

Article

Hyperspectral Image Denoising via Low-Rank Tucker Decomposition with Subspace Implicit Neural Representation

Cheng Cheng ¹, Dezhi Sun ¹, Yaoyuan Yang ¹, Zhoucheng Guo ¹ and Jiangjun Peng ^{1,2,*} ¹ School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, China² Shenzhen Research Institute of Northwestern Polytechnical University, Shenzhen 518057, China

* Correspondence: pengjj@nwpu.edu.cn

Abstract

Hyperspectral image (HSI) denoising is an important preprocessing step for downstream applications. Fully characterizing the spatial-spectral priors of HSI is crucial for denoising tasks. In recent years, denoising methods based on low-rank subspaces have garnered widespread attention. In the low-rank matrix factorization framework, the restoration of HSI can be formulated as a task of recovering two subspace factors. However, hyperspectral images are inherently three-dimensional tensors, and transforming the tensor into a matrix for operations inevitably disrupts the spatial structure of the data. To address this issue and better capture the spatial-spectral priors of HSI, this paper proposes a modeling approach named low-rank Tucker decomposition with subspace implicit neural representation (LRTSINR). This data-driven and model-driven joint modeling mechanism has the following two advantages: (1) Tucker decomposition allows for the characterization of the low-rank properties across multiple dimensions of the HSI, leading to a more accurate representation of spectral priors; (2) Implicit neural representation enables the adaptive and precise characterization of the subspace factor continuity under Tucker decomposition. Extensive experiments demonstrate that our method outperforms a series of competing methods.

Keywords: spatial-spectral prior; subspace factor continuity; multiway low-rank property; Tucker decomposition; implicit neural representation



Academic Editor: Salah Bourennane

Received: 27 June 2025

Revised: 25 July 2025

Accepted: 26 July 2025

Published: 18 August 2025

Citation: Cheng, C.; Sun, D.; Yang, Y.; Guo, Z.; Peng, J. Hyperspectral Image Denoising via Low-Rank Tucker Decomposition with Subspace Implicit Neural Representation. *Remote Sens.* **2025**, *17*, 2867. <https://doi.org/10.3390/rs17162867>

Copyright: © 2025 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 imaging (HSI) is a sophisticated imaging technique that captures data across hundreds of adjacent spectral bands within the electromagnetic spectrum. These rich spectral-spatial data enable a wide range of applications, including classification [1–4], super-resolution [5], haze removal [6,7], compressive sensing [8], spectral unmixing [9], and mineral exploration [10]. However, HSI data are often contaminated by noise due to sensor limitations, atmospheric effects, and environmental disturbances [11], which degrades spectral quality and adversely affects downstream tasks. To address this, it is crucial to extract and exploit prior information from the data to enhance denoising performance. Existing HSI denoising methods can generally be categorized into two groups based on their use of prior knowledge: model-based approaches and data-driven methods.

Model-based methods primarily leverage statistical techniques to construct regularization terms that encode prior information in hyperspectral images (HSI). The priors in HSI can be classified into two main types: spectral low-rankness [12–18] and spatial priors. Spatial priors typically include local smoothness [19–23] and non-local similarity [24–26].

However, the effectiveness of model-based methods in utilizing raw data priors tends to be limited [27]. As a result, fast model-based approaches that focus on subspace prior extraction have emerged [28–30]. These fast subspace-based algorithms are grounded in the linear mixing model of HSI [31]. According to the unmixing concept, the spectral signature at any spatial point (representing a specific material) can be expressed as a linear combination of the spectral signatures of multiple materials (a set of basis vectors). The spectral signatures of the various materials form the endmember matrix, while the coefficients that represent the abundance of materials at each point in space form the abundance map. If the number of bases in the endmember matrix is chosen to match the rank, the linear mixing model corresponds to low-rank matrix/tensor decomposition. These characteristics of the subspace factors align well with the spatial image structure and spectral continuity of the raw HSI data. Therefore, regularization terms can be directly defined on a smaller-sized subspace matrix to capture the structural priors of the HSI, leading to an acceleration in computation [30]. Existing subspace modeling methods are all based on matrix decomposition. However, the hyperspectral data themselves are three-dimensional cube data. Pulling them into a matrix for modeling will definitely destroy the spatial structure of the data, which is not conducive to the restoration task of high-dimensional images. In addition, both space-based and subspace-based methods face challenges related to insufficient prior characterization. Manually defined regularization terms may fail to capture the unique prior information inherent in the data fully [32].

