remote sensing

MDPI

Article

Hyperspectral Image Mixed Noise Removal Using Subspace Representation and Deep CNN Image Prior

Lina Zhuang [1,2], Michael K. Ng [1] and Xiyou Fu [3,4,5,*]

1 Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong, China; linaz@hku.hk (L.Z.); mng@maths.hku.hk (M.K.N.)
2 Shenzhen Institute of Research and Innovation, The University of Hong Kong, Shenzhen 518057, China
3 College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
4 MNR Key Laboratory for Geo-Environmental Monitoring of Great Bay Area, Shenzhen University, Shenzhen 518060, China
5 SZU Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518060, China
* Correspondence: fuxy0623@szu.edu.cn

Abstract: The ever-increasing spectral resolution of hyperspectral images (HSIs) is often obtained at the cost of a decrease in the signal-to-noise ratio (SNR) of the measurements. The decreased SNR reduces the reliability of measured features or information extracted from HSIs, thus calling for effective denoising techniques. This work aims to estimate clean HSIs from observations corrupted by mixed noise (containing Gaussian noise, impulse noise, and dead-lines/stripes) by exploiting two main characteristics of hyperspectral data, namely low-rankness in the spectral domain and high correlation in the spatial domain. We take advantage of the spectral low-rankness of HSIs by representing spectral vectors in an orthogonal subspace, which is learned from observed images by a new method. Subspace representation coefficients of HSIs are learned by solving an optimization problem plugged with an image prior extracted from a neural denoising network. The proposed method is evaluated on simulated and real HSIs. An exhaustive array of experiments and comparisons with state-of-the-art denoisers were carried out.

Keywords: hyperspectral image denoising; hyperspectral image restoration; low-rank representation; plug-and-play; sparse representation

check for updates

Citation: Zhuang, L.; Ng, M.K.; Fu, X. Hyperspectral Image Mixed Noise Removal Using Subspace Representation and Deep CNN Image Prior. *Remote Sens.* **2021**, *13*, 4098. https://doi.org/10.3390/rs13204098

Academic Editor: Paul Scheunders

Received: 29 August 2021
Accepted: 8 October 2021
Published: 13 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Hyperspectral cameras measure the radiation arriving at a sensor with high spectral resolution over a sufficiently broad spectral band such that the acquired spectrum can be used to uniquely characterize and identify any given material [1]. Hyperspectral imaging plays an important role in remote sensing and has been used in a wide array of applications, such as the identification of various minerals in mining and oil industries [2], monitoring the development and health of crops in agriculture [3], detecting the development of cracks in pavements in civil engineering [4], and so on.

A hyperspectral image (HSI) is a three-dimensional data cube, where the first two dimensions represent the spatial domain and the third dimension represents the spectral domain. In contrast to multispectral imaging, hyperspectral cameras capture electromagnetic information in hundreds of narrow spectral bands, instead of multiple wide spectral bands. However, due to the decrease of the width of spectral bands, hyperspectral cameras receive fewer photons per band and tend to acquire images with a lower signal-to-noise ratio (SNR). The decreased SNR reduces the reliability of measured features or information extracted from HSIs [1]. Therefore, hyperspectral image denoising is a fundamental inverse problem before further applications.

The degradations linked with various mechanisms also result in different types of noise, such as Gaussian noise, Poissonian noise, impulse noise, dead-lines/stripes, and

cross-track illumination variation. In this paper, we focus on the discussion of additive and signal-independent noise (namely, Gaussian noise, impulse noise, and dead-lines/stripes) and attack hyperspectral mixed noise composed of these additive noises.

The hyperspectral image denoising problem is usually studied by exploiting the characteristics of HSIs and noise. Some models for clean HSIs (i.e., priors, regularizers, constraints) have been demonstrated to work well in hyperspectral image denoising problems [5,6]. For example, the spectral–spatial adaptive hyperspectral total variation (SSAHTV) denoising method [7] uses a vector total variation (TV) regularization, which can promote localized step gradients within the image bands and align the "discontinuities" across the bands. Because of the very high spectral–spatial correlation, an HSI admits a sparse representation on a given basis or frame [8,9]; e.g., the Fourier basis [10], wavelet basis [11], Discrete Cosine basis, and data-adaptive basis [8]. For example, a 3D wavelet-based hyperspectral denoising method was introduced in [11], which took advantage of a fundamental property of HSIs; i.e., 3D wavelet coefficients of HSIs are sparse or compressible. The sparsity of coefficients means that the majority of coefficients are zero or close to zero. The compressibility of coefficients means that its elements have fast-decaying tails. In the 3D wavelet-based denoising method, the ℓ_1 norm, jointly with a quadratic data fidelity term, is used to promote sparsity on the wavelet coefficients.

The low rankness of HSIs in the spectral domain is also a widely used image prior for solving the denoising problem [12,13]. By minimizing the rank of the estimated image [14], one can remove a bulk of the Gaussian noise in observations due to the fact that Gaussian noise is full-rank. The non-convex rank constraint is usually relaxed by minimizing the nuclear norm of HSIs, as done in the low-rank matrix recovery (LRMR) method [15] and in the weighted sum of weighted tensor nuclear norm minimization (WSWTNNM) method [16]. The low-rank structure of HSIs is also exploited by representing the spectral vectors of the clean image in an orthogonal subspace in [8,17–20] for Gaussian noise removal and in [21,22] for mixed noise removal. Subspace representation is an explicit low-rank representation in the sense that the rank is constrained by the dimension of subspace.

The self-similarity of HSIs also has been exploited for HSI denoising problems mainly in two ways: (a) the non-local (generalized) mean: for each patch, seek for its similar patches in the image and produce a patch estimate based on the found patches [17]; (b) dictionary learning: express each patch as a sparse representation in a given dictionary, which may be learned from the data. Patch-based learning is arguably the state-of-the-art in HSI denoising [8,18].

Recently, deep learning techniques also have been developed for image denoising. Aiming at single-band images or RGB images, representative networks are feed-forward denoising convolutional neural networks (DnCNNs) [23], a fast and flexible denoising convolutional neural network (FFDNet) [24], and a convolutional blind denoising network (CBDNet) [25]. Some neural networks have been conceived for HSI; for example, a spatial–spectral gradient network (SSGN) [26], a CNN-based HSI denoising method HSI-DeNet [27], and a novel deep spatio-spectral Bayesian posterior (DSSBP) framework [28]. As the performance of deep learning based denoisers highly depends on the quality and quantity of training data, a challenge of deep-denoisers is the lack of real HSIs that can be used as training data or the simulation of pairs of clean–noisy images close to real HSIs. However, once a network is trained properly, deep-denoisers are much faster than the traditional machine learning (ML)-based denoisers. One potential solution is to incorporate deep-denoisers that have been well-trained using vast amounts of RGB images into the HSI denoising framework. This paper studies this possibility in the sense that we incorporate the well-known single-band deep denoiser, FFDNet, into a mixed noise removal framework derived using a traditional ML technique. These fall into the area of research called the plug-and-play technique [29,30] or regularization by denoising (RED) framework [31].

1.1. Related Work

The subspace representation of spectral vectors in HSIs has been successfully used to remove Gaussian noise by regularizing the representation coefficients of HSIs. We refer to representative work on global and nonlocal low-rank factorizations (GLF) [17], fast hyperspectral denoising (FastHyDe) [8], and non-local meets global (NGmeet) [18]. The idea of regularizing the subspace representation coefficients of HSIs underpins state-of-the-art Gaussian denoisers and has been extended to address mixed noise. The challenge of this extension lies in the estimation of the subspace.

A spectral subspace can be estimated from an observed HSI by simply carrying out the singular value decomposition (SVD) of observed image matrix when noise is i.i.d. Gaussian. When noise is a mixture of Gaussian noise, stripes, and impulse noise, the spectral subspace is usually estimated jointly with subspace coefficients of the HSI; for example, in double-factor-regularized low-rank tensor factorization (LRTF-DFR) [21] and subspace-based nonlocal low-rank and sparse factorization (SNLRSF) [22]. The joint estimation of the subspace and the corresponding coefficients of the HSI usually produce poor estimates of the subspace when HSI is affected by severe mixed noise. To sidestep the joint estimation, one strategy is to estimate the subspace and the corresponding coefficients of the HSI separately, which motivated us to develop the L1HyMixDe [32] method.

The proposed method in this paper is also based on two approaches: (a) learning subspace and coefficients of the HSI separately and (b) regularizing representation coefficients. However, the new work is distinct from L1HyMixDe in terms of the method of learning the subspace and the regularization imposed on the subspace representation coefficients. The main difference in subspace learning is that L1HyMixDe performs median filtering band by band, exploiting spatial correlation, whereas the new work estimates a coarse image using Hampel filtering pixel by pixel, exploiting spectral correlation. The Hampel filtering is more effective in outlier removal procedures. Furthermore, as anomalous target detection is an important task in hyperspectral imaging, it occurs that both anomalous targets and sparse noise are sparsely distributed in the spatial domain. If only the spatial information is considered—e.g., L1HyMixDe performs median filtering band by band, exploiting spatial correlation—then anomalous targets may be wrongly detected as sparse noise and would not be represented in the estimated subspace. This motivated us to develop a new method exploiting spectral information; i.e., spectral signatures of anomalous targets are smooth while spectral signatures of materials corrupted by sparse noise have unusual jumps in the value of the pixels. Furthermore, L1HyMixDe regularizes subspace representation coefficients with a non-local image prior, while this work adopts a more powerful deep CNN image prior. Compared with using non-local patch-based image priors, the new method using a deep-learning-based image prior is much faster as long as the deep network has been well trained. A computational efficient HSI mixed noise removal method is of importance in practice.

1.2. Contributions

The work aims to estimate a clean HSI from observations corrupted by mixed noise (containing Gaussian noise, impulse noise, and dead-lines/stripes) by exploiting two main characteristics of hyperspectral data, namely low-rankness in the spectral domain and high correlation in the spatial domain. Contributions of this work are summarized as follows:

- Instead of estimating the subspace basis and the corresponding coefficients of HSIs jointly and iteratively, we decouple the estimation of the subspace basis and the corresponding coefficients. A new subspace learning method, which works independently from coefficient estimation and is robust to mixed noise, is proposed.
- An image prior extracted from a state-of-the-art neural denoising network, FFDNet, is seamlessly embedded within our HSI mixed noise removal framework, which is a successful combination of the traditional machine learning technique and deep learning technique.

This paper is organized as follows. Section 2 formally introduces a mixed noise removal method—HySuDeep. Section 3 extends the proposed method to address mixed noise containing non-i.i.d. Gaussian noise. Sections 4 and 5 show and analyze the experimental results of the proposed method and the comparison methods. Finally, we present a conclusion of this paper in Section 6.

2. Problem Formulation

Some notations and tensor operations used in this paper and their definitions are provided in Table 1.

Table 1. Notations And Definitions.

Notation	Definition
$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$	Three-dimensional tensor (calligraphic letter)
$\mathbf{X}$	Matrix (boldface capital letter)
$\mathbf{x}$	Vector (boldface lowercase letter)
x	Scalar (italic lowercase letter)
Order N	Number of dimensions in a tensor
Mode n	nth dimension of a tensor
Mode-3 vectors of $\mathcal{X}$	I_3-dimensional vectors obtained from $\mathcal{X}$ by varying the 3rd index while keeping 1st and 2nd indices fixed.
Mode-3 slices of $\mathcal{X}$	Matrices obtained from $\mathcal{X}$ by varing every index but the 3rd index.
$\mathcal{X}(:,:,i)$	ith mode-3 slice of $\mathcal{X}$, a matrix obtained by fixing the mode-3 index of $\mathcal{X}$ to be i.
$\mathbf{X}_{(3)} \in \mathbb{R}^{I_3 \times (I_1 * I_2)}$	Mode-3 unfolding of $\mathcal{X}$. A tensor can be unfolded into a matrix by rearranging its mode-3 vectors, which are the column vectors of $\mathbf{X}_{(3)}$.
$\mathcal{X} \times_3 \mathbf{E}$	Tensor matrix multiplication. The mode-3 product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ by a matrix $\mathbf{E} \in \mathbb{R}^{J_n \times I_3}$ is a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times J_n}$, denoted as $\mathcal{Y} = \mathcal{X} \times_3 \mathbf{E}$, which is corresponding to a matrix multiplication, $\mathbf{Y}_{(3)} = \mathbf{E}\mathbf{X}_{(3)}$.
$\lVert \mathcal{X} \rVert_F$	The definition of Frobenius norm of a matrix is extended to a tensor as follows: $\lVert \mathcal{X} \rVert_F = \sqrt{\sum_{i_1,\ldots,i_N} \lvert x_{i_1,\ldots,i_N} \rvert^2}$

2.1. Observation Model

Let $\mathcal{X} \in \mathbb{R}^{r \times c \times n_b}$ denote an underlying clean HSI with $r \times c$ pixels and n_b bands. Assuming that noise is additive, we can write an observation model as

$$\mathcal{Y} = \mathcal{X} + \mathcal{S} + \mathcal{G}, \tag{1}$$

where $\mathcal{Y}, \mathcal{S}, \mathcal{G} \in \mathbb{R}^{r \times c \times n_b}$ are observed HSI data, sparse noise, and Gaussian noise, respectively. Elements in $\mathcal{G}$ are assumed to be i.i.d. Gaussian with zero-mean and variance σ^2. Non-i.i.d. noise is discussed in Section 3.

