic 3-D convolutional dictionary. Based on CD-CSC, we then develop a meaningful DU network for HSI denoising, termed as CD-CSCNet. Besides, three handcrafted deep modules are inserted into backbone network for the low-level features extraction, dynamic weights generation, and denoised HSI reconstruction, respectively. Moreover, a separable content-dependent 3-D convolution (SCD-Conv) is developed to build CD-CSCNet. To the best of our knowledge, our CD-CSCNet is the first DU HSI denoising method to leverage the insights of 3-D CSC and separable convolution. The main contributions of this article are summarized as follows:

1) We propose the CD-CSC, which expresses a HSI through unique 3-D convolutional filters. It can enhance the adaptivity of spatial-spectral features representation.
2) We design the SCD-Conv, which consists of one content-dependent spatial convolution and one spectral convolution. The SCD-Conv not only saves computational complexity, but also decouples the spatial and spectral features which can improve the flexibility of convolution.
3) We develop the CD-CSCNet by plugging three deep modules into fully interpretable DU network. It offers a learnable optimization solution of CD-CSC, and also provides an alternative DU framework with good tradeoff between performance and transparency.

The rest of this article is organized as follows. In Section II, we briefly introduce the background. Section III describes the proposed CD-CSCNet and core components. Section IV presents the experimental comparisons and discussions. Finally, Section V concludes this article.

II. Background

A. SC and LISTA

Sparse coding (SC) has long been an effective tool to represent signal. Given an observed signal $\mathbf{x} \in \mathbb{R}^n$, SC formulates it as a sparse combination of prototype features $\mathbf{\Phi} \in \mathbb{R}^{n\times m}(n < m)$, that is, $\mathbf{x} = \mathbf{\Phi}\boldsymbol{\alpha}$. The ℓ_1-norm-penalized least absolute shrinkage and selection operator (LASSO) problem is widely used to estimate the sparse coefficient $\boldsymbol{\alpha} \in \mathbb{R}^m$

$$\min_{\mathbf{x}} \lambda\|\boldsymbol{\alpha}\|_1 + \frac{1}{2}\|\mathbf{x} - \mathbf{\Phi}\boldsymbol{\alpha}\|_2^2 \tag{1}$$

where λ is the balance parameter. The iterative shrinkage-thresholding algorithm (ISTA) is a popular choice to solve problem (1), i.e.,

$$\boldsymbol{\alpha}^{(t+1)} = \mathcal{S}_{\frac{\lambda}{L}}\left(\boldsymbol{\alpha}^{(t)} + \frac{1}{L}\mathbf{\Phi}^\top\left(\mathbf{x} - \mathbf{\Phi}\boldsymbol{\alpha}^{(t)}\right)\right) \tag{2}$$

where L is the largest eigenvalue of $\mathbf{\Phi}^\top\mathbf{\Phi}$, and $\mathcal{S}_{\frac{\lambda}{L}}(\cdot)$ is the elementwise soft-thresholding operation, defined as $\mathcal{S}_\theta(a) = \text{sign}(a)\cdot\max(0, |a| - \theta)$.

DU constructs a differentiable estimator by parameterizing the iterative solution, originating from LISTA [57]. Following the computational flow of ISTA, LISTA converts each iteration as one layer, and forms a special recurrent neural network. The tth layer of LISTA is formulated as

$$\boldsymbol{\alpha}^{(t+1)} = \mathcal{S}_{\theta^{(t)}}\left(\mathbf{\Psi}_1\boldsymbol{\alpha}^{(t)} + \mathbf{\Psi}_2\mathbf{x}\right), t = 0, 1, \ldots, T-1 \tag{3}$$

where $\mathbf{\Psi}_1 = \mathbf{I} - \frac{1}{L}\mathbf{\Phi}^\top\mathbf{\Phi}$ and $\mathbf{\Psi}_2 = \frac{1}{L}\mathbf{\Phi}^\top$. The parameters $\{\mathbf{\Psi}_1, \mathbf{\Psi}_2, \theta^{(t)}\}$ can be trained end-to-end through a task-specific supervision manner.

B. CSC

CSC is a shift-invariant SC, which uses convolution instead of matrix multiplication [63]. Formally, let $\mathbf{X} \in \mathbb{R}^{M_1 \times M_2}$ be an observed image. CSC encodes $\mathbf{X}$ as the sum of a set of filters convolved with feature maps, that is

$$\mathbf{X} = \sum_{i=1}^{N} \mathbf{f}_i * \mathbf{A}_i \tag{4}$$

where $\mathbf{f}_i \in \mathbb{R}^{n_1 \times n_2}$ is the ith filters and $\mathbf{A}_i \in \mathbb{R}^{M_1 \times M_2}$ is the ith feature map. Different from the patchwise implementation of SC, CSC is capable of handling whole image. By introducing the variables $\boldsymbol{\mathcal{F}} = \{\mathbf{f}_i\}_{i=1}^N \in \mathbb{R}^{n_1 \times n_2 \times N}$ and $\boldsymbol{\mathcal{A}} = \{\mathbf{A}_i\}_{i=1}^N \in \mathbb{R}^{M_1 \times M_2 \times N}$, we simplify (4) as $\mathbf{X} = \boldsymbol{\mathcal{F}} * \boldsymbol{\mathcal{A}}$. Then, $\boldsymbol{\mathcal{A}}$ can be sought by the convolutional LASSO

$$\min_{\boldsymbol{\mathcal{A}}} \lambda\|\boldsymbol{\mathcal{A}}\|_1 + \frac{1}{2}\|\mathbf{X} - \boldsymbol{\mathcal{F}} * \boldsymbol{\mathcal{A}}\|_F^2. \tag{5}$$

Problem (5) can be solved by the convolutional ISTA (Conv-ISTA), that is

$$\boldsymbol{\mathcal{A}}^{(t+1)} = \mathcal{S}_{\frac{\lambda}{L}}\left(\boldsymbol{\mathcal{A}}^{(t)} + \frac{1}{L}\text{flip}(\boldsymbol{\mathcal{F}}) * \left(\mathbf{X} - \boldsymbol{\mathcal{F}} * \boldsymbol{\mathcal{A}}^{(t)}\right)\right) \tag{6}$$

where flip denotes the 180° rotation.

To represent volumetric data, such as HSI, dynamic MRI, and electromagnetic brain signal, the 3-D modification of CSC (3D CSC) has been developed. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M_1 \times M_2 \times C}$ be a volumetric data with C channels. The formula of 3-D CSC can be written as

$$\boldsymbol{\mathcal{X}} = \sum_{i=1}^{N} \boldsymbol{\mathcal{F}}_i * \boldsymbol{\mathcal{A}}_i \tag{7}$$

where $\boldsymbol{\mathcal{F}}_i \in \mathbb{R}^{n_1 \times n_2 \times n_3} (n_3 < C)$ is the ith 3-D filter. The 3-D filters in (7) can capture the 3-D patterns, which has great potential in volumetric data representation.

III. Proposed Method

A. Formulation of CD-CSC

Given a noisy HSI $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{M_1 \times M_2 \times C}$, HSI denoising aims to restore the latent noise-free HSI $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M_1 \times M_2 \times C}$ from $\boldsymbol{\mathcal{Y}}$, where M_1, M_2, and C denote the height, width, and number of spectral bands, respectively. The additive noise model is widely used to define noisy HSI, i.e.,

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} \tag{8}$$

where $\boldsymbol{\mathcal{N}} \in \mathbb{R}^{M_1 \times M_2 \times C}$ denotes the corrupted noise. In this article, we propose to use the 3-D CSC to represent $\boldsymbol{\mathcal{X}}$. Hence, the noise-free HSI restoration is changed into the convolutional sparse approximation problem.

The 3-D CSC model (7) is able to capture the spatial-spectral features in HSI. However, the current model adopts a set of static 3-D filters, which are shared for all HSIs. There is a lack of understanding the specificity of different HSI. Intuitively, the unique 3-D filters could offer more compact representation than the shared 3-D filters.

Based on this viewpoint, we propose the CD-CSC, which expresses each HSI using a set of own 3-D filters. Let $\{\boldsymbol{\mathcal{B}}_j \in \mathbb{R}^{n_1 \times n_2 \times n_3} | j = 1, 2, \ldots, K\}$ be a set of base 3-D filters. Each 3-D filter $\boldsymbol{\mathcal{F}}_i$ in CD-CSC is defined as the linear combination of base 3-D filters

$$\boldsymbol{\mathcal{F}}_i = \sum_{j=1}^{K} \mathbf{w}_{i,j}\boldsymbol{\mathcal{B}}_j,\ i = 1, 2, \ldots, N \tag{9}$$

where $\{\mathbf{w}_{i,j}\}_{i,j}$ are the dynamic weights. Then, putting (9) back into (7), we obtain the formula of CD-CSC

$$\boldsymbol{\mathcal{X}} = \sum_{i=1}^{N}\left(\sum_{j=1}^{K} \mathbf{w}_{i,j}\boldsymbol{\mathcal{B}}_j\right) * \boldsymbol{\mathcal{A}}_i. \tag{10}$$

Unless otherwise specified, the symbol $*$ denotes 3-D convolution hereinafter.

CD-CSC has three significances. First, in contrast to the static filters in (7), CD-CSC employs the unique filters for each HSI due to the dynamic weights $\{\mathbf{w}_{i,j}\}_{i,j}$. Second, (9) indicates that $\{\boldsymbol{\mathcal{F}}_i\}_{i=1}^N$ can be approximated by $\{\boldsymbol{\mathcal{B}}_j\}_{j=1}^K$ with $K < N$. The convolutional dictionary size is scalable by adjusting the ratio $r = N/K$. Third, $\{\boldsymbol{\mathcal{B}}_j\}_{j=1}^K$ can be regarded as the prototype spatial-spectral features. Although $\{\boldsymbol{\mathcal{B}}_j\}_{j=1}^K$ are also shared for all HSIs, they are quite different from the fixed filters in (7). In particular, $\{\boldsymbol{\mathcal{B}}_j\}_{j=1}^K$ serve as base filters to create unique filters for representation, which can be transferred into different hyperspectral modalities.