In contrast, data-driven methods exploit the powerful feature extraction capabilities of neural networks to learn prior information directly from HSI data. Depending on the training approach, these methods can be broadly categorized into three types: supervised learning, plug-and-play, and unsupervised learning. In the supervised learning approach, a deep denoising network is trained using paired clean–noisy HSI data. This method is intuitive and offers fast inference [33–39]. However, its performance may degrade significantly when the testing data differ from the training data, leading to poor generalization. The plug-and-play approach involves selecting a pre-trained network to function as the denoiser [40,41]. While this method can be effective, it often suffers from structure mismatches and noise level discrepancies between the training domain of the pre-trained model and the target HSI data, leaving room for further improvements. The unsupervised learning approach, on the other hand, directly uses a specially designed network to denoise images with random white noise input, as seen in the Deep Image Prior (DIP) technique [42–47]. The core idea is that convolutional neural networks can effectively encode image structure. However, due to the rich spectral and spatial information inherent in HSI data, it is challenging for a single network to fully capture both the spectral and spatial priors in an unsupervised setting. Additionally, determining the optimal early stopping criteria for the network is another challenge, as it is critical to obtain the best possible solution.

To alleviate these problems, we propose a hybrid approach that integrates model-driven and data-driven methodologies. Specifically, we employ low-rank decomposition to capture the spectral correlations inherent in HSI data, while leveraging neural networks to model spatial priors. Unlike previous methods that rely on matrix-based modeling, we directly adopt Tucker decomposition to represent the multi-dimensional low-rank structure of HSI. As illustrated in Figure 1a, Tucker decomposition allows us to constrain the rank along each mode by adjusting the dimensions of the corresponding factor matrices, thereby effectively modeling low-rank characteristics across all three dimensions. The empirical analysis shown in Figure 1c, conducted on the DC mall dataset, reveals significant low-rank properties along all spatial and spectral directions, highlighting the necessity of jointly modeling these dimensions to fully exploit the spectral prior. We further investigate the subspace factor matrices obtained via Tucker decomposition (i.e., Figure 1b), and observe a

strong continuity prior, particularly in the spectral mode, as shown in Figure 1d–f. This observation is intuitive—since the factor matrices are derived from the original data, they naturally inherit their spectral continuity and spatial smoothness. This continuity becomes an essential prior that, when accurately modeled, can significantly enhance denoising performance. To capture this continuity, a straightforward approach is to use total variation (TV) regularization. The TV regularization is in the form of differences between adjacent rows and columns, which means that the weight at each point is 1; that is, the continuity of each point is the same, which is inconsistent with the actual situation. As shown in Figure 1d–f, it reveals that the degree of continuity varies across different factors and among the columns within a single factor matrix. This motivates us to go beyond hand-crafted regularization and instead learn the priors adaptively from the data. Inspired by the Deep Image Prior (DIP) [45] and implicit neural representation (INR) [46] techniques, we adopt INR to parametrize each subspace factor to enhance interpretability. Specifically, we employ three fully connected neural networks with sine activation functions to parameterize each factor matrix obtained from the low-rank decomposition. The overall data parameterization process and network architecture are illustrated in Figure 2.

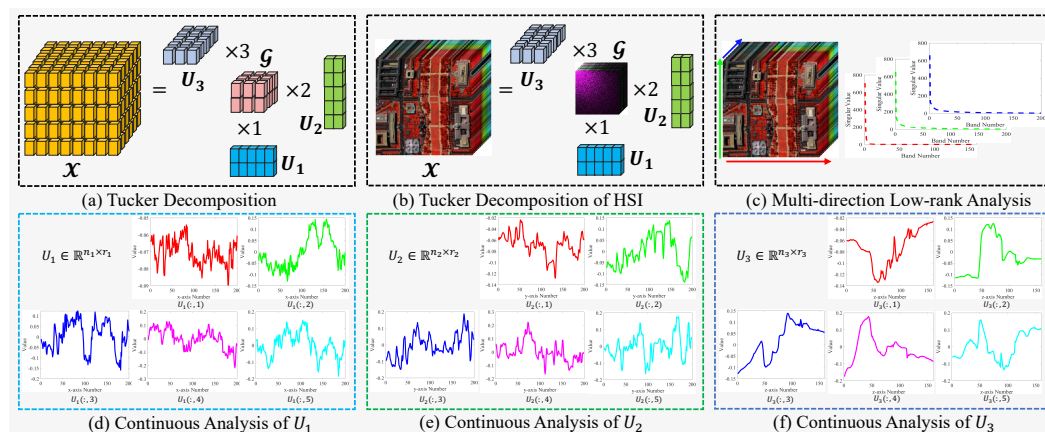


Figure 1. The spatial and spectral prior analysis for *DC mall* HSI data. (a) Tucker decomposition diagram—under Tucker decomposition, a third-order tensor can be expressed as the product of a core tensor $\mathcal{G}$ and three subspace factor matrices $\{U_i\}_{i=1}^3$, that is, $\mathcal{X} = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3$, where $\times_i$ is the product between the tensor and matrix; (b) The Tucker decomposition diagram of *DC mall* data; (c) The multi-way low-rank property of *DC mall* data, the right subfigures show the singular value curve of the three dimensions; (d) The continuous analysis of each column of the first subspace factor U_1 ; (e) The continuous analysis of each column of the first subspace factor U_2 ; (f) The continuous analysis of each column of the first subspace factor U_3 .