2.2. Subspace Representation of HSIs

The high correlation in the spectral domain of HSIs implies that spectral vectors (i.e., mode-3 vectors) of an HSI $\mathcal{X}$ can be well represented in a low-dimensional subspace [1,8,19]. We take a cropped Washington DC Mall image (shown in Figure 1a) as an example. To analyze the correlation of pixel values in the spectral domain, we obtained the autocorrelation matrix of its spectral vectors by computing $\mathbf{E}[\mathbf{X}_{(3)}\mathbf{X}_{(3)}^T]$. From Figure 1b, we can see that the first 100 bands are highly correlated. Due to the high spectral correlation, the autocorrelation matrix can be approximated by a low-rank matrix. The low-rankness may also be inferred in Figure 1c, where the first two singular values (the largest) dominate all other singular values, which decrease to near-zero rapidly. From Figure 1d,e, we can see that two spectral vectors of size 191 can be well approximated by vectors in a subspace with a dimension of 10.

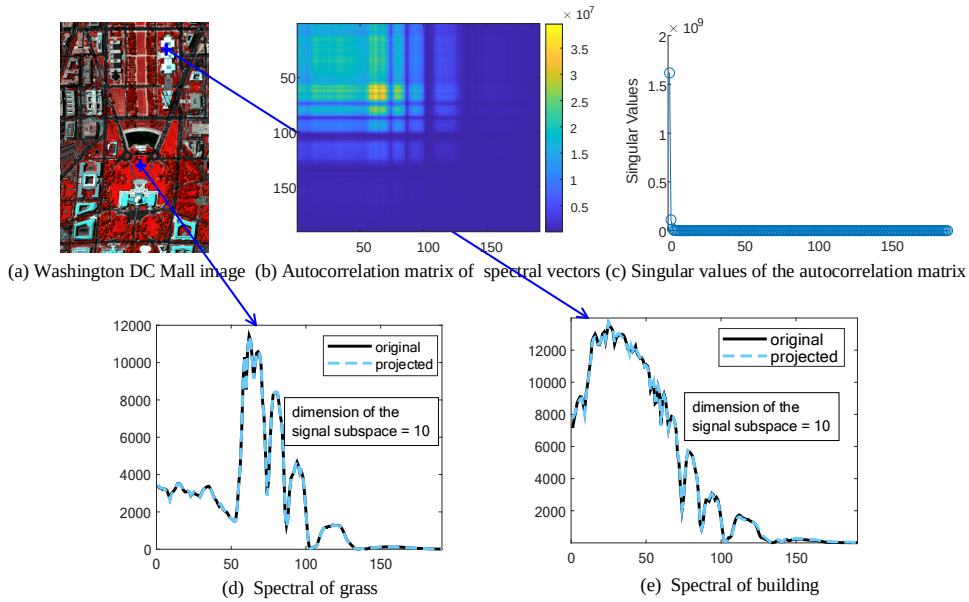

Figure 1. Subspace representation of spectral vectors in a subset of a Washington DC Mall image.

Based on the subspace representation of spectral vectors, we write

$$\mathcal{X} = \mathcal{Z} \times_3 \mathbf{E} + \Xi, \tag{2}$$

with $\mathbf{E} \in \mathbb{R}^{n_b \times p}$ ($p \ll n_b$), $\mathcal{Z} \in \mathbb{R}^{r \times c \times p}$, and $\Xi \in \mathbb{R}^{r \times c \times n_b}$. Matrix $\mathbf{E}$ holds an orthogonal basis for the signal subspace, and the entries of $\mathcal{Z}$ are representation coefficients of $\mathcal{X}$ with respect to $\mathbf{E}$. Hereafter, mode-3 slices of $\mathcal{Z}$ are termed the eigenimages. Tensor Ξ represents residuals of a low-dimensional representation of $\mathcal{X}$. When SVD is adopted, most of the image variation can be compactly represented in a low-dimensional subspace, and the residual tensor contains a small amount of image energy. Thus, we omit Ξ when reconstructing the image; that is, $\mathcal{X} = \mathcal{Z} \times_3 \mathbf{E}$. Figure 2 illustrates the subspace representation of a hyperspectral image. The observation model (1) can be written as

$$\mathcal{Y} = \mathcal{Z} \times_3 \mathbf{E} + \mathcal{S} + \mathcal{G}. \tag{3}$$

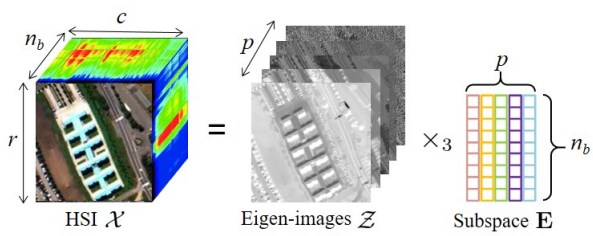

Figure 2. Subspace representation of a hyperspectral image.

The subspace representation of spectral vectors has been demonstrated to be a powerful tool for solving some hyperspectral inverse problems, such as hyperspectral image super-resolution [33], hyperspectral image segmentation [34], hyperspectral image classification [35], and hyperspectral unmixing [36]. In this paper, we propose a hyperspectral image mixed noise removal method using subspace representation and deep CNN Image prior (**HySuDeep**), the flowchart of which is illustrated in Figure 3.

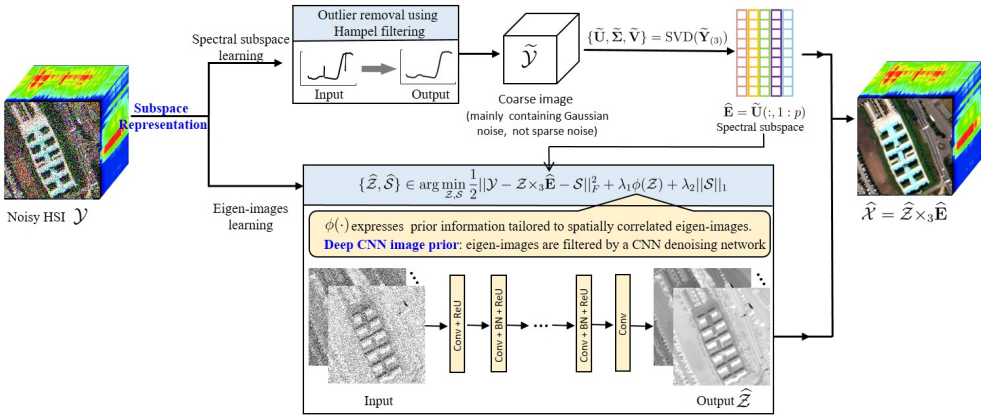

Figure 3. Flowchart of the proposed mixed noise removal method HySuDeep.

However, for the hyperspectral mixed noise removal problem, the challenge of exploiting subspace representation lies in the estimation of the subspace from observations corrupted not only by Gaussian noise but also by sparse noise.

2.3. Subspace Learning against Mixed Noise

When noise in an observed HSI is approximately Gaussian distributed, the subspace of the underlying clean image can be estimated easily from the noisy image by performing SVD on the noisy image matrix; that is, $\{\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}\} = \mathrm{SVD}(\mathbf{Y}_{(3)})$. The estimated subspace is spanned by k eigenvectors corresponding to the k largest eigenvalues; that is, $\widehat{\mathbf{E}} = [\mathbf{U}]_{:,1:k}$, where the subscript $(:, 1:k)$ means extracting the first k columns from a matrix.

However, when noise is a mixture of Gaussian noise and stripe noise, performing SVD on a noisy image matrix is not an optimal way to learn the spectral subspace. SVD aims to find the most important dimensions in the data by finding the directions of maximum variance, and the addition of i.i.d. Gaussian noise increases the variance of each dimension uniformly (thus, it does not change the order of singular values and consequently does not change the estimation of subspace), whereas the addition of stripes increases the variance of each dimension non-uniformly (thus, it does change the order of singular values). Experimental analysis is given in Section 4.3.

One of our contributions is to estimate the spectral subspace from observations corrupted by mixed noise and thus propose an efficient denoising method, which does not need to estimate spectral subspace iteratively.

2.3.1. Outlier Removal Using Hampel Filtering

Since the existence of sparse noise significantly disturbs the identification of signal subspace, we may estimate a coarse image by removing the sparse noise from observations. An outlier removal step applying Hampel filtering to spectral vectors of the observed image is introduced below.

Given a spectral vector of observed HSI, denoted as $\mathbf{y} = [y_1, y_2, \ldots, y_{n_b}]^T \in \mathbb{R}^{n_b}$, and a sliding window of length q, we define the point-to-point median and median absolute deviation (MAD) estimates for the samples in the ith band using

$$\begin{aligned} m_i = \quad & \mathrm{median}(y_{i-q}, y_{i-q+1}, y_{i-q+2}, \ldots, y_i, \ldots, \\ & y_{i+q-2}, y_{i+q-1}, y_{i+q}), \end{aligned} \tag{4}$$

and

$$\mathrm{MAD} = \mathrm{median}(|y_{i-q} - m_i|, \ldots, |y_{i+q} - m_i|), \tag{5}$$

respectively. The standard deviation (denoted as $\sigma_{i,MAD}$) is estimated by scaling the local median absolute deviation (MAD) by a factor of $\kappa = 1/(\sqrt{2}\mathrm{erf}^{-1}(1/2)) \approx 1.4826$:

$$\sigma_{i,MAD} = \kappa\mathrm{MAD}. \tag{6}$$

The Hampel filtering is a variation of the three-sigma rule of statistics, which is robust against outliers. If a sample y_i is such that

$$|y_i - m_i| > n_\sigma \sigma_{i,MAD} \tag{7}$$

for a given threshold n_σ, then the Hampel filtering declares y_i an outlier and replaces it with m_i; that is,

$$\tilde{y}_i = \begin{cases} m_i & \text{if } |y_i - m_i| > n_\sigma \sigma_{i,MAD}. \\ y_i & \text{otherwise.} \end{cases} \tag{8}$$

Here, we simply set n_σ to 3.

The coarse image $\widetilde{\mathcal{Y}}$ is obtained by applying the Hampel filtering (8) to each spectral vector of the observed image. We remark that although the Hampel filtering can detect the positions of impulse noise and stripes well, it may not produce exact spectral estimates for the polluted pixels. Thus, the goal of performing the Hampel filtering on the observed image is to obtain a coarse image for subspace learning.