Meanwhile, defining $\boldsymbol{\mathcal{A}} = \{\boldsymbol{\mathcal{A}}_i\}_{i=1}^N \in \mathbb{R}^{M_1 \times M_2 \times C \times N}$, $\boldsymbol{\mathcal{B}} = \{\boldsymbol{\mathcal{B}}_i\}_{i=1}^K \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times K}$, $\boldsymbol{\mathcal{F}} = \{\boldsymbol{\mathcal{F}}_i\}_{i=1}^N \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times N}$, and $\mathbf{W} \in \mathbb{R}^{N \times K}$, we then simplify (10) as

$$\boldsymbol{\mathcal{X}} = (\boldsymbol{\mathcal{B}} \times_4 \mathbf{W}) * \boldsymbol{\mathcal{A}} \tag{11}$$

where the symbol $\times_4$ denotes the 4-mode product of a tensor with a matrix. To be precise, the (i_1, i_2, i_3, j)th element of $\boldsymbol{\mathcal{B}} \times_4 \mathbf{W}$ is given by

$$(\boldsymbol{\mathcal{B}} \times_4 \mathbf{W})_{(i_1,i_2,i_3,j)} = \sum_{i_4=1}^{K} \boldsymbol{\mathcal{B}}_{(i_1,i_2,i_3,i_4)}\mathbf{W}_{(j,i_4)} \tag{12}$$

where $\boldsymbol{\mathcal{B}}_{(i_1,i_2,i_3,i_4)}$ and $\mathbf{W}_{(j,i_4)}$ are the (i_1, i_2, i_3, i_4)th element and (j, i_4)th element of $\boldsymbol{\mathcal{B}}$ and $\mathbf{W}$, respectively. Recalling the noisy HSI model (8), we reconstruct $\boldsymbol{\mathcal{A}}$ and $\mathbf{W}$ for representing $\boldsymbol{\mathcal{X}}$ through the following modified 3-D CSC optimization problem:

$$\min_{\boldsymbol{\mathcal{A}}, \mathbf{W}} \frac{1}{2}\|\boldsymbol{\mathcal{Y}} - (\boldsymbol{\mathcal{B}} \times_4 \mathbf{W}) * \boldsymbol{\mathcal{A}}\|_F^2 + \lambda_1\|\boldsymbol{\mathcal{A}}\|_1 + \lambda_2 g(\mathbf{W}) \tag{13}$$

where $g(\cdot)$ denotes the constraint on $\mathbf{W}$.

In general, $g(\cdot)$ is defined as a convex function, such as the ℓ_1-norm and ℓ_2-norm. With accurate formula of $g(\cdot)$, (13) can be solved through the alternating direction method of multipliers (ADMM). The basic procedure of ADMM is that $\boldsymbol{\mathcal{A}}$ and $\mathbf{W}$ are alternatively optimized with another one fixed, and iterated until the stopping criterion. However, the ADMM solver has several

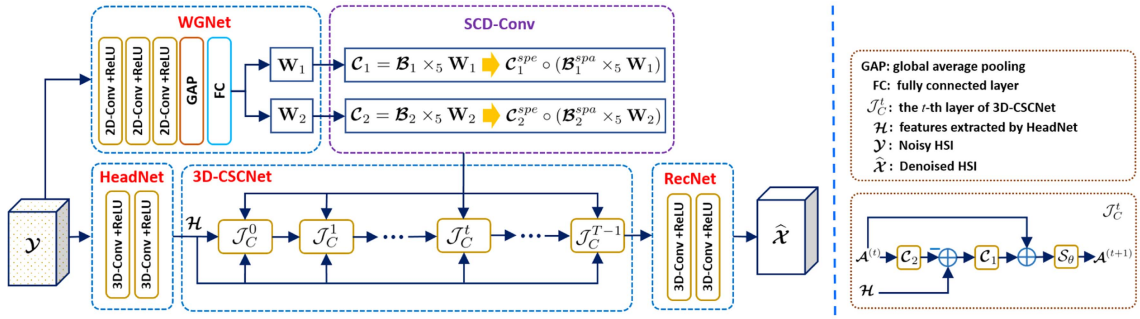


Fig. 1. Overall architecture of proposed CD-CSCNet.

disadvantages in slow convergence speed, intractable multiple parameters adjustment, and sensitivity to noise. Furthermore, it is still a challenge to define an universal explicit $g(\cdot)$ due to the high-dimension structure of HSI.

B. Architecture of CD-CSCNet

To avoid the disadvantages of optimization algorithm to solve problem (13), we propose a CD-CSC-inspired DU network, called CD-CSCNet. In particular, we use a separate subnetwork to predict $\mathbf{W}$ instead of explicit function $g(\cdot)$. Then, the problem of $\boldsymbol{\mathcal{A}}$ prediction is transformed as a standard 3-D CSC, which can be solved through the deep 3-D CSC network (3D-CSCNet) using the DU strategy. Fig. 1 shows the overall architecture of CD-CSCNet, which contains head feature extraction network (HeadNet), 3D-CSCNet, weight generation network (WGNet), and reconstruction network (RecNet).

1) HeadNet: To enhance the effectiveness and accuracy of spatial-spectral representation, we first employ the HeadNet to extract low-level features from input noisy HSI $\boldsymbol{\mathcal{Y}}$. Specifically, the HeadNet is composed of two $3 \times 3 \times 3$ convolutional layers, and each of which applies the ReLU function. Let $\boldsymbol{\mathcal{H}}$ be the extracted features by HeadNet. Without loss of generality, the size of $\boldsymbol{\mathcal{H}}$ is still denoted as $M_1 \times M_2 \times C$. In the following experiments, the channels of two 3-D convolutioanl layers in HeadNet are all set to 16.

2) 3D-CSCNet: 3D-CSCNet aims to calculate the sparse feature maps $\boldsymbol{\mathcal{A}}$ from $\boldsymbol{\mathcal{H}}$. We assume for a moment that the 3-D filters in CD-CSC are given, and defined as $\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{B}} \times_4 \mathbf{W}$. Then, the optimization problem for calculating $\boldsymbol{\mathcal{A}}$ is

$$\min_{\boldsymbol{\mathcal{A}}} \lambda \|\boldsymbol{\mathcal{A}}\|_1 + \frac{1}{2}\|\boldsymbol{\mathcal{H}} - \boldsymbol{\mathcal{F}} * \boldsymbol{\mathcal{A}}\|_F^2. \tag{14}$$

The iterative solution of problem (14) is

$$\boldsymbol{\mathcal{A}}^{(t+1)} = \mathcal{S}_{\frac{\lambda}{L}}\left(\boldsymbol{\mathcal{A}}^{(t)} + \frac{1}{L}\text{flip}(\boldsymbol{\mathcal{F}}) * \left(\boldsymbol{\mathcal{H}} - \boldsymbol{\mathcal{F}} * \boldsymbol{\mathcal{A}}^{(t)}\right)\right). \tag{15}$$

Using DU, we reformulate (15) as a feed-forward 3-D CNN with T layers, namely 3D-CSCNet. The tth layer $\mathcal{J}_C^t$ $(t = 0, 1, \ldots, T-1)$ is defined as

$$\boldsymbol{\mathcal{A}}^{(t+1)} = \mathcal{S}_\theta\left(\boldsymbol{\mathcal{A}}^{(t)} + \boldsymbol{\mathcal{C}}_1 * \left(\boldsymbol{\mathcal{H}} - \boldsymbol{\mathcal{C}}_2 * \boldsymbol{\mathcal{A}}^{(t)}\right)\right) \tag{16}$$

where θ is a learnable threshold parameter for $\mathcal{S}_\theta(\cdot)$, $\boldsymbol{\mathcal{C}}_1 \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times C \times N}$ and $\boldsymbol{\mathcal{C}}_2 \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times N \times C}$ are the 3-D convolutional layers for encoding and decoding, respectively. The dimension in $\boldsymbol{\mathcal{C}}_1$ ($\boldsymbol{\mathcal{C}}_2$) is denoted as "height $\times$ width $\times$ depth $\times$ in channels $\times$ out channels." Comparing with (15) requiring hundreds of iterations, T in 3D-CSCNet is much smaller, which is set to 10 in our experiments.

3) WGNet: The goal of WGNet is to predict the weights of convolutional kernels ($\boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{\mathcal{C}}_2$) in (16) adaptively. As we have seen, $\boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{\mathcal{C}}_2$ are the parameterized $\frac{1}{L}\text{flip}(\boldsymbol{\mathcal{F}})$ and $\boldsymbol{\mathcal{F}}$, respectively. Since $\boldsymbol{\mathcal{F}}$ is content-dependent, we also define $\boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{\mathcal{C}}_2$ using the concept of content dependent. Thus, the related convolution operation is called as the content-dependent 3-D convolution (CD-Conv). Let $\{\boldsymbol{\mathcal{B}}_1, \boldsymbol{\mathcal{B}}_2\}$ and $\{\mathbf{W}_1, \mathbf{W}_2\}$ be the base convolutional kernels and the dynamic weights, respectively. Based on the concept of CD-Conv, $\boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{\mathcal{C}}_2$ are defined as

$$\boldsymbol{\mathcal{C}}_1 = \boldsymbol{\mathcal{B}}_1 \times_5 \mathbf{W}_1,\ \boldsymbol{\mathcal{C}}_2 = \boldsymbol{\mathcal{B}}_2 \times_5 \mathbf{W}_2 \tag{17}$$

where $\times_5$ denotes the 5-mode product similar to (12). Section II-I-C will present a separable refinement of $\boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{\mathcal{C}}_2$ with more details.

To avoid the choice of $g(\cdot)$ in problem (13), we resort to the implicit WGNet to predict $\mathbf{W}_1$ and $\mathbf{W}_2$ for each input $\boldsymbol{\mathcal{Y}}$ adaptively. As shown in Fig. 1, WGNet consists of three 3×3 convolutional layers with ReLU, one global average pooling (GAP) layer, and one fully connected (FC) layer. Specifically, the channels of three convolutional layers are set to 32, 64, and 128 channels, respectively. The output of FC layer is with the size of $2NK$, which is then reshaped as two $N \times K$ matrices, namely $\mathbf{W}_1$ and $\mathbf{W}_2$.

4) RecNet: RecNet reconstructs the final denoised HSI $\widehat{\boldsymbol{\mathcal{X}}}$ from the sparse features $\boldsymbol{\mathcal{A}}^{(T)}$ extracted by 3D-CSCNet, that is, $\widehat{\boldsymbol{\mathcal{X}}} = \text{RecNet}(\boldsymbol{\mathcal{A}}^{(T)})$. RecNet contains two $3 \times 3 \times 3$ convolutional layers with ReLU.

C. Separable Implementation of CD-Conv

The key element of CD-CSCNet is the CD-Conv ($\boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{\mathcal{C}}_2$) in (16). The naive CD-Conv is based on typical 3-D convolution, which requires more expensive computational budget and memory than 2-D convolution, especially for the HSI with both large spatial size and lots of spectral bands. We adopt

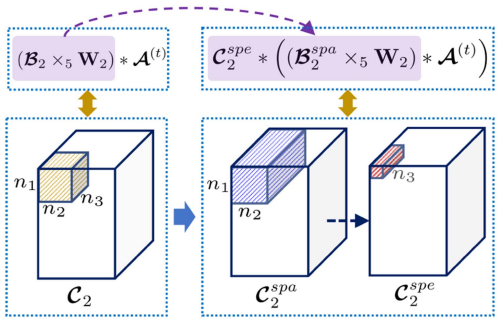


Fig. 2. Illustrations of CD-Conv and SCD-Conv about $\mathcal{C}_2$ in (16).

the separable 3-D convolution proposal to refine CD-Conv, and develop a novel separable CD-Conv (SCD-Conv). The definition of SCD-Conv considers two fundamental aspects: *separable* and *content-dependent*. We take the decoder $\mathcal{C}_2$ as example to describe SCD-Conv in details.

In the aspect of *separable*, SCD-Conv decouples each 3-D convolution into one spatial-convolution and one spectral-convolution. Formally, the $n_1 \times n_2 \times n_3$ 3-D convolution is decomposed into one $n_1 \times n_2 \times 1$ spatial-convolution ($\mathcal{C}_2^{spa}$) and one $1 \times 1 \times n_3$ spectral-convolution ($\mathcal{C}_2^{spe}$). Then, $\mathcal{C}_2$ is approximated by cascading $\mathcal{C}_2^{spa}$ and $\mathcal{C}_2^{spe}$. Thus, the term $\mathcal{C}_2 * \mathcal{A}^{(t)}$ in (16) can be rewritten as

$$\mathcal{C}_2 * \mathcal{A}^{(t)} = \mathcal{C}_2^{spe} * \left(\mathcal{C}_2^{spa} * \mathcal{A}^{(t)}\right). \tag{18}$$

In the aspect of *content-dependent*, a straightforward approach is to adopt two dynamic networks to generate the kernels of $\mathcal{C}_2^{spa}$ and $\mathcal{C}_2^{spe}$. The diversity of spectral feature is less than spatial feature, and it is unnecessary to specialize $\mathcal{C}_2^{spe}$. Hence, we propose to apply the concept of content-dependent to create $\mathcal{C}_2^{spa}$, while $\mathcal{C}_2^{spe}$ is shared for all HSIs. Let $\mathcal{B}_2^{spa} \in \mathbb{R}^{n_1 \times n_2 \times 1 \times N \times K}$ and $\mathbf{W}_2 \in \mathbb{R}^{N \times K}$ be the base part of spatial-convolution and the dynamic weights, respectively. Then, $\mathcal{C}_2^{spa}$ is defined as

$$\mathcal{C}_2^{spa} = \mathcal{B}_2^{spa} \times_5 \mathbf{W}_2. \tag{19}$$

Replacing $\mathcal{C}_2^{spa}$ in (18) with (19), we obtain the implementation of SCD-Conv on $\mathcal{C}_2$

$$\mathcal{C}_2 * \mathcal{A}^{(t)} = \mathcal{C}_2^{spe} * \left((\mathcal{B}_2^{spa} \times_5 \mathbf{W}_2) * \mathcal{A}^{(t)}\right) \tag{20}$$

where $\mathcal{C}_2^{spe} \in \mathbb{R}^{1 \times 1 \times n_3 \times N \times C}$. Fig. 2 illustrates the comparison between CD-Conv and SCD-Conv.

SCD-Conv has three advantages in terms of model-interpretable, content-dependent, and complexity reduction:

- *Model-Interpretable:* The separable convolution in (18) is interpreted as double dictionaries 3-D CSC. Specifically, $\mathcal{C}_2^{spa}$ and $\mathcal{C}_2^{spe}$ can be regarded as the spatial and spectral convolutional dictionaries, respectively.
- *Content-Dependent:* $\mathcal{C}_2^{spa}$ is determined through the adaptively predicted $\mathbf{W}_2$, which has great ability at representing own spatial content of each HSI.
- *Complexity Reduction:* The computational complexity of CD-Conv for $\mathcal{C}_2$ is $O(n_1 n_2 n_3 CN + NK)$, which is then reduced to $O(n_1 n_2 CN + n_3 CN + NK)$ in SCD-Conv.

In the same way, the encoder $\mathcal{C}_1$ in (16) is also carried out using SCD-Conv. Let $\mathcal{C}_1^{spa}$ and $\mathcal{C}_1^{spe}$ be the spatial-convolution and spectral-convolution, respectively. Based on the concept of content-dependent, $\mathcal{C}_1^{spa}$ is defined as $\mathcal{C}_1^{spa} = \mathcal{B}_1^{spa} \times_5 \mathbf{W}_1$. Analogous to (20), the implementation of SCD-Conv on $\mathcal{C}_1$ is formulated as

$$\mathcal{E}^{(t)} = \mathcal{H} - \mathcal{C}_2 * \mathcal{A}^{(t)}, \mathcal{C}_1 * \mathcal{E}^{(t)} = \mathcal{C}_1^{spe} * \left(\mathcal{C}_1^{spa} * \mathcal{E}^{(t)}\right). \tag{21}$$

To this end, incorporating (20) and (21) into (16), we obtain the computational procedures of $\mathcal{J}_C^t$

$$\begin{cases} \mathcal{C}_1^{spa} = \mathcal{B}_1^{spa} \times_5 \mathbf{W}_1, \mathcal{C}_2^{spa} = \mathcal{B}_2^{spa} \times_5 \mathbf{W}_2 \\ \mathcal{C}_2 * \mathcal{A}^{(t)} = \mathcal{C}_2^{spe} * (\mathcal{C}_2^{spa} * \mathcal{A}^{(t)}) \\ \mathcal{E}^{(t)} = \mathcal{H} - \mathcal{C}_2 * \mathcal{A}^{(t)}, \mathcal{C}_1 * \mathcal{E}^{(t)} = \mathcal{C}_1^{spe} * \left(\mathcal{C}_1^{spa} * \mathcal{E}^{(t)}\right) \\ \mathcal{A}^{(t+1)} = \mathcal{S}_\theta\left(\mathcal{A}^{(t)} + \mathcal{C}_1 * \mathcal{E}^{(t)}\right). \end{cases} \tag{22}$$

To visually show the superiority of SCD-Conv, we visualize the feature maps $\mathcal{A}^{(t)}$ at different layers, extracted by the CD-CSCNet with normal 3-D Conv, CD-Conv, and SCD-Conv, respectively. A noisy HSI [see Fig. 3(b)] is simulated from clean HSI [see Fig. 3(a)] by adding Gaussian noise, and then it is fed into the pretrained CD-CSCNet. The obtained feature maps $\mathcal{A}_i^{(t)} \in \mathbb{R}^{M_1 \times M_2 \times C}$ are processed by $\text{mean}(\text{abs}(\mathcal{A}_i^{(t)}), 3)$, where $\text{abs}(\cdot)$ and $\text{mean}(\cdot, 3)$ compute the absolute value and the mean value in the 3-D, respectively. From Fig. 3, we can obtain the following findings:

1) The features extracted the normal 3-D Conv without content-dependent are not sparse sufficiently, as shown in Fig. 3(c)–(e). In contrast, the content-dependent convolutions (CD-Conv and SCD-Conv) can provide more sparse and sharp features.
2) The comparisons of Fig. 3(f)–(h) and the comparisons of Fig. 3(i)–(k) show that the features become more sparse and more sharp with the layer t increasing.
3) Comparing with CD-Conv [see Fig. 3(f)–(h)], the separable approach (SCD-Conv) [see Fig. 3(i)–(k)] can capture the salient features more completely.

D. Training Strategy and Implement Details

The mean square error loss function is used, which is defined as

$$\mathcal{L}(\mathbf{\Theta}) = \sum \|\text{CD-CSCNet}(\mathcal{Y}_k) - \mathcal{X}_k\|_F^2 \tag{23}$$

where $\{\mathcal{X}_k, \mathcal{Y}_k\}$ are the kth pair of clean-noisy HSIs, and $\mathbf{\Theta}$ denotes all trainable parameters.

We use the ADAM optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for training network parameters. The batch size and epoch are set as to 16 and 100, respectively. The learning rate is initialized as 2×10^{-4}, and descended by a factor of 0.1 at 70 and 90 epoches, respectively. CD-CSCNet is executed using PyTorch and trained on a NVIDIA GeForce RTX 3090 GPU.

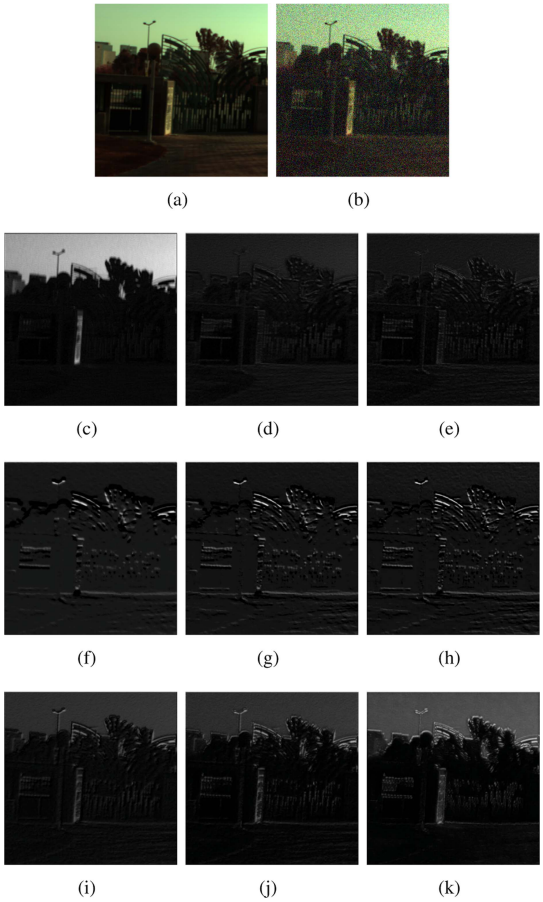


Fig. 3. Visualization of feature maps $\mathcal{A}^{(t)}$ obtained by different convolutions. (a) Clean HSI (composite of bands 31, 11, and 6). (b) Noisy HSI. (c) Normal 3-D Conv ($t = 1$). (d) Normal 3-D Conv ($t = 3$). (e) Normal 3-D Conv ($t = 5$). (f) CD-Conv ($t = 1$). (g) CD-Conv ($t = 3$). (h) CD-Conv ($t = 5$). (i) SCD-Conv ($t = 1$). (j) SCD-Conv ($t = 3$). (k) SCD-Conv ($t = 5$).

IV. Experiments

A. Experimental Settings

1) Training Dataset: The ICVL[1] dataset is used to build training set. Specifically, the ICVL dataset contains 201 real-world objects, and each image is with size of $1392 \times 1300 \times 31$. We first select 100 images. A total of 30 000 patches with size of $50 \times 50 \times 31$ are randomly extracted from these 100 images, which are then used as training set. Moreover, all images are normalized into [0,1].