2.3.2. Subspace Identification

Given the coarse image $\widetilde{\mathbf{Y}}$ (mainly containing Gaussian noise, not sparse noise), the subspace basis can be directly obtained by performing SVD on $\widetilde{\mathcal{Y}}$, as $\widetilde{\mathcal{Y}}$ mainly contains Gaussian noise which is independent and identically distributed (i.i.d.). That is, the signal subspace is approximately spanned by the first p left-singular vectors of $\widetilde{\mathbf{Y}}$:

$$\widehat{\mathbf{E}} = \widetilde{\mathbf{U}}(:, 1:p), \tag{9}$$

where $\widetilde{\mathbf{U}} \in \mathbb{R}^{n_b \times n_b}$ is an orthogonal matrix and $\{\widetilde{\mathbf{U}}, \widetilde{\mathbf{\Sigma}}, \widetilde{\mathbf{V}}\} = \text{SVD}(\widetilde{\mathbf{Y}}_{(3)})$ with singular values in $\widetilde{\mathbf{\Sigma}}$ ordered by non-increasing magnitude. The dimension of subspace, p, can be automatically estimated by the HySime method [37].

2.4. Fast Eigenimage Learning

2.4.1. Objective Function

Given the estimate of spectral subspace and the noise assumption (i.e., a mixture of i.i.d. Gaussian noise and sparse noise), we estimate the unknown variables, $\mathcal{Z}$ and $\mathcal{S}$, in (3) by solving

$$\{\widehat{\mathcal{Z}}, \widehat{\mathcal{S}}\} \in \arg\min_{\mathcal{Z},\mathcal{S}} \frac{1}{2}||\mathcal{Y} - \mathcal{Z} \times_3 \widehat{\mathbf{E}} - \mathcal{S}||_F^2 + \lambda_1 \phi(\mathcal{Z}) + \lambda_2 ||\mathcal{S}||_1. \tag{10}$$

The first term on the above optimization problem is derived from the observation model (3). The third term is a regularizer with respect to matrix $\mathcal{S}$. The ℓ_1 norm of $\mathcal{S}$, given by $\|\mathcal{S}\|_1 = \sum_{i=1}^{r} \sum_{j=1}^{c} \sum_{k=1}^{n_b} |\mathcal{S}(i,j,k)|$, promotes the sparsity of $\mathcal{S}$. Finally, $\lambda_1, \lambda_2 \geq 0$ are regularization parameters that trade off the importance of the respective regularizers. If the ϕ is a convex function, then the optimization (10) is a convex problem.

The second term in the optimization problem (10) is a regularization expressing prior information tailored to spatially correlated eigenimages [8,17]. We use the plug-and-play technique [29,30] to assign a prior for eigenimages; rather than investing efforts in designing new regularization for eigenimages, we can plug an image prior extracted from an off-the-shelf denoiser conceived for natural images. A deep image prior extracted from a CNN-based denoising network, FFDnet, is employed, and thus an explicit definition of the function $\phi(\cdot)$ is not given in optimization (10). More detailed analysis with regard to the deep prior $\phi(\cdot)$ can be found in Section 2.4.3.

Let $\mathcal{A}$ =concat$([\mathcal{Z}, \mathcal{S}], 3)$ be a $r \times c \times (p + n_b)$ tensor that concatenates the eigenimages $\mathcal{Z}$ and the sparse noise $\mathcal{S}$ along the third dimension. The optimization problem (10) can be rewritten as

$$\begin{aligned} \widehat{\mathcal{A}} \quad &\in \arg\min_{\mathcal{A}} \tfrac{1}{2}\|\mathcal{Y} - \mathcal{A} \times_3 [\widehat{\mathbf{E}}, \mathbf{I}_{n_b}]\|_F^2 + \\ &\lambda_1 \phi(\mathcal{A} \times_3 [\mathbf{I}_p, \mathbf{0}_{(p \times n_b)}]) + \lambda_2 \|\mathcal{A} \times_3 [\mathbf{0}_{(n_b \times p)}, \mathbf{I}_{n_b}]\|_1, \end{aligned} \tag{11}$$

where $\mathbf{I}_a$ denotes an identity matrix of size a and $\mathbf{0}_{(a\times b)}$ is a zero matrix of size $a \times b$.

2.4.2. Solver

The optimization problem (11) can be solved by the constrained-split augmented Lagrangian shrinkage algorithm (C-SALSA) [38,39]. C-SALSA is an instance of an alternating direction method of multipliers (ADMM) [40] that was developed to solve the constrained optimization formulation of regularized image restoration. By using variable splitting, we can convert the original optimization into a constrained one:

$$\begin{aligned} \min_{\mathcal{A},\mathcal{V}_1,\mathcal{V}_2,\mathcal{V}_3} \quad & \tfrac{1}{2}\|\mathcal{Y}-\mathcal{V}_1\|_F^2 + \lambda_1\phi(\mathcal{V}_2) + \lambda_2\|\mathcal{V}_3\|_1 \\ \text{s.t.} \quad & \mathcal{V}_1 = \mathcal{A}\times_3[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}] \\ & \mathcal{V}_2 = \mathcal{A}\times_3[\mathbf{I}_p, \mathbf{0}_{(p\times n_b)}] \\ & \mathcal{V}_3 = \mathcal{A}\times_3[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}]. \end{aligned}$$

The augmented Lagrangian function of the above optimization is

$$\begin{aligned} L(\mathcal{A},\mathcal{V}_1,\mathcal{V}_2,\mathcal{V}_3,\mathcal{D}_1,\mathcal{D}_2,\mathcal{D}_3) = & \\ & \tfrac{1}{2}\|\mathcal{Y}-\mathcal{V}_1\|_F^2 + \lambda_1\phi(\mathcal{V}_2) + \lambda_2\|\mathcal{V}_3\|_1 \\ & + \tfrac{\mu_1}{2}\|\mathcal{V}_1 - \mathcal{A}\times_3[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}] - \mathcal{D}_1\|_F^2 \\ & + \tfrac{\mu_2}{2}\|\mathcal{V}_2 - \mathcal{A}\times_3[\mathbf{I}_p, \mathbf{0}_{(p\times n_b)}] - \mathcal{D}_2\|_F^2 \\ & + \tfrac{\mu_3}{2}\|\mathcal{V}_3 - \mathcal{A}\times_3[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}] - \mathcal{D}_3\|_F^2, \\ & + \text{constant} \end{aligned} \tag{12}$$

where $\mu_1, \mu_2, \mu_3 > 0$ are C-SALSA penalty parameters.

The application of C-SALSA to (12) leads to Algorithm 1. A MATLAB demo of the proposed HySuDeep method will available at https://github.com/LinaZhuang (accessed on Nov. 1, 2021).

Algorithm 1 HySuDeep for mixed noise containing i.i.d. Gaussian noise

1: Set $k = 0$, choose $\mu_1, \mu_2, \mu_3 > 0$, $\mathcal{V}_{1,0}$, $\mathcal{V}_{2,0}$, $\mathcal{D}_{1,0}$, $\mathcal{D}_{2,0}$.
2: **repeat**
3: $\mathcal{A}_{k+1} = \arg\min_{\mathcal{A}} \frac{\mu_1}{2}\|\mathcal{V}_{1,k} - \mathcal{A}\times_3[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}] - \mathcal{D}_{1,k}\|_F^2 + \frac{\mu_2}{2}\|\mathcal{V}_{2,k} - \mathcal{A}\times_3[\mathbf{I}_p, \mathbf{0}_{(p\times n_b)}] - \mathcal{D}_{2,k}\|_F^2 + \frac{\mu_3}{2}\|\mathcal{V}_{3,k} - \mathcal{A}\times_3[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}] - \mathcal{D}_{3,k}\|_F^2$
4: $\mathcal{V}_{1,k+1} = \arg\min_{\mathcal{V}_1} \frac{1}{2}\|\mathcal{Y} - \mathcal{V}_1\|_F^2 + \frac{\mu_1}{2}\|\mathcal{V}_1 - \mathcal{A}_{k+1}\times_3[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}] - \mathcal{D}_{1,k}\|_F^2$
5: $\mathcal{V}_{2,k+1} = \arg\min_{\mathcal{V}_2} \lambda_1\phi(\mathcal{V}_2) + \frac{\mu_2}{2}\|\mathcal{V}_2 - \mathcal{A}_{k+1}\times_3[\mathbf{I}_p, \mathbf{0}_{(p\times n_b)}] - \mathcal{D}_{2,k}\|_F^2$
6: $\mathcal{V}_{3,k+1} = \arg\min_{\mathcal{V}_3} \lambda_2\|\mathcal{V}_3^T\|_1 + \frac{\mu_3}{2}\|\mathcal{V}_3 - \mathcal{A}_{k+1}\times_3[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}] - \mathcal{D}_{3,k}\|_F^2$
7: $\mathcal{D}_{1,k+1} = \mathcal{D}_{1,k} - (\mathcal{V}_{1,k+1} - \mathcal{A}_{k+1}\times_3[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}])$
8: $\mathcal{D}_{2,k+1} = \mathcal{D}_{2,k} - (\mathcal{V}_{2,k+1} - \mathcal{A}_{k+1}\times_3[\mathbf{I}_p, \mathbf{0}_{(p\times n_b)}])$
9: $\mathcal{D}_{3,k+1} = \mathcal{D}_{3,k} - (\mathcal{V}_{3,k+1} - \mathcal{A}_{k+1}\times_3[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}])$
10: $k \to k+1$
11: **until** stopping criterion is satisfied.

The optimizations with regard to $\mathcal{A}$ and $\mathcal{V}_1$ (on lines 3 and 4 of Algorithm 1) are quadratic problems, whose solutions are given by

$$\begin{aligned} & \mathcal{A}_{k+1} \\ = & \Big(\mu_1(\mathcal{V}_1 - \mathcal{D}_1)\times_3[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}]^T + \mu_2(\mathcal{V}_2 - \mathcal{D}_2)\times_3 \\ & [\mathbf{I}_p, \mathbf{0}_{p\times n_b}]^T + \mu_3(\mathcal{V}_3 - \mathcal{D}_3)\times_3[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}]^T\Big)\times_3 \\ & \Big(\mu_1[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}]^T[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}] + \mu_2[\mathbf{I}_p, \mathbf{0}_{p\times n_b}]^T \\ & [\mathbf{I}_p, \mathbf{0}_{p\times n_b}] + \mu_3[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}]^T[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}]\Big)^{-1} \end{aligned} \tag{13}$$

(for line 3) and

$$\mathcal{V}_{1,k+1} = (1+\mu_1)^{-1}\left[\mathcal{Y} + \mu_1(\mathcal{A}_{k+1}\times_3[\widehat{\mathbf{E}}, \mathbf{I}_{n_b}] + \mathcal{D}_{1,k})\right]. \tag{14}$$

(for line 4).

Line 6 of Algorithm 1 is a proximity operator of the ℓ_1 norm applied to $\mathcal{V}'_{3,k} = \mathcal{A}_{k+1}\times_3[\mathbf{0}_{(n_b\times p)}, \mathbf{I}_{n_b}] + \mathcal{D}_{3,k}$. It can be solved by an element-wise soft-threshold function; that is,

$$\begin{aligned} \mathcal{V}_{3,k+1} = & \arg\min_{\mathcal{V}_3} \lambda_2\|\mathcal{V}_3\|_1 + \frac{\mu_3}{2}\|\mathcal{V}_3 - \mathcal{V}'_{3,k}\|_F^2 \\ = & [\text{thresh}(\mathcal{V}'_{3,k}(i,j,k), \lambda/\mu_3),\ i = 1,\ldots,r, \\ & j = 1,\ldots,c, k = 1,\ldots,n_b], \end{aligned} \tag{15}$$

and threshold (x, τ) returns the soft thresholding of x (where τ is the threshold value) [41]:

$$x \mapsto \frac{\max(|x| - \tau, 0)}{\max(|x| - \tau, 0) + \tau}x.$$

Line 5 of Algorithm 1 is a proximity operator of ϕ applied to $\mathcal{V}'_{2,k} = \mathcal{A}_{k+1}\times_3[\mathbf{I}_p, \mathbf{0}_{(p\times n_b)}] + \mathcal{D}_{2,k}$. As the scaled Lagrange multiplier $\mathcal{D}_{2,k}$ tends to 0 as the C-SALSA iteration goes on, $\mathcal{V}'_{2,k}$ tends to be close to $\mathcal{Z}_{k+1}$. Considering that orthogonal projection is a decorrelation transformation and mode-3 slices of $\mathcal{Z}$ tend to be decorrelated, it is reasonable to decouple $\phi(\cdot)$ with respect to the mode-3 slices; that is,

$$\phi(\mathcal{V}'_{2,k}) = \sum_{i=1}^{p}\phi_i\Big(\mathcal{V}'_{2,k}(:,:,i)\Big), \tag{16}$$

where $\mathcal{V}'_{2,k}(:,:,i)$ denotes the ith mode-3 slice. The solution of the subproblem on line 5 can be decoupled with regard to mode-3 slices of $\mathcal{V}'_{2,k}$ and can be written as

$$\mathcal{V}_{2,k+1} = \Psi_{\lambda_1\phi/\mu_2}(\mathcal{V}'_{2,k}) = \begin{bmatrix} \Psi_{\lambda_1\phi_1/\mu_2}\Big(\mathcal{V}'_{2,k}(:,:,1)\Big) \\ \vdots \\ \Psi_{\lambda_1\phi_p/\mu_2}\Big(\mathcal{V}'_{2,k}(:,:,p)\Big) \end{bmatrix} \tag{17}$$

where

$$\Psi_{\lambda\phi_i}(\mathbf{U}) = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\|_F^2 + \lambda\phi_i(\mathbf{X}) \tag{18}$$

is the so-called Moreau proximity operator (MPO) [42] of ϕ or the denoising operator.