2) Simulated Noise Settings: The performance assessment is conducted on the simulated noise and real noise. For a fair performance comparison, we follow the simulated noise settings in [47], defined as:

- *Case 1* *i.i.d. Gaussian noise:* All bands are corrupted by additive Gaussian noise with the same standard deviation $\sigma = \{30, 50, 70\}$.
- *Case 2* *Non-i.i.d. Gaussian noise:* Each band is corrupted by additive Gaussian noise with different standard deviation σ, randomly drawn from the interval [30,70].
- *Case 3* *Gaussian noise + Stripe noise:* All bands are corrupted by the non-i.i.d. Gaussian noise. Then, 30% of bands are disturbed by additive stripes. The number of stripes is $5\% - 15\%$ of columns.
- *Case 4* *Gaussian noise + Deadline noise:* The non-i.i.d. Gaussian noise is added into all bands as Case 2. Then, $5\% - 15\%$ of columns in the randomly selected 30% of bands are affected by the deadlines.
- *Case 5* *Gaussian noise + Impluse noise:* We first add the Non-i.i.d. Gaussian noise into all bands, and then the impulse noise with the percentage of $50\% - 70\%$ is applied to randomly selected 30% of bands.
- *Case 6* *Mixture noise:* At least one type of noise mentioned above is randomly selected and added to each band.

Moreover, we train a specific network for each simulated noise case, and then assess performance using the corresponding pre-trained network.

3) Compared Methods: We compare our CD-CSCNet with several popular HSI denoising methods selected from different categories, including one filtering-based method (BM4D[2] [14]), six optimization model-based methods (LRTV[3] [39], LRTDTV[4] [41], TDL[5] [28], LLRT[6] [43], NMoG[7] [25], and IT-SReg[8] [27]), four pure data-driven DL-based methods (HDCNN[9] [46], QRNN[10] [55], SQAD[11] [52], and GRN[12] [47]), and two DU-based methods (FastHyMix[13] [61] and T3SC[14] [62]). All compared methods are reproduced with default settings. According to the ablation studies in Section IV-E, the major hyper-parameters in CD-CSCNet are set as $T = 10$, $N = 64$, and $r = 4$.

4) Evaluation Indexes: For quantitative comparisons, we adopt three indexes, including the peak signal-to-noise ratio (PSNR), structure similarity (SSIM), and spectral angle mapper (SAM), which can quantify the visual quality, structure similarity, and spectral fidelity, respectively. Larger PSNR and SSIM values, and smaller value of SAM indicates the better denoised result. Furthermore, the average inference time and the number of model parameters are adopted for computational complexity comparison.

B. Simulated Noise Experiment on ICVL Dataset

In this section, we perform a simulated noise experiment (Cases 1–6) on the ICVL dataset. Except the images used in

[1][Online]. Available: http://icvl.cs.bgu.ac.il/hyperspectral/

[2][Online]. Available: http://www.cs.tut.fi/foi/

[3][Online]. Available: http://www.lmars.whu.edu.cn/prof_web/zhanghongyan/Homepage.html

[4][Online]. Available: http://gr.xjtu.edu.cn/en/web/dymeng/3

[5][Online]. Available: http://gr.xjtu.edu.cn/web/dymeng/

[6][Online]. Available: https://owuchangyuo.github.io/

[7][Online]. Available: https://github.com/xiangyongcao/NMoG_RPCA

[8][Online]. Available: http://gr.xjtu.edu.cn/en/web/dymeng/3

[9][Online]. Available: https://github.com/qzhang95/HSID-CNN

[10][Online]. Available: https://github.com/Vandermode/QRNN3D

[11][Online]. Available: https://github.com/EtPan/SQAD

[12][Online]. Available: https://github.com/xiangyongcao/GRN

[13][Online]. Available: https://github.com/LinaZhuang/HSI-MixedNoise Removal-FastHyMix

[14][Online]. Available: https://github.com/inria-thoth/T3SC/tree/main

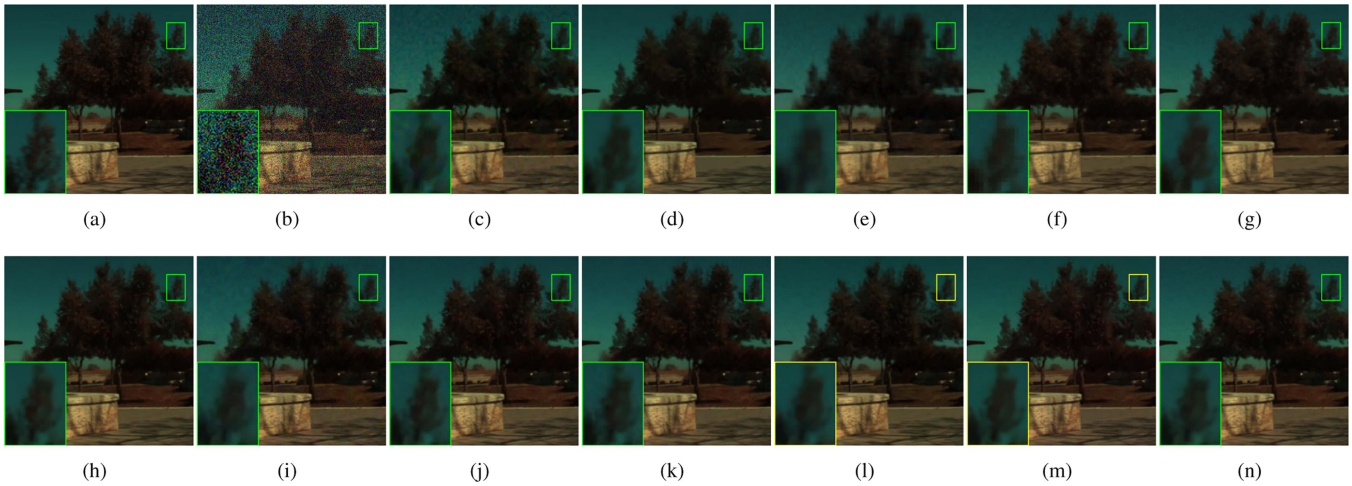


Fig. 4. Denoised results of different methods on the ICVL dataset for Case 1 at $\sigma = 50$ (composite of bands 31, 11, and 6). (a) Ground truth. (b) Noisy HSI. (c) BM4D. (d) LLRT. (e) LRTDTV. (f) TDL. (g) ITSReg. (h) GRN. (i) HDCNN. (j) QRNN. (k) SQAD. (l) FastHyMix. (m) T3SC. (n) Ours.

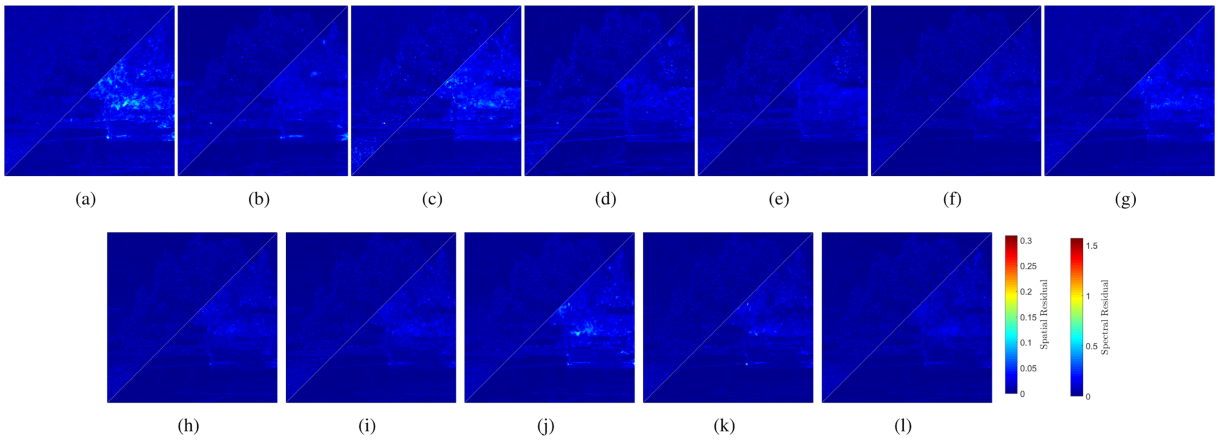


Fig. 5. Residual maps of different methods on the ICVL dataset for Case 1 at $\sigma = 50$ (left: spatial residual map, right: spectral residual map). (a) BM4D. (b) LLRT. (c) LRTDTV. (d) TDL. (e) ITSReg. (f) GRN. (g) HDCNN. (h) QRNN. (i) SQAD. (j) FastHyMix. (k) T3SC. (l) Ours.

training set, we select other 50 images for testing. Then, each image is centrally cropped into $512 \times 512 \times 31$.

Fig. 4 illustrates the visual comparison of different methods on "*prk_0328-0945*" image for Case 1 at noise level $\sigma = 50$. In addition, a closeup on tree branch of each image is shown synchronously. Compared with other methods, the results of BM4D [see Fig. 4(c)] still have noticeable noise, and obtain a little poor denoising results. LRTDTV, TDL, ITSReg, and HDCNN generate some blurring and cause some spatial details distortions, as shown in Fig. 4(e)–(g), and (i), respectively. The closeup of Fig. 4(k) shows that SQAD introduces some artifacts. In contrast, LLRT [see Fig. 4(d)], GRN [see Fig. 4(h)], QRNN [see Fig. 4(j)], FastHyMix [see Fig. 4(l)], T3SC [see Fig. 4(m)], and our method [see Fig. 4(n)] exhibit better spatial visual effects. Furthermore, Fig. 5 displays the spatial residual and spectral residual, as shown in the left part and right part of each image, respectively. Specifically, the absolute value of bandwise spatial residual is calculated band-by-band, and then the average spatial residual value among all bands is illustrated. Meanwhile, the spectral residual is computed from the SAM index. Fig. 5 shows that our method obtains the fewest spatial and spectral distortions.

Tables I and II present the average indexes results on the ICVL dataset with Case 1 and Cases 2–6, respectively. It can be observed that the DL-based methods achieve better results than the filtering- and optimization model-based methods. For Case 1, T3SC is slightly better than our method at low noise level, but our method provides the best results at high noise level. Moreover, our method obtains the best PSNR, SSIM, and SAM values at Cases 2-6. In particular, our method surpasses the second best PSNR method by 0.16 dB for Case 1 at $\sigma = 70$ and even 1.65 dB for Case 6. From the SAM index, our method reduces the spectral distortion about 10.77% and 16.47% over the second best for Case 1 at $\sigma = 70$ and Case 6, respectively. The indexes values verify the superiority of our method on the spatial and spectral preservations quantitatively. In addition, Fig. 6 depicts the PSNR and SSIM of different methods for each band in Case 1 at noise level $\sigma = 50$. From Fig. 6, it can be found that our method obtains the best PSNR and SSIM values for most of bands.