2.4.3. Plug-and-Play Prior, $\phi(\cdot)$

Instead of investing great effort in conceiving a new eigenimage regularization $\phi_i(\mathbf{X})$, we make use of the plug-and-play (PnP) prior [8,29,30]. The PnP tool is a powerful framework that solves an inverse image problem by directly using an existing regularizer from a state-of-the-art image denoiser. The PnP framework has been applied to hyperspectral image fusion, spectral unmixing, hyperspectral image inpainting, and anomaly detection, and produces surprisingly good image recovery results. Inspired by these successful applications of PnP, we plug the prior of a deep denoising network, FFDnet [24], into (18), leading to

$$\widehat{\mathcal{V}}_{2,k}(:,:,i) \leftarrow \text{FFDNet}(\mathcal{V}'_{2,k}(:,:,i)), \tag{19}$$

where FFDNet$(\cdot)$ represents the network and outputs a denoised image. It is worth noting that other state-of-the-art image denoisers can also be plugged in (18). We choose FFDNet in this work based on the following considerations: (a) deep-learning-based FFDNet is much faster than other machine learning-based denoisers as long as it has been well trained and (b) FFDNet is flexible and able to address images with various noise levels without retraining the network.

Since the denoiser FFDNet, plugged into the subproblem with regard to $\mathcal{V}_2$, is not a proximity operator, we do not have a theoretical convergence guaranteed for the implemented variant of C-SALSA. However, impressive performances using plugged non-convex regularizations have been observed in a number of inverse problems [19,43,44]. The numerical convergence of the proposed HySuDeep is analyzed in Section 4.5.

2.5. HSI Recovery

After estimating $\widehat{\mathcal{Z}}$, the denoised image is obtained as follows:

$$\widehat{\mathcal{X}} = \widehat{\mathcal{Z}} \times_3 \widehat{\mathbf{E}}. \tag{20}$$

3. Model Extension to Non-i.i.d. Gaussian Noise

Gaussian noise in the observation model (1) is assumed to be i.i.d., meaning that the model is simplified and we can focus on the discussion of the image structure. Let $\mathbf{C}_\lambda = \mathbb{E}[\mathbf{n}_i \mathbf{n}_i^T]$ (where $\mathbf{n}_i$ is a generic column of unfolded mode-3 matrix $\mathbf{N}_{(3)}$) define the covariance matrix of the spectral noise. We have $\mathbf{C}_\lambda = \sigma^2 \mathbf{I}_{n_b}$ when the noise is i.i.d.

However, we remark that Gaussian noise in real HSIs tends to be non-i.i.d.; that is, it is pixel-wise independent but band-wise dependent. Before implementing denoising according to model (1), we need to convert the non-i.i.d. scenario into an i.i.d. scenario by implementing data whitening:

$$\bar{\mathcal{Y}} = \mathcal{Y} \times_3 \sqrt{\mathbf{C}_\lambda^{-1}}, \tag{21}$$

where $\sqrt{\mathbf{C}_\lambda}$ and $\mathbf{C}_\lambda^{-1}$ denote the square root of $\mathbf{C}_\lambda$ and the inverse of $\mathbf{C}_\lambda$, respectively.

The estimate of the covariance matrix, $\mathbf{C}_\lambda$, is challenging when noise in observations is a mixture of Gaussian and sparse noise. If the HSI is only corrupted by Gaussian noise, then the noise covariance matrix can be estimated by exploiting the spectral correlation of HSI. The spectrally uncorrelated components extracted from the HSI are considered as Gaussian noise, whose covariance matrix can be computed easily. This problem has been studied deeply, and we refer readers to representative works [37,45]. However, when the noise in the image is a mixture of Gaussian noise and sparse noise, the spectrally uncorrelated components extracted from the HSI are mixed noise, instead of single Gaussian noise. To solve this problem, we propose a method to split two kinds of noise and estimate $\mathbf{C}_\lambda$ in two steps, as listed below:

- Estimate a coarse image, $\widetilde{\mathcal{Y}}$, by removing the sparse noise from observation. A outlier removal step applying Hampel filtering to spectral vectors of the observed image is given in (8).

- We apply linear regression to each band of the image $\widetilde{\mathcal{Y}}$; i.e., each band is represented as a linear combination of the remaining bands [37]. That is,

$$[\widetilde{\mathbf{Y}}_{(3)}^T]_{:,i} = [\widetilde{\mathbf{Y}}_{(3)}^T]_{\alpha_i}\boldsymbol{\beta}_i + \boldsymbol{\xi}_i, \tag{22}$$

where $\widetilde{\mathcal{Y}}_{(3)}^T$ denotes the transpose of the mode-3 unfolding matrix $\widetilde{\mathbf{Y}}_{(3)}$, the subscript $[\cdot]_{:,i}$ means extracting the ith column from a matrix, a matrix with the subscript $[\cdot]_{\alpha_i}$ means the matrix including all columns except ith column, $\boldsymbol{\beta}_i \in \mathbb{R}^{n_b-1}$ denotes regression coefficients, and $\hat{\boldsymbol{\xi}}_i \in \mathbb{R}^n$ denotes regression error.
The regression coefficients $\boldsymbol{\beta}_i$ can be estimated by the least squares method; i.e.,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|[\widetilde{\mathbf{Y}}_{(3)}^T]_{:,i} - [\widetilde{\mathbf{Y}}_{(3)}^T]_{\alpha_i}\boldsymbol{\beta}_i\|_F^2. \tag{23}$$

Given $\hat{\boldsymbol{\beta}}$, the regression error, $\hat{\boldsymbol{\xi}}_i$, is computed by

$$\hat{\boldsymbol{\xi}}_i = [\widetilde{\mathbf{Y}}_{(3)}^T]_{:,i} - [\widetilde{\mathbf{Y}}_{(3)}^T]_{\alpha_i}\hat{\boldsymbol{\beta}}_i. \tag{24}$$

The regression errors are taken as a coarse estimate of the Gaussian noise; thus, its covariance matrix, $\mathbf{C}_\lambda = \mathrm{diag}([\sigma_1^2, \sigma_2^2, \ldots, \sigma_{n_b}^2])$, can be estimated by

$$\sigma_i^2 = \mathrm{var}(\hat{\boldsymbol{\xi}}_i), \tag{25}$$

where $\mathrm{var}(\cdot)$ is a function computing the variance of the elements of the input vector.

Given the estimate of $\mathbf{C}_\lambda$, we can convert band-dependent Gaussian noise to i.i.d. Gaussian noise via (21). Here, we analyze the impact of image conversion on Gaussian noise. Given the image conversion,

$$\bar{\mathcal{Y}} = \mathcal{X} \times_3 \sqrt{\mathbf{C}_\lambda^{-1}} + \mathcal{S} \times_3 \sqrt{\mathbf{C}_\lambda^{-1}} + \mathcal{G} \times_3 \sqrt{\mathbf{C}_\lambda^{-1}}, \tag{26}$$

we can compute the spectral covariance of the Gaussian noise in the converted image as follows:

$$\begin{aligned} \bar{\mathbf{C}}_\lambda &= \mathbb{E}[\sqrt{\mathbf{C}_\lambda^{-1}}\mathbf{n}_i(\sqrt{\mathbf{C}_\lambda^{-1}}\mathbf{n}_i)^T] = \sqrt{\mathbf{C}_\lambda^{-1}}\mathbb{E}[\mathbf{n}_i\mathbf{n}_i^T]\sqrt{\mathbf{C}_\lambda^{-1}}^T \\ &= \sqrt{\mathbf{C}_\lambda^{-1}}\mathbf{C}_\lambda\sqrt{\mathbf{C}_\lambda^{-1}}^T = \mathbf{I}. \end{aligned} \tag{27}$$

From (27), we can see that after image conversion, the Gaussian noise is i.i.d. and standard distributed, which meets the noise assumption in model (1); thus, the converted image, $\bar{\mathcal{Y}}$, can be denoised as discussed in Section 2. Finally, we reconstruct the clean image as follows:

$$\hat{\mathcal{X}} = \hat{\bar{\mathcal{X}}} \times_3 \sqrt{\mathbf{C}_\lambda}, \tag{28}$$

where $\hat{\bar{\mathcal{X}}}$ is the estimated clean version of image $\bar{\mathcal{Y}}$.

The pseudocode in Algorithm 2 shows how HySuDeep is implemented to reduce mixed noise for an HSI. Given an HIS of size r (rows) $\times c$ (columns) $\times n_b$ (bands) with subspace dimension p ($p \ll n_b$), the computational complexity of obtaining a coarse image $\widetilde{\mathcal{Y}}$ in line 2 and $\mathbf{C}_\lambda$ in line 3 is $\mathcal{O}(r * c * n_b * q * \log q)$ and $\mathcal{O}(r * c * n_b^3)$, respectively. The Gaussian noise-whitening in line 4, its inverse transformation in line 8, and the image reconstruction step in line 7 have the same computational complexity; that is, $\mathcal{O}(r \times c \times n_b^2)$. The estimation of the spectral subspace in line 5 and eigenimages in line 6, respectively, cost $\mathcal{O}(r^2 * c^2 * n_b)$ and $\mathcal{O}(r * c * (p + n_b)^2 + (p + n_b)^3 + p * d)$. Here, d represents the computational complexity of denoising an eigenimage using FFDNet. Consequently, the overall computational complexity of HySuDeep is $\mathcal{O}(r * c * n_b^3 + r^2 * c^2 * n_b + p * d)$.

Algorithm 2 HySuDeep for mixed noise containing non-i.i.d. Gaussian noise

1: Input a noisy HSI, $\mathcal{Y}$, and parameter λ_2.
2: Estimate a coarse image $\widetilde{\mathcal{Y}}$ via (8).
3: Estimate $\mathbf{C}_\lambda$ from $\widetilde{\mathcal{Y}}$ via (25).
4: Obtain an whitened image $\bar{\mathcal{Y}}$ using (21).
5: Denoise the whitened image $\bar{\mathcal{Y}}$ using Algorithm 1, and obtain $\widehat{\bar{\mathcal{X}}}$.
6: Compute $\widehat{\mathcal{X}} = \widehat{\bar{\mathcal{X}}} \times_3 \sqrt{\mathbf{C}_\lambda}$, an estimate of the clean HSI.

4. Experiments with Simulated Images

Experiments were carried out on three simulated datasets to evaluate the performance of the proposed HySuDeep method compared with six state-of-the-art denoising methods for HSI mixed noise removal.

4.1. Simulation of Noisy Images and Comparisons

4.1.1. Simulation of Noisy Images

Three public hyperspectral datasets (shown in Figure 4a–c) were employed to simulate the noisy images. Following the same procedure in [32], we generated noisy HSIs of Pavia University data (of size 310(rows) $\times$ 250(columns) $\times$ 87(bands)), a subregion of Washington DC Mall data (of size 150(rows) $\times$ 200(columns) $\times$ 191(bands)), and Terrain data (of size 500(rows) $\times$ 307(columns) $\times$ 166(bands)).