TABLE I
INDEXES RESULTS OF ALL METHODS ON THE ICVL DATASET WITH CASES 1 AT DIFFERENT NOISE LEVELS

σ	Indexes	Noisy	BM4D	TDL	ITSReg	LLRT	LRTV	NMoG	LRTDTV	HDCNN	QRNN	SQAD	GRN	FastHyMix	T3SC	Ours
30	PSNR	18.59	38.11	40.59	41.31	41.62	34.32	32.81	38.13	40.43	42.27	42.01	42.13	41.78	**43.35**	43.02
	SSIM	0.106	0.937	0.958	0.962	0.968	0.902	0.738	0.939	0.955	0.971	0.969	0.974	0.967	**0.977**	0.976
	SAM	0.702	0.103	0.056	0.071	0.052	0.068	0.141	0.077	0.064	0.059	0.063	0.058	0.064	0.053	**0.044**
50	PSNR	14.15	35.34	38.19	38.96	38.36	32.51	29.54	36.14	38.68	40.01	39.52	40.06	38.93	**40.85**	40.73
	SSIM	0.044	0.895	0.936	0.944	0.941	0.881	0.578	0.918	0.939	0.958	0.955	0.962	0.944	0.963	**0.964**
	SAM	0.887	0.136	0.071	0.076	0.074	0.086	0.18	0.096	0.077	0.067	0.066	0.062	0.088	0.063	**0.053**
70	PSNR	11.23	33.53	36.62	37.31	36.78	31.19	27.35	34.60	37.41	37.95	37.67	38.82	36.93	38.98	**39.14**
	SSIM	0.023	0.857	0.914	0.931	0.925	0.864	0.471	0.899	0.928	0.938	0.941	0.953	0.921	0.951	**0.954**
	SAM	1.011	0.164	0.083	0.084	0.085	0.106	0.213	0.112	0.085	0.083	0.073	0.065	0.110	0.073	**0.058**

The best and second best results are shown in boldface and underline, respectively.

TABLE II
INDEXES RESULTS OF ALL METHODS ON THE ICVL DATASET WITH CASES 2-6

Case	Indexes	Noisy	BM4D	TDL	ITSReg	LLRT	LRTV	NMoG	LRTDTV	HDCNN	QRNN	SQAD	GRN	FastHyMix	T3SC	Ours
2	PSNR	14.54	35.17	30.71	37.47	34.59	32.71	30.53	36.49	38.92	40.04	39.04	39.65	38.98	40.76	**40.79**
	SSIM	0.057	0.885	0.616	0.922	0.821	0.884	0.626	0.923	0.941	0.957	0.951	0.959	0.945	0.963	**0.965**
	SAM	0.908	0.143	0.298	0.101	0.171	0.087	0.174	0.093	0.075	0.071	0.081	0.066	0.097	0.065	**0.055**
3	PSNR	14.63	34.79	30.07	37.23	33.58	32.76	30.11	36.38	38.74	39.84	39.38	39.72	37.97	40.36	**40.52**
	SSIM	0.057	0.876	0.588	0.918	0.783	0.884	0.614	0.922	0.938	0.958	0.955	0.961	0.928	0.962	**0.965**
	SAM	0.903	0.154	0.313	0.106	0.195	0.088	0.252	0.097	0.078	0.072	0.071	0.065	0.116	0.072	**0.055**
4	PSNR	14.58	32.32	27.81	33.76	31.25	31.36	29.51	34.42	38.63	39.81	39.06	38.87	34.32	39.89	**40.93**
	SSIM	0.057	0.831	0.537	0.868	0.737	0.869	0.617	0.903	0.938	0.958	0.953	0.956	0.853	0.960	**0.967**
	SAM	0.916	0.173	0.351	0.145	0.209	0.129	0.251	0.120	0.078	0.069	0.066	0.071	0.141	0.080	**0.053**
5	PSNR	12.66	28.76	24.08	29.54	27.71	31.28	25.95	35.28	36.91	38.13	38.03	35.05	31.76	33.40	**39.46**
	SSIM	0.044	0.713	0.368	0.779	0.629	0.853	0.492	0.909	0.918	0.941	0.945	0.915	0.836	0.855	**0.958**
	SAM	0.905	0.435	0.541	0.417	0.422	0.228	0.483	0.121	0.120	0.091	0.084	0.145	0.450	0.315	**0.071**
6	PSNR	12.45	26.26	21.12	26.94	25.35	29.76	24.62	33.02	35.67	37.64	36.57	33.83	26.82	31.31	**39.29**
	SSIM	0.041	0.653	0.283	0.739	0.541	0.831	0.465	0.891	0.905	0.941	0.931	0.898	0.695	0.832	**0.957**
	SAM	0.922	0.472	0.591	0.453	0.461	0.296	0.496	0.137	0.125	0.092	0.085	0.147	0.485	0.357	**0.071**

The best and second best results are shown in boldface and underline, respectively.

TABLE III
COMPUTATIONAL COMPLEXITIES OF ALL METHODS ON THE ICVL DATASET WITH CASE 1

	BM4D	TDL	ITSReg	LLRT	LRTV	NMoG	LRTDTV	HDCNN	QRNN	SQAD	GRN	FastHyMix	T3SC	Ours
Time [s]	236.555	47.265	2812.476	2921.458	461.247	115.796	250.549	0.894	0.446	0.705	0.063	2.544	3.162	1.638
#Params [M]	—	—	—	—	—	—	—	0.40	0.86	0.31	0.38	—	1.06	0.45

The inference times of BM4D, TDL, ITSREG, LLRT, LRTV, NMOG, LRTDTV, and FASTHYMIX are cpu times. The inference times of HDCNN, QRNN, SQAD, GRN, T3SC and our method are GPU times.

Table III presents the computational complexity comparisons. BM4D, TDL, ITSReg, LLRT, LRTV, NMoG, LRTDTV, and FastHyMix are implemented by MATLAB, so the CPU times are reported. HDCNN, QRNN, SQAD, GRN, T3SC, and our method are executed using Python and running on GPU. Thus, the inference times are measured as the GPU times. Our method needs a few more inference time than other DL-based methods. Due to the considerable performance improvement over DL-based methods, such computational complexity of our method is acceptable.

C. Simulated Noise Experiment on Remote Sensing Dataset

In this section, the performance evaluation is conducted on the simulated noisy remote sensing images at the most severe noise Case 6. The OHS-3C dataset[15] is tested, which is acquired by the Zhuhai-1 OHS-3C hyperspectral satellite. First, we select two HSIs from OHS-3C dataset, that is, OHS-3C-UZ and OHS-3C-XJ, which are captured at Surxondaryo Bandikhan, Uzbekistan, and Xinjiang Uygur Autonomous Region, China, respectively. Each dataset is with the size $5057 \times 5056 \times 32$. Then, we crop one subimage of size $500 \times 500 \times 32$ from each dataset for testing. Besides, 2000 overlapped patches with size of $50 \times 50 \times 32$ are sampled from the rest parts to fine-tune the pretrained network for the OHS-3C-UZ and OHS-3C-XJ datasets, respectively. During fine-tuning, the epoch is set as 20. The learning rate starts from 2×10^{-4} and descends a factor of 0.1 after 5 epoches. GRN is not used in this experiment, since it cannot restore the HSI with different bands.

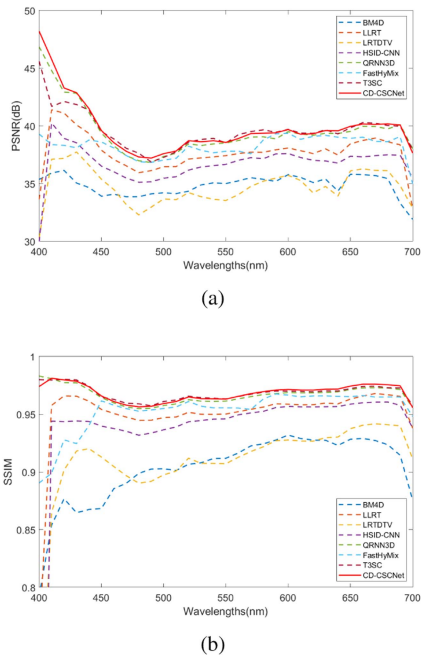
Figs. 7 and 8 show the false-color denoised results and spatial/spectral residual maps of different methods on the OHS-3C-UZ dataset at Case 6. BM4D [see Fig. 7(c)] introduce some artifacts and lose some spectral information. LLRT [see Fig. 7(d)], TDL [see Fig. 7(g)], and ITSReg [see Fig. 7(h)] fail to remove the severe complex noise. LRTV [see Fig. 7(f)] removes most of the noise, but produces blurring. The inconsistent color of Fig. 7(m) with ground truth [see Fig. 7(a)] indicates that T3SC generates apparent spectral distortion. In contrast, HDCNN,

[15][Online]. Available: https://www.obtdata.com/#/dataExpress

TABLE IV
INDEXES RESULTS OF ALL METHODS ON THE OHS-3C-UZ AND OHS-3C-XJ DATASETS WITH CASE 6

Dataset	Indexes	Noisy	BM4D	TDL	ITSReg	LLRT	LRTV	NMoG	LRTDTV	HDCNN	QRNN	SQAD	FastHyMix	T3SC	Ours
OHS-3C-UZ	PSNR	11.79	25.35	21.30	24.85	24.76	27.20	25.22	27.64	30.54	26.39	26.76	25.85	27.50	**32.44**
	SSIM	0.047	0.607	0.315	0.604	0.503	0.686	0.516	0.745	0.801	0.796	0.805	0.565	0.789	**0.854**
	SAM	0.556	0.142	0.213	0.150	0.147	0.062	0.159	0.105	0.170	0.195	0.163	0.133	0.089	**0.049**
OHS-3C-XJ	PSNR	12.58	24.36	17.62	23.36	22.73	26.71	23.75	27.89	28.43	26.48	26.99	24.41	25.81	**31.54**
	SSIM	0.069	0.558	0.197	0.522	0.407	0.662	0.462	0.712	0.738	0.759	0.758	0.497	0.702	**0.825**
	SAM	0.552	0.172	0.358	0.187	0.209	0.085	0.140	0.087	0.226	0.209	0.163	0.176	0.115	**0.056**

The best and second best results are shown in boldface and underline, respectively.



(a)

(b)

Fig. 6. PSNR and SSIM of the different methods results for each band on the ICVL dataset for Case 1 at $\sigma = 50$. (a) PSNR for each band. (b) SSIM for each band.

QRNN, SQAD, and our method obtain better visual effects. The detailed comparisons on the closeups show that our method generates the sharpest edges. Moreover, the visual comparisons on residual maps in Fig. 8 also verify that the denoised HSI obtained by our method is closest to ground truth in terms of spatial and spectral preservations.

Figs. 9 and 10 display the false-color composite of denoised results and the related residual maps for the OHS-3C-XJ dataset, respectively. Fig. 9(d) and (g) indicate that LLRT and TDL is unable to remove the complex noise. BM4D and ITSReg remove most Gaussian noise, but still contain some stripe noise, evidenced by the blue stripes in the top buildings region of Fig. 9(c) and (h). LRTDTV [see Fig. 9(e)], LRTV [see Fig. 9(f)], HDCNN [see Fig. 9(i)], QRNN [see Fig. 9(j)], FastHyMix [see Fig. 9(l)] and T3SC [see Fig. 9(m)] generate oversmooth edges, and lose some spatial details. Fig. 9(k) shows that SQAD introduces some artifacts. In contrast, our method [see Fig. 9(n)] can more effectively preserve the spatial structure and spectral information. This statement can also be proved by Fig. 10(l), obtained by our method, has the fewer highlighted regions than other residual maps.

Table IV reports the objective performance comparisons of all methods on the OHS-3C-UZ and OHS-3C-XJ datasets with complex noise Case 6. It can be seen that our method achieves remarkable gains over compared methods in terms of PSNR, SSIM, and SAM indexes.

D. Real Noise Experiment on Remote Sensing Dataset

In this section, we assess the performance of our method on the real noisy HSI. The real noise experiment not only verifies the practical application, but also evaluates the generalization of HSI denoising methods. The Shanghai dataset[16] is employed as the testing image, which is with the size of $300 \times 300 \times 155$. The Shanghai dataset is obtained by GaoFen-5 sensor, containing dense stripes and deadlines. Due to lack of ground truth, the network model trained on the ICVL dataset is directly transferred to the Shanghai dataset without additional fine-tuning. In addition, we adopt the first 155 bands of Washington DC Mall dataset[17] to train T3SC, and then it is used to restore the Shanghai dataset. Fig. 11 depicts the denoised results (composite of bands 152, 96, and 43) of all methods. The visual comparisons of closeups demonstrate that our method achieves the comparable visual quality to popular DL-based methods in terms of noise reduction and edge sharpness.

E. Ablation Studies

In this section, the ablations studies are given to investigate the effects of major hyperparameters in our method, including the unfolding number T, the channel number N in $\mathcal{C}_1^{spa}(\mathcal{C}_2^{spa})$, and the ratio $r = N/K$. The experiments are conducted on the ICVL dataset with Case 1 ($\sigma = 50$).

1) Unfolding number T: T represents the number of $\mathcal{J}_C^t$ ($t = 0, 1, \ldots, T-1$) used in CD-CSCNet, which controls the depth of network. We consider $T = \{2, 4, 6, 8, 10, 12\}$, and the ablation results are presented in Table V. It can be observed that the performance is improved with increasing T. Such improvement is waning with larger T. Specifically, $T = 10$ obtains 0.17 dB PSNR gain over $T = 8$. The comparison between $T = 12$ and $T = 10$ indicates that PSNR is only improved by 0.02 dB, and the improvements of SSIM and SAM are negligible, but the inference time increases 17.3%. For a tradeoff between denoising performance and computational cost, T is set to 10 in the experiments.

[16][Online]. Available: http://hipag.whu.edu.cn/resourcesdownload.html
[17][Online]. Available: https://engineering.purdue.edu/biehl/MultiSpec/hyperspectral.html

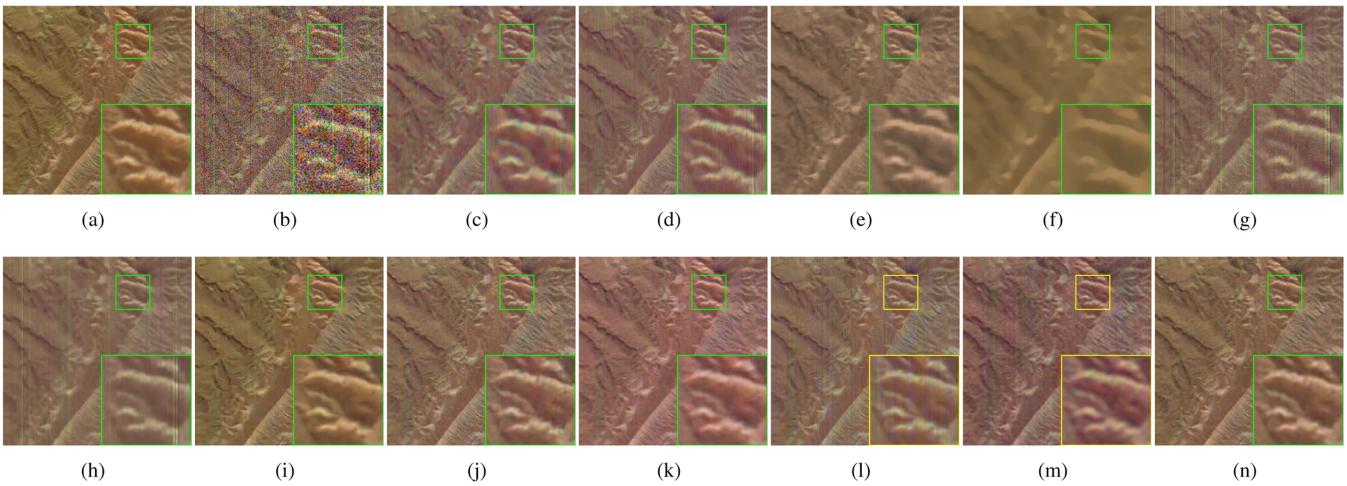


Fig. 7. Denoised results of different methods on the OHS-3C-UZ dataset for Case 6 (composite of bands 30, 16, and 6). (a) Ground truth. (b) Noisy HSI. (c) BM4D. (d) LLRT. (e) LRTDTV. (f) LRTV. (g) TDL. (h) ITSReg. (i) HDCNN. (j) QRNN. (k) SQAD. (l) FastHyMix. (m) T3SC. (n) Ours.

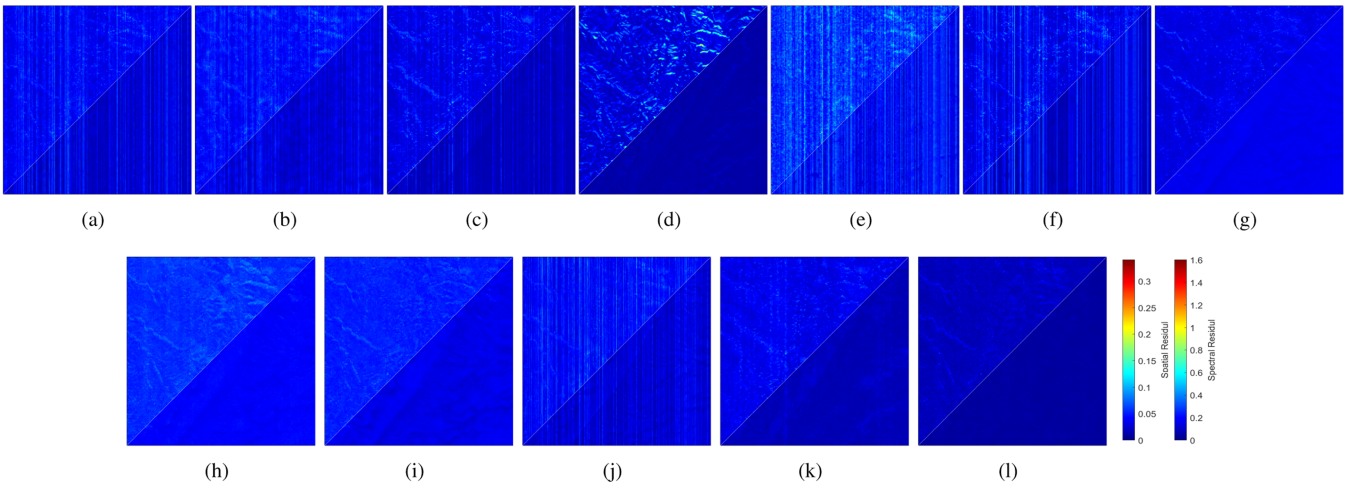


Fig. 8. Residual maps of different methods on the OHS-3C-UZ dataset for Case 6 (left: spatial residual map, right: spectral residual map). (a) BM4D. (b) LLRT. (c) LRTDTV. (d) LRTV. (e) TDL. (f) ITSReg. (g) HDCNN. (h) QRNN. (i) SQAD. (j) FastHyMix. (k) T3SC. (l) Ours.

TABLE V
ABLATION STUDY ON THE UNFOLDING NUMBER T

T	PSNR	SSIM	SAM	Time [s]
2	39.41	0.954	0.071	0.521
4	40.04	0.959	0.061	0.798
6	40.34	0.961	0.057	1.079
8	40.56	0.963	0.053	1.361
10	40.73	0.964	0.053	1.638
12	40.75	0.964	0.052	1.922

2) Channel number N in $\mathcal{C}_1^{spa}(\mathcal{C}_2^{spa})$: The channel numbers in $\mathcal{C}_1^{spa}$ and $\mathcal{C}_2^{spa}$ determine the size of convolution kernels, which also correspond to the size of convolutional sparse dictionaries. Without loss of generality, we assume that $\mathcal{C}_1^{spa}$ and $\mathcal{C}_2^{spa}$ have the same channels. We choose N among $\{32, 64, 128\}$. Table VI presents the ablation results on N. It can be seen that the PSNR, SSIM, and SAM scores are better when N is larger, but with cost of larger model complexity. Specifically, $N = 128$ obtains significant improvements over $N = 64$, but the inference time increase 142.3%, and the number of parameters is more than three times. To guarantee the computation efficiency, we set $N = 64$ in the experiments.