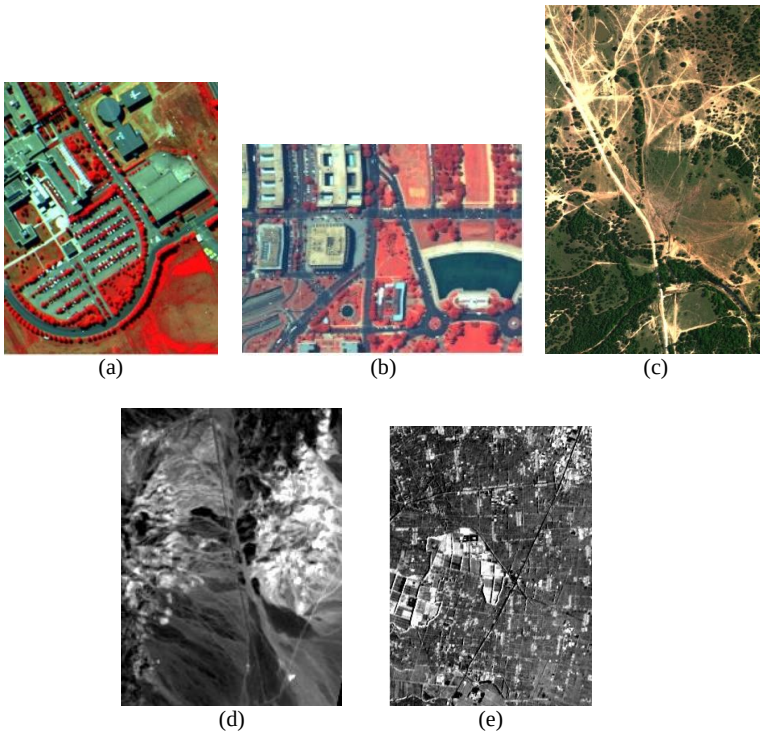

Figure 4. HSIs used in the experiments. (**a**) Pavia University data. (**b**) Washington DC Mall data. (**c**) Terrain data. (**d**) Hyperion Cuprite. (**e**) Tiangong-1.

For the first two datasets, we removed bands that were severely corrupted by water vapor in the atmosphere. To obtain relatively clean images, we projected spectral vectors of each image onto a subspace spanned by eight principal eigenvectors of each image. The projection of each image was considered to be a clean image. To simulate noisy HSIs, we added four kinds of additive noise into images as follows:

Case 1 (Gaussian non-i.i.d. noise): Synthetic data with Gaussian noise. The noise in ith pixel vector $\mathbf{g}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{D}^2)$, where $\mathbf{D}$ is a diagonal matrix with diagonal elements sampled from a uniform distribution $U(0, 0.01)$.

Case 2 (Gaussian noise + stripes): Synthetic data with Gaussian noise (described in case 1) and oblique stripe noise randomly affecting 30% of the bands and, for each band, about 10% of the pixels, at random.

Case 3 (Gaussian noise + "Salt and Pepper" noise): Synthetic data with Gaussian noise (described in case 1) and "Salt and Pepper" noise with a noise density of 0.5%, meaning approximately 0.5% of elements in $\mathcal{X}$ are affected.

Case 4 (Gaussian noise + stripes + "Salt and Pepper" noise): Synthetic data with Gaussian noise (described in case 1), random oblique stripes (described in case 2) and "Salt and Pepper" noise (described in case 3).

To evaluate the impact of denoising methods on the hyperspectral unmixing task, we simulated a clean semi-real HSI based on the publically available Terrain image (Figure 4c) following generation steps in [36]. A MATLAB demo used to generate a semi-real HSI is available at https://github.com/LinaZhuang/NMF-QMV_demo.The original Terrain image has a size of 500(rows) $\times$ 307(columns) $\times$ 166(bands) and is mainly composed of soil, tree, grass, and shadow. The number of endmembers is empirically set to 5, as performed in [36,46,47]. Briefly, a clean Terrain image was synthesized based on a linear mixing model [1]; i.e., $\mathcal{X} = \mathcal{A} \times_3 \mathbf{M}$, where $\mathbf{M}$ and $\mathcal{A}$ are endmembers and abundances, respectively, estimated from the original Terrain image. Next, we generated a noisy Terrain HSI by adding the Gaussian noise and oblique stripe noise (as described in case 2)—i.e., $\mathcal{Y} = \mathcal{A} \times_3 \mathbf{M} + \mathcal{S} + \mathcal{G}$—yielding an MPSNR of 28.90 dB.

4.1.2. Comparisons

To thoroughly evaluate the performance of the proposed method, six state-of-the-art HSI denoising methods were selected for comparison: the fast hyperspectral denoising method (FastHyDe) [8], the noise-adjusted iterative low-rank matrix approximation method (NAIRLMA) [48], the spatio-spectral total variation based method (SSTV) [49], the low-rank matrix recovery method (LRMR) [15], double-factor-regularized low-rank tensor factorization (LRTF-DFR) [21], and the ℓ_1-norm based hyperspectral mixed noise denoising method (L1HyMixDe) [32]. These compared methods were carefully selected. The FastHyDe served as a benchmark to see whether mixed noise could be simply addressed by Gaussian denoisers. The NAIRLMA, SSTV, and LRMR are from highly cited papers working on hyperspectral mixed noise based on low-rank and sparse representations. The LRTF-DFR and L1HyMixDe methods are subspace representation methods and remove noise by filtering the subspace coefficients of HSIs. All experiments were implemented in MATLAB (R2016a) on Windows 10 with an Intel Core i7-7700HQ 2.8-GHz processor and 24 GB RAM. A MATLAB demo of this work will be available at https://github.com/LinaZhuang for the sake of reproducibility.

The size of the sliding window in the Hampel filter, q, was fixed to 7. The dimension of the subspace, p, for FastHyDe, LRTF-DFR, L1HyMixDe, and HySuDeep methods was estimated by HySime [37] automatically. The other parameters of FastHyDe, NAILRMA, LRTF-DFR, and L1HyMixDe were set as suggested in their original references. We fine-tuned the parameters λ and μ in SSTV and parameters r and s in LRMR to obtain the optimal results.

For quantitative assessment, the peak signal-to-noise ratio (PSNR) index, the structural similarity (SSIM) index, and the feature similarity (FSIM) index of each band were calculated. The mean PSNR (MPSNR), mean SSIM (MSSIM), and mean FSIM (MFSIM) of the proposed and compared methods on Pavia University Data and Washington DC Mall Data are presented in Table 2, where we have highlighted the best results in bold.

Table 2. Performance of the proposed and comparison methods on Pavia University Data and Washington DC Mall Data.

Cases	Indexes	Noisy	FastHyDe [8]	NAILRMA [48]	SSTV [49]	LRMR [15]	LRTF-DFR [21]	L1HyMixDe [32]	HySuDeep
				Pavia University data					
Case 1	**MPSNR**	34.29	**50.50**	47.25	45.73	43.72	43.77	47.62	49.08
	MSSIM	0.8221	**0.9981**	0.9949	0.9929	0.9867	0.9906	0.9971	0.9978
	MFSIM	0.9272	**0.9991**	0.9977	0.9966	0.9945	0.9949	0.9986	0.999
	Time (s)	-	**10**	185	282	107	141	114	31
Case 2	**MPSNR**	27.77	40.94	32.33	35.06	33.58	43.43	47.26	**48.55**
	MSSIM	0.7451	0.9087	0.9236	0.9584	0.9176	0.9902	0.9969	**0.9975**
	MFSIM	0.8683	0.9569	0.9519	0.9702	0.9461	0.9948	0.9985	**0.9989**
	Time (s)	-	**9**	182	268	67	141	156	33
Case 3	**MPSNR**	22.56	39.66	37.03	43.85	43.37	43.72	47.90	**48.93**
	MSSIM	0.6892	0.9781	0.9556	0.9876	0.9858	0.9905	0.9971	**0.9977**
	MFSIM	0.8780	0.9870	0.9799	0.9948	0.9942	0.9948	0.9986	**0.999**
	Time (s)	-	**9**	123	250	87	141	112	32
Case 4	**MPSNR**	19.29	31.83	28.20	33.59	33.64	43.32	47.43	**48.31**
	MSSIM	0.6258	0.8249	0.8531	0.9463	0.9153	0.9901	0.9968	**0.9972**
	MFSIM	0.8267	0.9103	0.9169	0.9656	0.9455	0.9947	0.9985	**0.9987**
	Time (s)	-	**10**	193	274	89	140	114	38
				Washington DC Mall data					
Case 1	**MPSNR**	19.13	**42.04**	37.22	26.78	27.15	36.96	39.52	41.17
	MSSIM	0.7781	**0.9985**	0.9962	0.9674	0.9710	0.9961	0.9981	0.9983
	MFSIM	0.8634	**0.9980**	0.9931	0.9630	0.9706	0.9959	0.9978	0.999
	Time (s)	-	**6**	218	228	55	145	40	29
Case 2	**MPSNR**	12.52	30.58	20.56	24.08	25.39	34.12	36.65	**40.05**
	MSSIM	0.6851	0.8608	0.8440	0.9512	0.9636	0.9909	0.9946	**0.9978**
	MFSIM	0.7973	0.9368	0.8940	0.9547	0.9663	0.9948	0.9956	**0.9987**
	Time (s)	-	**5**	196	234	37	145	64	29
Case 3	**MPSNR**	7.29	29.78	24.94	26.70	27.29	36.62	39.90	**40.75**
	MSSIM	0.6421	0.9818	0.9642	0.9667	0.9707	0.9959	0.9982	**0.9981**
	MFSIM	0.8052	0.9849	0.9713	0.9625	0.9706	0.9958	0.9978	**0.9988**
	Time (s)	-	**5**	137	229	40	144	42	30
Case 4	**MPSNR**	3.84	20.34	15.82	24.03	24.94	34.03	36.81	**39.07**
	MSSIM	0.5676	0.8029	0.7958	0.9506	0.9617	0.9910	0.9947	**0.997**
	MFSIM	0.7508	0.8886	0.8663	0.9544	0.9649	0.9947	0.9956	**0.998**
	Time (s)	-	**5**	178	207	43	144	52	30

4.2. Mixed Noise Removal

We compare the proposed method HySuDeep with FastHyDe [8], NAILRMA [48], SSTV [49], LRMR [15], LRTF-DFR [21], and L1HyMixDe [32] methods in terms of MPSNR, MSSIM, and MFSIM (provided in Tables 2 and 4). The band-wise PSNR is depicted in Figure 5 for quantitative assessment.

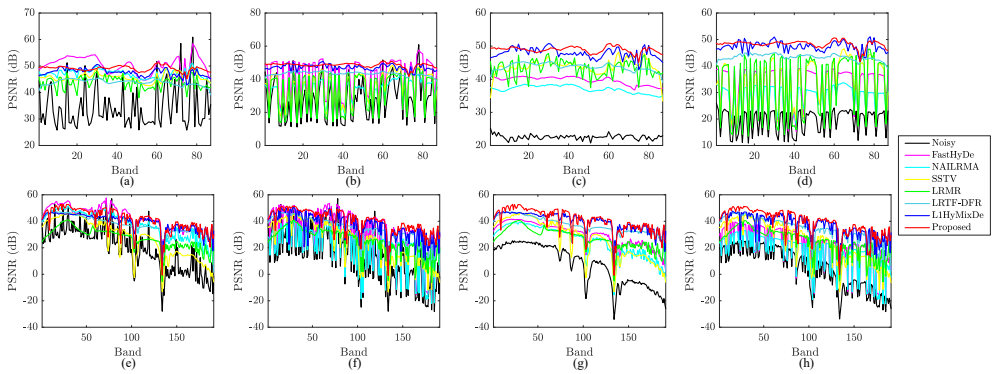

Figure 5. Band-wise PSNR values for denoised Pavia University data in the first row and for denoised Washington DC Mall data in the second row. Subfigures in (**a**,**e**), (**b**,**f**), (**c**,**g**), (**d**,**h**) corresponds to case 1, case 2, case 3, and case 4, respectively.