TABLE VI
ABLATION STUDY ON THE CHANNEL NUMBER N

N	PSNR	SSIM	SAM	Time [s]	#Params [M]
32	40.29	0.961	0.058	0.798	0.21
64	40.73	0.964	0.053	1.638	0.45
128	41.07	0.966	0.05	3.969	1.36

3) Ratio $r = N/K$: r aims to adjust the size of $\mathbf{W} \in \mathbb{R}^{N\times K}$, and determines the channel number in $\mathcal{B}_1^{spa}(\mathcal{B}_2^{spa})$ through

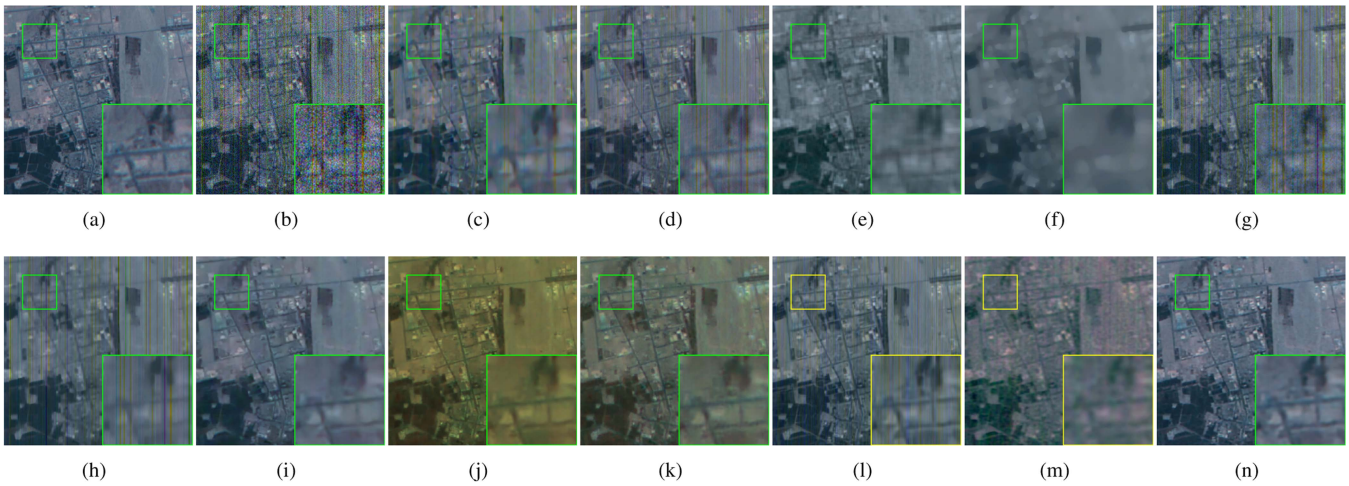


Fig. 9. Denoised results of different methods on the OHS-3C-XJ dataset for Case 6 (composite of bands 16, 8, and 2). (a) Ground truth. (b) Noisy HSI. (c) BM4D. (d) LLRT. (e) LRTDTV. (f) LRTV. (g) TDL. (h) ITSReg. (i) HDCNN. (j) QRNN. (k) SQAD. (l) FastHyMix. (m) T3SC. (n) Ours.

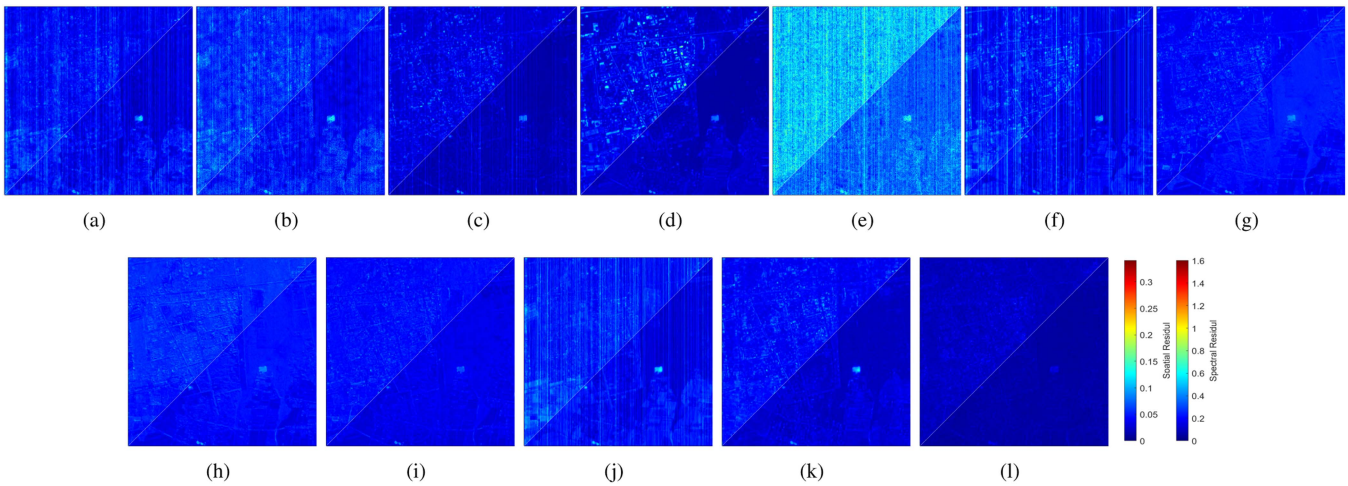


Fig. 10. Residual maps of different methods on the OHS-3C-XJ dataset for Case 6 (left: spatial residual map, right: spectral residual map). (a) BM4D. (b) LLRT. (c) LRTDTV. (d) LRTV. (e) TDL. (f) ITSReg. (g) HDCNN. (h) QRNN. (i) SQAD. (j) FastHyMix. (k) T3SC. (l) Ours.

TABLE VII
ABLATION STUDY ON THE RATIO r

r	PSNR	SSIM	SAM	Time [s]	#Params [M]
2	40.85	0.965	0.051	1.642	0.72
4	40.73	0.964	0.053	1.638	0.45
6	40.43	0.962	0.054	1.635	0.34

TABLE VIII
ABLATION STUDY ON THE TYPE OF CONVOLUTION

Type of Conv	PSNR	SSIM	SAM	Time [s]	#Params [M]
Normal 3D Conv	39.76	0.959	0.063	1.702	0.12
CD-Conv	40.67	0.963	0.054	2.162	0.49
SCD-Conv	40.73	0.964	0.053	1.638	0.45

$\boldsymbol{\mathcal{C}}_1^{spa} = \boldsymbol{\mathcal{B}}_1^{spa} \times_5 \mathbf{W}_1 (\boldsymbol{\mathcal{C}}_2^{spa} = \boldsymbol{\mathcal{B}}_2^{spa} \times_5 \mathbf{W}_2)$. In this perspective, r can also be called as compression ratio. Larger r implies bigger compression ratio, and can generate smaller size of $\boldsymbol{\mathcal{B}}_1^{spa} (\boldsymbol{\mathcal{B}}_2^{spa})$. It can be found in Table VII that larger r yields smaller computational complexity, but causes a certain performance reduction. Specifically, comparing with $r = 2$, $r = 4$ reduces the PSNR by 0.12 dB and saves number of parameters by 37.5%. Although $r = 6$ reduces by half the number of parameters, the PSNR score descends 0.42 dB. Therefore, r is set to 4 in our experiments.

4) SCD-Conv: To study the effectiveness of SCD-Conv, we carry out other two variants of CD-CSCNet, that is, the SCD-Conv is replaced as normal 3-D Conv and CD-Conv, respectively. The training and testing of these two variants are the same as the above experiments. Table VIII gives the comparisons among the normal 3-D Conv, CD-Conv, and SCD-Conv. The comparison between normal 3-D Conv and CD-Conv indicates that the content dependent approach can obtain 0.91 dB PSNR

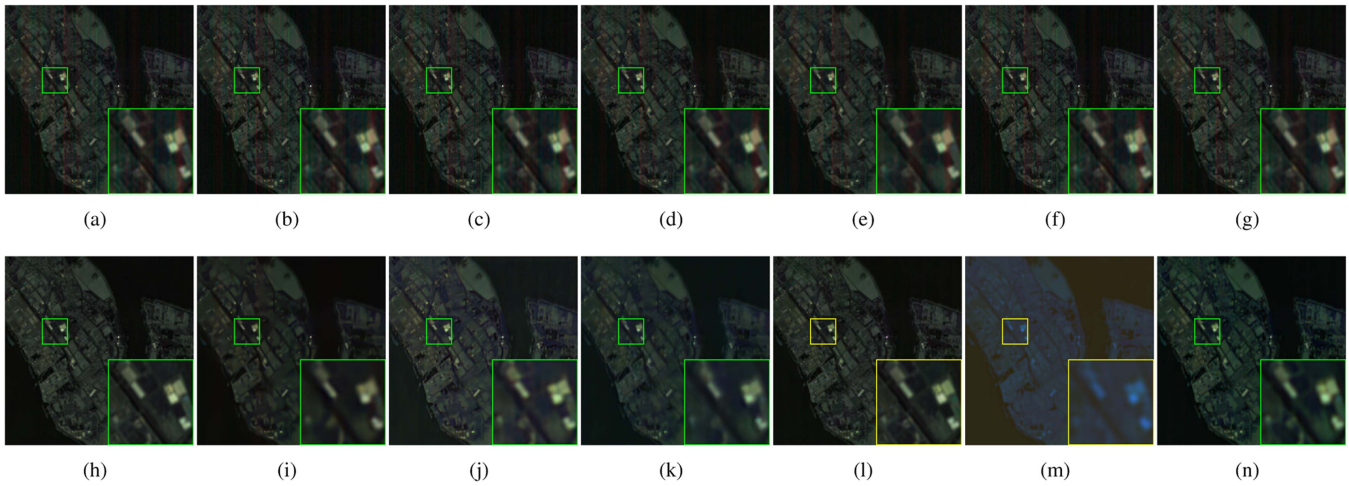


Fig. 11. Denoised results of different methods on the Shanghai dataset for real noise (composite of bands 152, 96, and 43). (a) Noisy HSI. (b) BM4D. (c) LLRT. (d) LRTDTV. (e) LRTV. (f) TDL. (g) ITSReg. (h) NMoG. (i) HDCNN. (j) QRNN. (k) SQAD. (l) FastHyMix. (m) T3SC. (n) Ours.

gains. The comparison between CD-Conv and SCD-Conv verifies that the separable approach offers 0.06 dB PSNR gains further and reduces the inference time by 24.2%.

V. Conclusion

In this article, we proposed the CD-CSCNet, a novel DU based HSI denoising method, which exploits the interpretability of 3-D CSC and the power of DL. Specifically, we first developed the CD-CSC model, which expresses any HSI using the content-dependent 3-D filters. The unique 3-D filters can improve the accuracy and adaptivity of joint spatial-spectral representation. Then, we proposed an end-to-end CD-CSCNet by integrating three implicit deep modules (head feature extraction network, weight generator network and reconstruction network) into the deeply unfolded 3-D CSC network. Moreover, we designed a CD-Conv operation in CD-CSCNet using the insight of content-dependent, which is then extended into its separable version SCD-Conv. Extensive simulated noise and real noise experiments on benchmark datasets demonstrated the superiority of our method. In the future work, we will study lightweight approach to shorten the inference time of our method further.