Among the compared methods, only FastHyDe is shown to address pure Gaussian noise, not mixed noise. We include FastHyDe to evaluate whether mixed noise can be

reduced by a simple Gaussian denoiser. As shown in Table 2, FastHyDe outperformed other methods in case 1 (where HSIs are corrupted by only Gaussian noise). However, when the images were contaminated by mixed noise in Cases 2–4, the results of FastHyDe show that heavy mixed noise cannot be reduced well by a Gaussian denoiser. The existence of mixed noise in HSIs calls for effective denoising techniques.

To address mixed noise, an additive term is usually introduced to model sparse noise so that the entries in the data fidelity term, $(\mathcal{Y} - \mathcal{X} - \mathcal{S})$, follow Gaussian distributions. The proposed HySuDeep, NAIRLMA, SSTV, LRMR, and LRTF-DFR methods fall in this line of research. The critical differences among these methods are the regularizations imposed on the HSI, $\mathcal{X}$. SSTV minimizes the total variation of the HSI in the spectral–spatial domain. NAILRMR and LRMR impose spectral low-rankness in spatial patches of the image. LRTF-DFR and the proposed HySuDeep both enforce the low-rankness of spectral vectors by subspace representation. To exploit the spatial correlation of eigenimages, LRTF-DFR minimizes the spatial difference (i.e., total variation) of eigenimages, while HySuDeep uses a CNN-regularization (i.e., a deep image prior extracted from a CNN network).

The proposed method uniformly yields the best results in Cases 2–4, as shown in Tables 2 and 4. The main reason is that, compared with SSTV, NAILRMR, LRMR, LRTF-DFR, and L1HyMixDe methods, our method uses a more efficient spatial regularization term. Although the CNN regularization works like a black box, we can see its superiority over TV regularization (used in SSTV and LRTF-DFR) and patch-based regularization (used in NAILRMA, LRMR, and L1HyMixDe) from Tables 2 and 4. Furthermore, HySuDeep is similar to the L1HyMixDe method in terms of using subspace representations of spectral vectors and imposing a regularization on representation coefficients. Tables 2 and 4 show that HySuDeep achieves better results than L1HyMixDe. The difference is caused by the accuracy of spectral subspace learning. Compared with L1HyMixDe, HySuDeep elaborates an outlier removal operator, which improves the estimates of subspace $\widehat{\mathbf{E}}$ (see discussion in Section 4.3).

For visual comparison, we display the 37th band of Pavia University data and the 126th band of Washington DC Mall data in Figures 6 and 7, respectively. For case 1 (Gaussian noise), all methods can reduce noise significantly. As shown in Figures 6 and 7, SSTV and LRMR are able to remove light stripes, but still leave some wide stripes. Heavy noise still remains in the results of NAILRMA and FastHyDe. LRTF-DFR, L1HyMixDe, and HySuDeep methods visually yield comparable results in Figures 6 and 7. To show their differences, we generated and presented the residual images in Figure 8a,b, where we can see the results of proposed HySuDeep contain less residual compared with LRTF-DFR and L1HyMixDe.

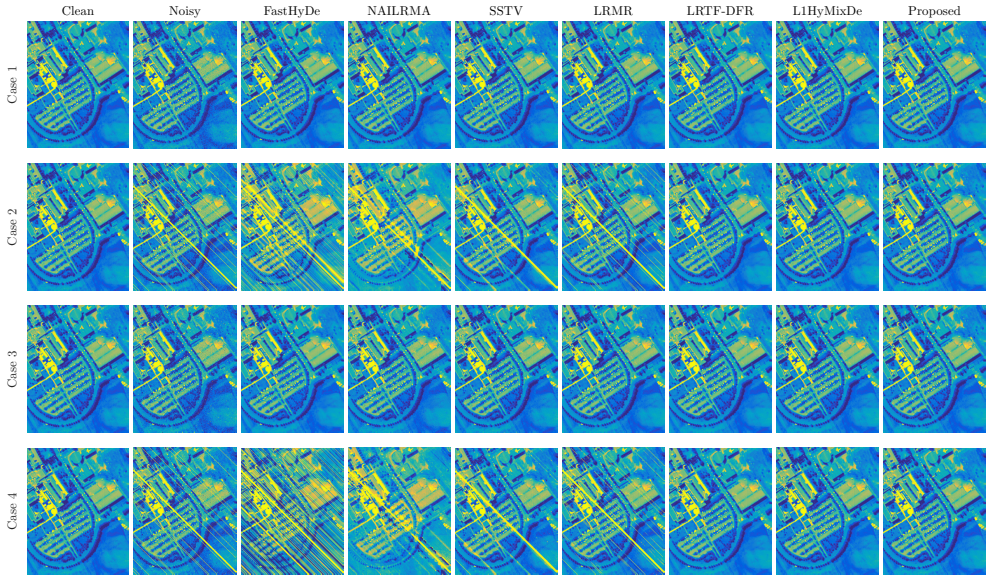

Figure 6. Band 37 of Pavia University data before and after denoising in four cases.

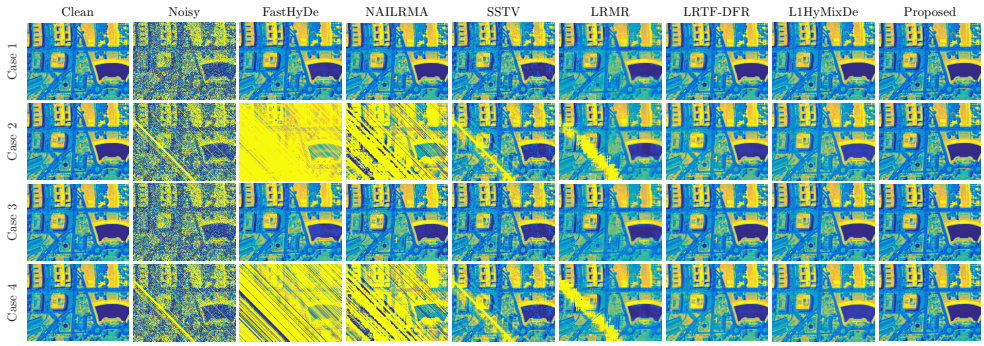

Figure 7. Band 126 of the Washington DC Mall data before and after denoising in four cases.

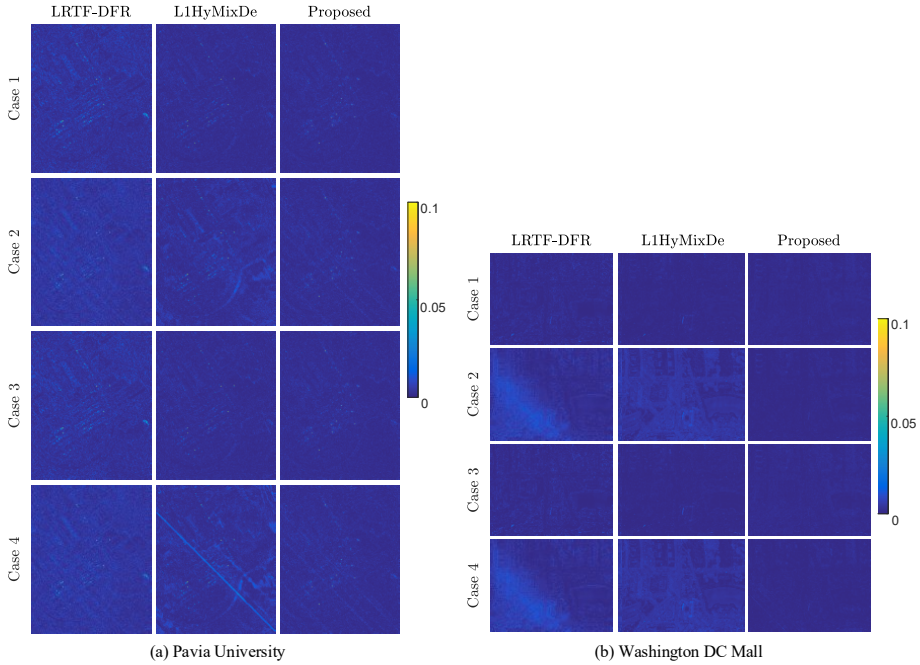

Figure 8. Differences between the clean images and denoised images in four cases for LRTF-DFR, L1HyMixDe, and the proposed method. (a) Pavia University data; (b) Washington DC Mall data.

4.3. Subspace Learning against Mixed Noise

One of the contributions of this paper is the introduction of an orthogonal subspace identification method that is robust to mixed noise. We conducted experiments using simulated images to compare the proposed subspace learning method with representative subspace learning methods: namely, SVD, RPCA, L1HyMixDe [32], and HySime [37]. Results are reported in Table 3, where we show two metrics, namely $\gamma_1 = \|\mathcal{X} \times_3 \widehat{\mathbf{E}}^T \times_3 \widehat{\mathbf{E}}\|_F^2 / \|\mathcal{X}\|_F^2$ and $\gamma_2 = \|\mathcal{N} \times_3 \widehat{\mathbf{E}}^T \times_3 \widehat{\mathbf{E}}\|_F^2 / \|\mathcal{N}\|_F^2$, measuring the relative power of the clean spectral vectors and noise lying in the estimated subspace, respectively.

In Case 1, two images were corrupted only by Gaussian noise, and values of γ_1 for all methods are higher than 0.9993, implying that all the methods can estimate a subspace representing image signal very well. Case 3 shows results similar to Case 1; that is, all the methods achieve relatively high γ_1. We conclude that when noise is Gaussian distributed or a mixture of Gaussian and "Salt and Pepper" noise, signal subspace learning is not a challenging task and we can simply use SVD. The reason is that Gaussian noise and "Salt and Pepper" noise are randomly and uniformly distributed in each channel; thus, the noise increases singular values in the direction of each eigenvector almost uniformly and does not change the order of the singular values of clean images. The signal subspace is approximately spanned by p singular vectors of the noisy image corresponding to the largest p singular values.

However, subspace learning from noisy images in Case 2 and Case 4 is challenging and we cannot simply use SVD because the stripe noise usually exist in specific channels, instead of being uniformly distributed in all channels. The stripe noise will significantly increase the variances of those channels affected by stripe noise. Due to the fact that SVD tends to learn a subspace representing information in channels with high variances, SVD is not suitable for Case 2 and Case 4. This inference is consistent with the results in Table 3, where SVD obtains the worst results in terms of γ_1 and γ_2 in case 2 and case 4. If we only focus on the table rows highlighted with gray color, the proposed learning method performs better than others. Among the compared methods, RPCA learns a subspace representing low-rank signal and excluding sparse noise. However, stripe noise is also low-rank [50], and RPCA cannot separate an image signal and stripes using only a low-rank regularization. HySime is conceived based on an assumption of Gaussian noise and is clearly not suitable for mixed noise. L1HyMixDe performs median filtering band by band,

exploiting spatial correlation, whereas HySuDeep estimates a coarse image using Hampel filtering pixel by pixel, exploiting spectral correlation. As shown in gray rows of Table 3, HySuDeep uniformly provides the best performances.

Table 3. Quantitative comparison of representation ability of subspaces learned by different methods.

		SVD [51]	RPCA [52]	L1HyMixDe [32]	HySime [37]	HySuDeep
Pavia University data						
Case 1	γ_1	0.9994	0.9993	0.9995	**1.0000**	0.9996
	γ_2	0.1789	0.1816	0.1753	**0.0954**	0.1500
Case 2	γ_1	0.9854	0.9868	0.9994	0.9881	**0.9996**
	γ_2	0.2552	0.2526	0.1599	0.2270	**0.1314**
Case 3	γ_1	0.9993	0.9991	0.9995	**0.9997**	0.9996
	γ_2	0.1095	0.1102	0.1066	**0.0909**	0.1019
Case 4	γ_1	0.9855	0.9867	0.9993	0.9869	**0.9997**
	γ_2	0.2355	0.2338	0.1526	0.2109	**0.1271**
Washington DC Mall data						
Case 1	γ_1	0.9998	0.9998	0.9998	**1.0000**	0.9998
	γ_2	0.0772	0.0775	0.0751	**0.0466**	0.0630
Case 2	γ_1	0.9826	0.9838	0.9997	0.9878	**0.9997**
	γ_2	0.2738	0.2700	0.0978	0.2426	**0.0585**
Case 3	γ_1	0.9997	0.9997	**0.9998**	**0.9998**	**0.9998**
	γ_2	0.0483	0.0480	0.0456	**0.0406**	0.0441
Case 4	γ_1	0.9825	0.9834	0.9997	0.9868	**0.9997**
	γ_2	0.2342	0.2321	0.0883	0.2110	**0.0600**

Note: $\gamma_1 = \frac{\|\mathcal{X}\times_3\widehat{\mathbf{E}}^T\times_3\widehat{\mathbf{E}}\|_F^2}{\|\mathcal{X}\|_F^2}$ and $\gamma_2 = \frac{\|\mathcal{N}\times_3\widehat{\mathbf{E}}^T\times_3\widehat{\mathbf{E}}\|_F^2}{\|\mathcal{N}\|_F^2}$.