References

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[2] Y. Shen, L. Shi, J. Zhao, Y. Dong, and L. Wang, "Fully convolutional spectral–spatial fusion network integrating supervised contrastive learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 9077–9088, 2023.

[3] Y. Zhu, K. Yuan, W. Zhong, and L. Xu, "Spatial–spectral convnext for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5453–5463, 2023.

[4] S. Wang, Z. Liu, Y. Chen, C. Hou, A. Liu, and Z. Zhang, "Expansion spectral–spatial attention network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 6411–6427, 2023.

[5] X. Qiao, S. K. Roy, and W. Huang, "Rotation is all you need: Cross dimensional residual interaction for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5387–5404, 2023.

[6] D. Zhu, B. Du, and L. Zhang, "Target dictionary construction-based sparse representation hyperspectral target detection methods," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1254–1264, Apr. 2019.

[7] W. Xie, X. Zhang, Y. Li, K. Wang, and Q. Du, "Background learning based on target suppression constraint for hyperspectral target detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5887–5897, 2020.

[8] K. Yu et al., "A parallel algorithm for hyperspectral target detection based on weighted alternating direction method of multiplier," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 8274–8285, 2023.

[9] B. Rasti, Y. Chang, E. Dalsasso, L. Denis, and P. Ghamisi, "Image restoration for remote sensing: Overview and toolbox," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 2, pp. 201–230, Jun. 2022.

[10] H. Othman and S.-E. Qian, "Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 397–408, Feb. 2006.

[11] G. Chen and S.-E. Qian, "Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 973–980, Mar. 2011.

[12] B. Rasti, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Hyperspectral image denoising using 3D wavelets," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 1349–1352.

[13] T. Lin and S. Bourennane, "Hyperspectral image processing by jointly filtering wavelet componenttensor," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 6, pp. 3529–3541, Jun. 2013.

[14] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119–133, Jan. 2013.

[15] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral–spatial adaptive total variation model," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3660–3677, Oct. 2012.

[16] H. K. Aggarwal and A. Majumdar, "Hyperspectral image denoising using spatio-spectral total variation," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 442–446, Mar. 2016.

[17] Y. Chang, L. Yan, H. Fang, and C. Luo, "Anisotropic spectral-spatial total variation model for multispectral remote sensing image destriping," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1852–1866, Jun. 2015.

[18] L. Zhang, Y. Qian, J. Han, P. Duan, and P. Ghamisi, "Mixed noise removal for hyperspectral image with l_0-l_{1-2} SSTV regularization," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 5371–5387, 2022.

[19] S. Takemoto, K. Naganuma, and S. Ono, "Graph spatio-spectral total variation model for hyperspectral image denoising," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6012405.

[20] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4729–4743, Aug. 2014.

[21] Y. Xie, Y. Qu, D. Tao, W. Wu, Q. Yuan, and W. Zhang, "Hyperspectral image restoration via iteratively regularized weighted Schatten p-norm minimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4642–4659, Aug. 2016.

[22] Y. Chen, T.-Z. Huang, W. He, X.-L. Zhao, H. Zhang, and J. Zeng, "Hyperspectral image denoising using factor group sparsity-regularized nonconvex low-rank approximation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5515916.

[23] F. Xu, Y. Chen, C. Peng, Y. Wang, X. Liu, and G. He, "Denoising of hyperspectral image using low-rank matrix factorization," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1141–1145, Jul. 2017.

[24] B. Du, Z. Huang, and N. Wang, "A bandwise noise model combined with low-rank matrix factorization for hyperspectral image denoising," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1070–1081, Apr. 2018.

[25] Y. Chen, X. Cao, Q. Zhao, D. Meng, and Z. Xu, "Denoising hyperspectral image withnon-i.i.d. noise structure," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 1054–1066, Mar. 2018.

[26] J. Xue, Y. Zhao, W. Liao, and S. G. Kong, "Joint spatial and spectral low-rank regularization for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1940–1958, Apr. 2018.

[27] Q. Xie et al., "Multispectral images denoising by intrinsictensor sparsity regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1692–1700.

[28] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang, "Decomposable nonlocaltensor dictionary learning for multispectral image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2949–2956.

[29] X. Bai, F. Xu, L. Zhou, Y. Xing, L. Bai, and J. Zhou, "Nonlocal similarity basednonnegative tucker decomposition for hyperspectral image denoising," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 701–712, Feb. 2018.

[30] J. Xue, Y. Zhao, W. Liao, and J. C.-W. Chan, "Nonlocal low-rank regularizedtensor decomposition for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5174–5189, Jul. 2019.

[31] X. Gong, W. Chen, and J. Chen, "A low-ranktensor dictionary learning method for hyperspectral image denoising," *IEEE Trans. Signal Process.*, vol. 68, pp. 1168–1180, 2020.

[32] Y. Chen, S. Wang, and Y. Zhou, "Tensor nuclear norm-based low-rank approximation with total variation regularization," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1364–1377, Dec. 2018.

[33] Y. Chang, L. Yan, X.-L. Zhao, H. Fang, Z. Zhang, and S. Zhong, "Weighted low-ranktensor recovery for hyperspectral image restoration," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4558–4572, Nov. 2020.

[34] H. Fan, Y. Chen, Y. Guo, H. Zhang, and G. Kuang, "Hyperspectral image restoration using low-ranktensor recovery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 10, pp. 4589–4604, Jul. 2017.

[35] Y. Chen, W. He, N. Yokoya, T.-Z. Huang, and X.-L. Zhao, "Nonlocaltensor-ring decomposition for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1348–1362, Feb. 2020.

[36] N. Liu, W. Li, R. Tao, Q. Du, and J. Chanussot, "Multigraph-based low-ranktensor approximation for hyperspectral image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, May 2022, Art. no. 5530314.

[37] B. Zhao, M. O. Ulfarsson, J. R. Sveinsson, and J. Chanussot, "Hyperspectral image denoising using spectral-spatial transform-based sparse and low-rank representations," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522125.

[38] Y. Chang, L. Yan, B. Chen, S. Zhong, and Y. Tian, "Hyperspectral image restoration: Where does the low-rank property exist," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6869–6884, Aug. 2021.

[39] W. He, H. Zhang, L. Zhang, and H. Shen, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 178–188, Jan. 2016.

[40] J. Cai, W. He, and H. Zhang, "Anisotropic spatial–spectral total variation regularized double low-rank approximation for HSI denoising and destriping," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5536619.

[41] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X.-L. Zhao, and D. Meng, "Hyperspectral image restoration via total variation regularized low-ranktensor decomposition," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1227–1243, Apr. 2018.

[42] H. Zeng, S. Huang, Y. Chen, H. Luong, and W. Philips, "All of low-rank and sparse: A recast total variation approach to hyperspectral denoising," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 7357–7373, 2023.

[43] Y. Chang, L. Yan, and S. Zhong, "Hyper-Laplacian regularized unidirectional low-ranktensor recovery for multispectral image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5901–5909.

[44] T. Xie, S. Li, and J. Lai, "Adaptive rank and structured sparsity corrections for hyperspectral image restoration," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8729–8740, Sep. 2022.

[45] Y. Chang, L. Yan, H. Fang, S. Zhong, and W. Liao, "HSI-DeNet: Hyperspectral image restoration via convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 667–682, Feb. 2019.

[46] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang, "Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1205–1218, Feb. 2019.

[47] X. Cao, X. Fu, C. Xu, and D. Meng, "Deep spatial-spectral global reasoning network for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5504714.

[48] Y. Yuan, H. Ma, and G. Liu, "Partial-DNet: A novelblind denoising model with noise intensity estimation for HSI," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5505913.

[49] Z. Wang, Z. Shao, X. Huang, J. Wang, and T. Lu, "SSCAN: A spatial–spectral cross attention network for hyperspectral image denoising," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 5508805.

[50] C. Zou, C. Zhang, M. Wei, and C. Zou, "Enhanced channel attention network with cross-layer feature fusion for spectral reconstruction in the presence of Gaussian noise," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 9497–9508, 2022.

[51] W. Shen, J. Liu, J. Li, and C. Tian, "Deep tensor attention prior network for hyperspectral image denoising," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 6448–6462, 2023.

[52] E. Pan, Y. Ma, X. Mei, F. Fan, J. Huang, and J. Ma, "SQAD: Spatial-spectralquasi-attention recurrent network for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5524814.

[53] G. Fu, F. Xiong, J. Lu, J. Zhou, and Y. Qian, "Nonlocal spatial–spectral neural network for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5541916.

[54] H. Pan, F. Gao, J. Dong, and Q. Du, "Multiscale adaptive fusion network for hyperspectral image denoising," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 3045–3059, 2023.

[55] K. Wei, Y. Fu, and H. Huang, "3-Dquasi-recurrent neural network for hyperspectral image denoising," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 363–375, Jan. 2021.

[56] M. Li, J. Liu, Y. Fu, Y. Zhang, and D. Dou, "Spectral enhanced rectangle transformer for hyperspectral image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 5805–5814.

[57] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[58] H. Sun, M. Liu, K. Zheng, D. Yang, J. Li, and L. Gao, "Hyperspectral image denoising via low-rank representation and CNN denoiser," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 716–728, 2022.

[59] F. Xiong, J. Zhou, Q. Zhao, J. Lu, and Y. Qian, "MAC-Net: Model-aided nonlocal neural network for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5519414.

[60] B. Lin, X. Tao, and J. Lu, "Hyperspectral image denoising via matrix factorization and deep prior regularization," *IEEE Trans. Image Process.*, vol. 29, pp. 565–578, 2020.

[61] L. Zhuang and M. K. Ng, "Fasthymix: Fast and parameter-free hyperspectral image mixed noise removal," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4702–4716, Aug. 2023.

[62] T. Bodrito, A. Zouaoui, J. Chanussot, and J. Mairal, "A trainable spectral-spatial sparse coding model for hyperspectral image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 5430–5442.

[63] C. Garcia-Cardona and B. Wohlberg, "Convolutional dictionary learning: A comparative review and new algorithms," *IEEE Trans. Comput. Imag.*, vol. 4, no. 3, pp. 366–381, Sep. 2018.

Haitao Yin (Member, IEEE) received the B.S. degree in mathematics and applied mathematics, the M.S. degree in applied mathematics, and the Ph.D degree in control science and engineering from Hunan University, Changsha, China, in 2007, 2009, and 2012, respectively.

From 2019 to 2020, he was a Research Associate with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. He is currently an Associate Professor with the College of Automation and the College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include deep learning, sparse representation, and image processing.

Hao Chen received the B.E. degree in electrical engineering and automation from the Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, China, in 2021. He is currently working toward the master's degree in electronic information with Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China.

His research interests include remote sensing image processing and deep learning.