4.4. Analysis of Regularization Parameters

There are two regularization parameters—namely, λ_1 and λ_2—in the objective function of the proposed HySuDeep method. Controlling the trade-off between Gaussian noise reduction and detail preservation, the parameter λ_1 is related to the standard deviation of Gaussian noise, which can be estimated via (25).

The second parameter λ_2 controls the sparsity of estimation of sparse noise $\mathcal{S}$. The setting of λ_2 should depend on the intensity of sparse noise. Figure 9 depicts the denoising performance of HySuDeep in the Pavia University image and in the Washington DC Mall image as a function of the value of parameter λ_2. When the value of λ_2 is set to $\{1, 3, 5, 7, 9\}$, denoising results are acceptable in both images. Therefore, we simply set λ_2 to 3 in all experiments in this paper.

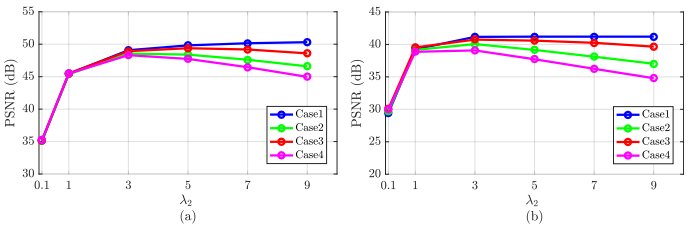

Figure 9. Denoising performance of HySuDeep as a function of the regularization parameter λ_2. (**a**) Pavia University image. (**b**) Washington DC Mall image.

4.5. Numerical Convergence of the HySuDeep

Since CNN-regularization is incorporated into (10), the proposed HySuDeep cannot be considered as a convex optimization problem; thus, its theoretical convergence is not guaranteed. However, its numerical convergence is systematically observed when the augmented Lagrangian parameters are set to $\mu_i = 1$ $(i = 1, 2, 3)$. As shown in Figure 10, the relative change converges to near zero after 15 iterations, implying the convergence of the proposed method can be numerically guaranteed.

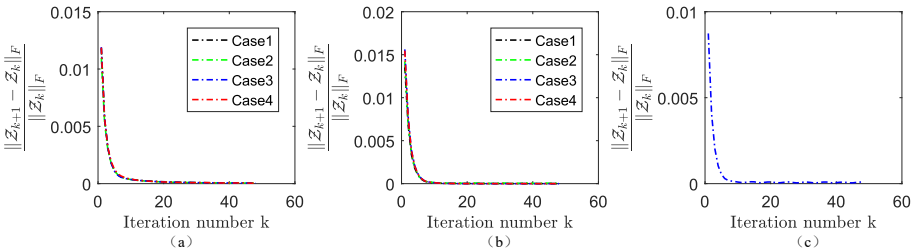

Figure 10. Relative change of $\mathcal{Z}$ as a function of the iteration number of the HySuDeep method applied to Pavia University data in (**a**), Washington DC Mall data in (**b**), and Terrain image in (**c**).

4.6. Application in Hyperspectral Unmixing

Hyperspectral denoising is usually implemented as a preprocessing step for subsequent applications. This subsection takes hyperspectral unmixing as an example to evaluate whether the image denoising step has a positive impact on the performance of subsequent applications. Spectral unmixing mainly involves two stages: (i) identifying materials present in the scene (termed endmembers) and (ii) estimating the fraction (or abundance) of each material in each pixel. The hyperspectral unmixing task was chosen considering that the endmember extraction step can be very sensitive to sparse noise.

We first performed denoising on the Terrain image using FastHyDe, NAILRMA, SSTV, LRMR, LRTF-DFR, L1HyMixDe, and HySuDeep. The mean PSNR (MPSNR), mean SSIM (MSSIM), and mean FSIM (MFSIM) of the proposed and compared methods on Terrain image are presented in Table 4, where we have highlighted the best results in bold. Then, we unmixed the spectra of the denoised images by using vertex component analysis (VCA) [53] to estimate the endmembers and used fully constrained least squares (FCLS) [54] to estimate the abundances. Two metrics were computed: the normalized mean square error (NMSE) of endmembers $\mathbf{A}$ and abundances $\mathbf{S}$, denoted as NMSE_A and NMSE_S, respectively. As reported in Table 4, LRMR, LRTF-DFR, L1HyMixDe, and HySuDeep obtain a lower NMSE of endmembers than the counterpart image without denoising processing. For images denoised by FastHyDe, NAILRMA, and SSTV, the errors in the estimates of endmembers directly result in high errors in the estimates of abundances. Among LRMR, LRTF-DFR, L1HyMixDe, and HySuDeep, the proposed HySuDeep leads to the lowest NMSE of abundances.

Table 4. Denoising performance of the proposed and comparison methods on the Terrain image and unmixing performance (in terms of $\text{NMSEA}_\mathbf{M}$ and $\text{NMSES}_\mathcal{A}$) of "VCA + FCLS" applied to the denoised images.

	Noisy	FastHyDe	NAILRMA	SSTV	LRMR	LRTF-DFR	L1HyMixDe	HySuDeep
MPSNR	28.90	48.94	40.72	35.58	39.84	45.84	50.84	**51.91**
MSSIM	0.7296	0.9933	0.9843	0.9168	0.9831	0.9932	0.9975	**0.9987**
MFSIM	0.8719	0.9961	0.9882	0.9548	0.9830	0.9953	0.9984	**0.9989**
Time (s)	-	**20**	587	951	129	572	562	**137**
$\textbf{NMSEA}_\mathbf{M}$	0.10	0.19	0.17	0.20	0.05	0.06	**0.03**	0.04
$\textbf{NMSES}_\mathcal{A}$	0.52	0.48	0.47	0.48	0.46	0.51	0.15	**0.09**

Note: $\text{NMSEA}_\mathbf{M}$ and $\text{NMSES}_\mathcal{A}$ denote root mean square error of endmembers and abundances, respectively.

5. Experimental Results for Real Images

We evaluate the performance of the proposed method on two real HSI datasets, as shown in Figure 4d,e.

5.1. Hyperion Cuprite Dataset

The Cuprite HSI was captured at Cuprite, NV, USA, by the Hyperion sensor, which divides the spectrum from 355 nm to 2577 nm into 242 channels with a spectral resolution of 10 nm. The spatial resolution of the image is 30 m. A subregion of size 240 × 178 pixels with 177 spectral channels (after removing very low SNR channels) was cropped for the test. Four bands are shown in Figure 11, where bands 93, 133, and 134 were corrupted

by severe stripes, and band 156 was affected by a dead-line. The full-resolution images of Figure 11 will be available at https://github.com/LinaZhuang (accessed on Nov. 1, 2021). All methods (except NAILRMA) are able to remove the dead-line in band 156. For severe stripes in the other three bands, we can see that FastHyDe, LRMR, L1HyMixDe, and HySuDeep achieved good restoration results while obvious stripes remained within the results of other methods.

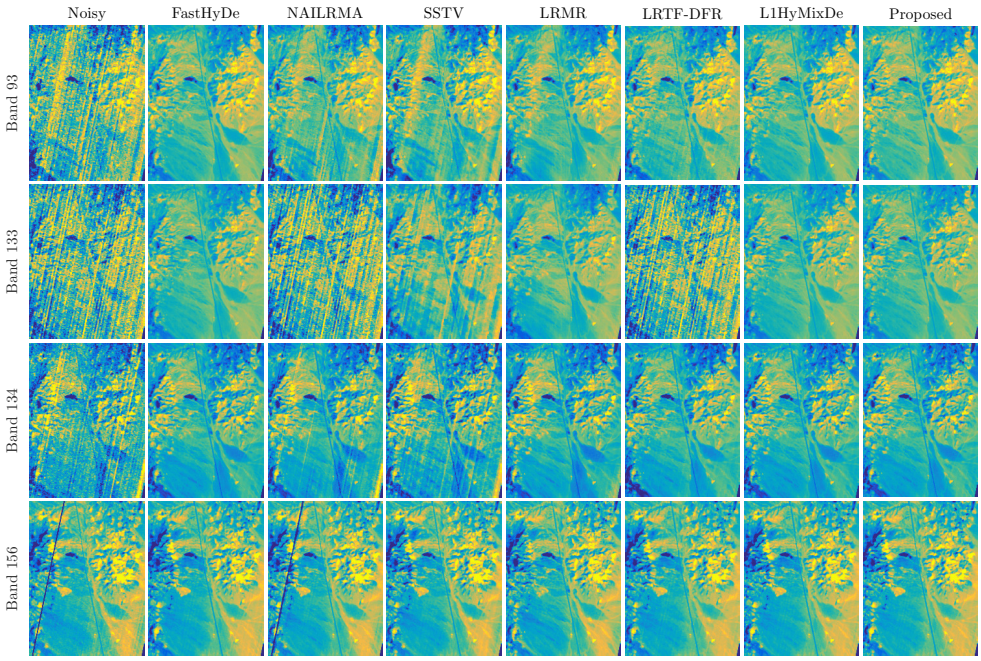

Figure 11. Bands 93, 133, 134, and 156 of the Hyperion Cuprite data before and after denoising.

5.2. Tiangong-1 Dataset

The Tiangong-1 dataset was acquired over an area of Qinghai Province, China in May 2013, by a sensor placed in the Tiangong-1 imager, which has a 75-band push broom scanner with nominal bandwidths of 23 nm short wave infrared (SWIR), covering from 800 nm to 2500 nm. A subregion image of size 351 $\times$ 253 pixels was tested. Three bands displaying strong noise are shown in Figure 12, where band 1 has severe cross-track illumination error, and bands 36 and 65 contain obvious stripes. The full-resolution images of Figure 12 will be available at https://github.com/LinaZhuang (accessed on Nov. 1, 2021). Comparing the images before denoising and after denoising, we can see that FastHyDe, L1HyMixDe, and HySuDeep can correct the illumination error in band 1. In bands 36 and 65, FastHyDe, LRMR, L1HyMixDe, and HySuDeep can alleviate stripe noise. If we focus on band 65, our HySuDeep method obtains the visually best result considering the restoration of the areas in the red circles.

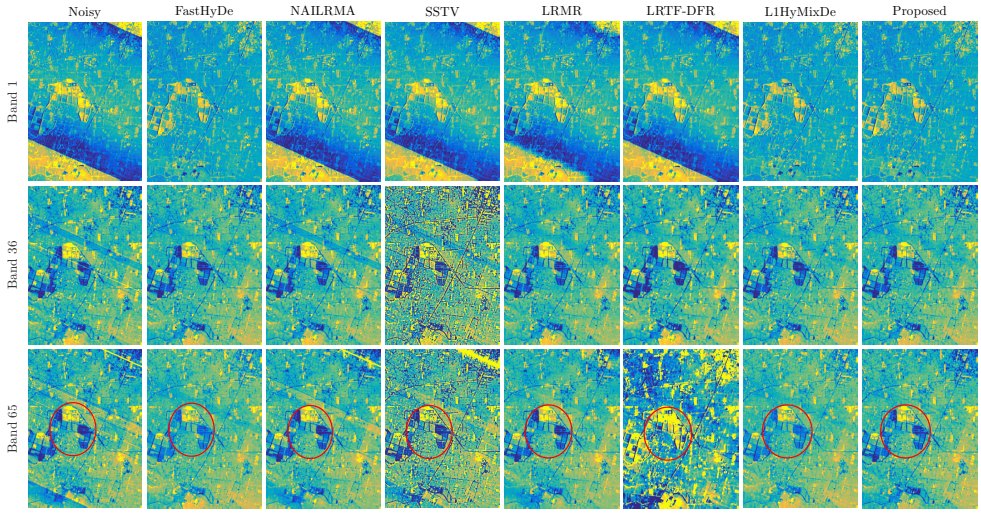

Figure 12. Bands 1, 36, and 65 of the Tiangong-1 data before and after denoising.

6. Conclusions

This paper introduces a hyperspectral mixed noise removal method, HySuDeep, by exploiting two important characteristics of HSIs. HySuDeep takes advantage of the spectral low-rankness of HSIs by representing clean spectral vectors in a low-dimensional subspace, which significantly improves the conditioning of the denoising problem. The spatial correlation of HSIs is exploited by adding CNN regularization for eigenimages. Although the network was trained using grayscale images acquired from commercial cameras, without using remote sensing images, the network still achieves impressive performance for HSI denoising. The reason is that both kinds of images are natural images, sharing the same properties, such as local and non-local similarity, and piece-wise smoothness. Therefore, the image prior learned by the network from grayscale images is also applicable to HSIs. Experimental results on both simulated and real HSIs show the superiority of the proposed method for mixed noise in HSIs.

Author Contributions: Conceptualization, L.Z. and X.F. ; methodology, L.Z.; validation, L.Z. and X.F.; formal analysis, L.Z.; writing—original draft preparation, L.Z.; writing—review and editing, M.K.N.; visualization, X.F.; supervision, M.K.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 42001287. The work of M.K.N. was partially supported by the HKRGC GRF 12300218, 12300519, 17201020, and 17300021.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bioucas-Dias, J.; Plaza, A.; Dobigeon, N.; Parente, M.; Du, Q.; Gader, P.; Chanussot, J. Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 354–379, doi:10.1109/JSTARS.2012.2194696.
2. Ellis, J.M.; Davis, H.H.; Zamudio, J.A. Exploring for onshore oil seeps with hyperspectral imaging. *Oil Gas J.* **2001**, *99*, 49–58.
3. Lowe, A.; Harrison, N.; French, A.P. Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress. *Plant Methods* **2017**, *13*, 1–12.
4. Abdellatif, M.; Peel, H.; Cohn, A.G.; Fuentes, R. Pavement Crack Detection from Hyperspectral Images Using a Novel Asphalt Crack Index. *Remote Sens.* **2020**, *12*, 3084.
5. Chang, Y.; Yan, L.; Fang, H.; Liu, H. Simultaneous destriping and denoising for remote sensing images with unidirectional total variation and sparse representation. *IEEE Geosci. Remote Sens. Lett.* **2013**, *11*, 1051–1055.

6. Xie, Q.; Zhao, Q.; Meng, D.; Xu, Z.; Gu, S.; Zuo, W.; Zhang, L. Multispectral images denoising by intrinsic tensor sparsity regularization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): Las Vegas, USA, 2016; pp. 1692–1700.
7. Yuan, Q.; Zhang, L.; Shen, H. Hyperspectral image denoising employing a spectral–spatial adaptive total variation model. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 3660–3677.
8. Zhuang, L.; Bioucas-Dias, J. Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 730–742.
9. Zhao, Y.Q.; Yang, J. Hyperspectral image denoising via sparse representation and low-rank constraint. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 296–308.
10. Jiang, S.; Hao, X. Hybrid Fourier-wavelet image denoising. *Electron. Lett.* **2007**, *43*, 1081–1082.
11. Rasti, B.; Sveinsson, J.R.; Ulfarsson, M.O.; Benediktsson, J.A. Hyperspectral image denoising using first order spectral roughness penalty in wavelet domain. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *7*, 2458–2467.
12. Fan, H.; Li, C.; Guo, Y.; Kuang, G.; Ma, J. Spatial-spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6196–6213.
13. Chang, Y.; Yan, L.; Zhong, S. Hyper-laplacian regularized unidirectional low-rank tensor recovery for multispectral image denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): Hawaii, USA, 2017; pp. 4260–4268.
14. Xie, T.; Li, S.; Sun, B. Hyperspectral Images Denoising via Nonconvex Regularized Low-Rank and Sparse Matrix Decomposition. *IEEE Trans. Image Process.* **2020**, *29*, 44–56, doi:10.1109/TIP.2019.2926736.
15. Zhang, H.; He, W.; Zhang, L.; Shen, H.; Yuan, Q. Hyperspectral image restoration using low-rank matrix recovery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 4729–4743.
16. Wang, M.; Wang, Q.; Chanussot, J.; Li, D. Hyperspectral Image Mixed Noise Removal Based on Multidirectional Low-Rank Modeling and Spatial–Spectral Total Variation. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 488–507.
17. Zhuang, L.; Fu, X.; Ng, M.K.; Bioucas-Dias, J.M. Hyperspectral Image Denoising Based on Global and Nonlocal Low-Rank Factorizations. *IEEE Trans. Geosci. Remote. Sens.* **2021**, doi: 10.1109/TGRS.2020.3046038.
18. He, W.; Yao, Q.; Li, C.; Yokoya, N.; Zhao, Q. Non-Local Meets Global: An Integrated Paradigm for Hyperspectral Denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019; pp. 6868–6877.
19. Zhuang, L.; Gao, L.; Zhang, B.; Fu, X.; Bioucas-Dias, J.M. Hyperspectral image denoising and anomaly detection based on low-rank and sparse representations. *IEEE Trans. Geosci. Remote. Sens.* **2020**, doi: 10.1109/TGRS.2020.3040221.
20. Lin, J.; Huang, T.Z.; Zhao, X.L.; Jiang, T.X.; Zhuang, L. A Tensor Subspace Representation-Based Method for Hyperspectral Image Denoising. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *59*, 7739–7757, doi:10.1109/TGRS.2020.3032168.
21. Zheng, Y.B.; Huang, T.Z.; Zhao, X.L.; Chen, Y.; He, W. Double-factor-regularized low-rank tensor factorization for mixed noise removal in hyperspectral image. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8450–8464.
22. Cao, C.; Yu, J.; Zhou, C.; Hu, K.; Xiao, F.; Gao, X. Hyperspectral image denoising via subspace-based nonlocal low-rank and sparse factorization. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 973–988.
23. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155, doi:10.1109/TIP.2017.2662206.
24. Zhang, K.; Zuo, W.; Zhang, L. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Trans. Image Process.* **2018**, *27*, 4608–4622.
25. Guo, S.; Yan, Z.; Zhang, K.; Zuo, W.; Zhang, L. Toward Convolutional Blind Denoising of Real Photographs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR): Long Beach, USA, 2019; pp. 1712–1722, doi:10.1109/CVPR.2019.00181.
26. Zhang, Q.; Yuan, Q.; Li, J.; Liu, X.; Shen, H.; Zhang, L. Hybrid noise removal in hyperspectral imagery with a spatial-spectral gradient network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7317–7329.
27. Chang, Y.; Yan, L.; Fang, H.; Zhong, S.; Liao, W. HSI-DeNet: Hyperspectral Image Restoration via Convolutional Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 667–682, doi:10.1109/TGRS.2018.2859203.
28. Zhang, Q.; Yuan, Q.; Li, J.; Sun, F.; Zhang, L. Deep spatio-spectral Bayesian posterior for hyperspectral image non-i.i.d. noise removal. *ISPRS J. Photogramm. Remote Sens.* **2020**, *164*, 125–137, doi:10.1016/j.isprsjprs.2020.04.010.
29. Venkatakrishnan, S.V.; Bouman, C.A.; Wohlberg, B. Plug-and-Play priors for model based reconstruction. In Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing, 2013; pp. 945–948, doi:10.1109/GlobalSIP.2013.6737048.
30. Chan, S.H.; Wang, X.; Elgendy, O.A. Plug-and-play ADMM for image restoration: Fixed-point convergence and applications. *IEEE Trans. Comput. Imaging* **2016**, *3*, 84–98.
31. Romano, Y.; Elad, M.; Milanfar, P. The little engine that could: Regularization by denoising (RED). *SIAM J. Imaging Sci.* **2017**, *10*, 1804–1844.
32. Zhuang, L.; Ng, M.K. Hyperspectral Mixed Noise Removal By ℓ_1-Norm-Based Subspace Representation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1143–1157, doi:10.1109/JSTARS.2020.2979801.
33. Simoes, M.; Bioucas-Dias, J.; Almeida, L.B.; Chanussot, J. A convex formulation for hyperspectral image superresolution via subspace-based regularization. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 3373–3388.

34. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Spectral-Spatial Hyperspectral Image Segmentation Using Subspace Multinomial Logistic Regression and Markov Random Fields. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 809–823, doi:10.1109/TGRS.2011.2162649.
35. Gao, L.; Li, J.; Khodadadzadeh, M.; Plaza, A.; Zhang, B.; He, Z.; Yan, H. Subspace-Based Support Vector Machines for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 349–353, doi:10.1109/LGRS.2014.2341044.
36. Zhuang, L.; Lin, C.; Figueiredo, M.A.T.; Bioucas-Dias, J.M. Regularization Parameter Selection in Minimum Volume Hyperspectral Unmixing. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9858–9877, doi:10.1109/TGRS.2019.2929776.
37. Bioucas-Dias, J.M.; Nascimento, J.M.P. Hyperspectral Subspace Identification. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2435–2445, doi:10.1109/TGRS.2008.918089.
38. Afonso, M.V.; Bioucas-Dias, J.M.; Figueiredo, M.A. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Trans. Image Process.* **2010**, *20*, 681–695.
39. Afonso, M.; Bioucas-Dias, J.; Figueiredo, M. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Process.* **2010**, *19*, 2345–2356.
40. Eckstein, J.; Bertsekas, D.P. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **1992**, *55*, 293–318.
41. Combettes, P.; Patric, J.C. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 185–212.
42. Figueiredo, M.A.; Bioucas-Dias, J.M. Restoration of Poissonian images using alternating direction optimization. *IEEE Trans. Image Process.* **2010**, *19*, 3133–3145.
43. Teodoro, A.M.; Bioucas-Dias, J.M.; Figueiredo, M.A. A convergent image fusion algorithm using scene-adapted Gaussian-mixture-based denoising. *IEEE Trans. Image Process.* **2018**, *28*, 451–463.
44. Dian, R.; Li, S.; Kang, X. Regularizing hyperspectral and multispectral image fusion by CNN denoiser. *IEEE Trans. Neural Networks Learn. Syst.* **2021**, *32*, 1124–1135.
45. Gao, L.; Du, Q.; Zhang, B.; Yang, W.; Wu, Y. A comparative study on linear regression-based noise estimation for hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 488–498.
46. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.M.; Chanussot, J. Hyperspectral Remote Sensing Data Analysis and Future Challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36, doi:10.1109/MGRS.2013.2244672.
47. Altmann, Y.; Pereyra, M.; Bioucas-Dias, J.M. Collaborative sparse regression using spatially correlated supports-application to hyperspectral unmixing. *IEEE Trans. Image Process.* **2015**, *24*, 5800–5811.
48. He, W.; Zhang, H.; Zhang, L.; Shen, H. Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3050–3061.
49. Aggarwal, H.K.; Majumdar, A. Hyperspectral image denoising using spatio-spectral total variation. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 442–446.
50. Chang, Y.; Yan, L.; Wu, T.; Zhong, S. Remote sensing image stripe noise removal: From image decomposition perspective. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7018–7031.
51. Golub, G.H.; Reinsch, C. Singular value decomposition and least squares solutions. In *Linear Algebra*; Springer: Berlin/Heidelberg, Germany, 1971; pp. 134–151.
52. Wright, J.; Ganesh, A.; Rao, S.; Peng, Y.; Ma, Y. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. *Adv. Neural Inf. Process. Syst.* **2009**, *58*, 2080–2088.
53. Nascimento, J.; Dias, J. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 898–910.
54. Bioucas-Dias, J.M.; Figueiredo, M. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In Proceedings of the 2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS): Reykjavík, Iceland, 2010; pp: 1-4.