In summary, this work makes the following contributions:

1. The paper adopts Tucker decomposition to explicitly model the low-rank structure of hyperspectral data across both spatial and spectral dimensions. By decomposing the data into a core tensor and a set of mode-specific factor matrices, this method effectively captures intrinsic correlations along each mode.
2. The paper identifies a novel continuity prior in the factor matrices obtained through Tucker decomposition, revealing that the inherent local smoothness and continuity of hyperspectral data are preserved in its subspace representations. This finding provides a new and effective way to characterize the spatial information of HSI data.
3. The paper proposes leveraging implicit neural representations to model the continuity of factor matrices. By doing so, each column of a factor matrix can adaptively learn its intrinsic level of continuity, enabling more accurate and flexible encoding of continuity priors.

- The paper develops an unsupervised optimization framework that requires no additional training data. This approach combines the advantages of both model-driven and data-driven methods, eliminating concerns over generalization, and achieves state-of-the-art denoising performance in extensive experiments.

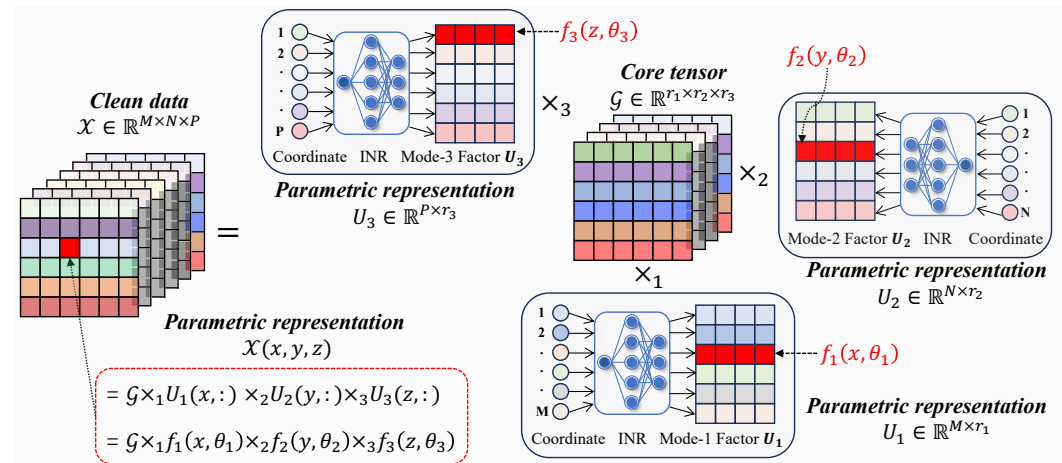


Figure 2. Schematic diagram of the parametric representation framework for subspace factors under Tucker decomposition. According to the definition of Tucker decomposition, any point on the original data can be represented by the following equation: $\mathcal{X}(x, y, z) = \mathcal{G}(i, k, j) f_x(x, \Theta_1) f_y(y, \Theta_2) f_z(z, \Theta_3)$. Therefore, to capture the spatial information of the original data, it is sufficient to ensure that the columns of each subspace factor exhibit continuity. This continuity can be controlled by a neural network in the framework of neural network parametrization.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 presents our proposed model and its optimization process. Section 4 presents the experimental results. Finally, the conclusions are drawn in Section 5. Throughout the paper, we denote scalars, vectors, matrices, and tensors in light, bold lower-case, bold upper-case, and upper cursive letters, respectively.

2. Related Work

In this section, we provide a brief overview of the current model-based approaches and several data-driven methods for HSI noise removal.

2.1. Model-Driven Methods

Model-based methods focus on designing regularizers to represent the prior information in HSI data. Common priors include low-rank properties in the spectral dimension, as well as local smoothness and non-local self-similarity in the spatial domain. Low-rank priors assume that HSI data lie in a low-dimensional subspace along the spectral direction, typically modeled using nuclear norms or low-rank matrix/tensor decomposition (LRMD/LRTD) methods like RPCA [13], WNNM [48,49], TRPCA [15,50], HyRes [51], LRMR [14], and NMoG [52]. Local smoothness assumes that neighboring elements in HSI are often similar, and this is usually represented by total variation (TV) regularizers, integrated into low-rank frameworks for better denoising results. Methods like E-3DTV [17], LRTV [19], LRTDTV [20], and CTV [53] apply this regularization. Non-local self-similarity refers to repeating patterns across different spatial areas, and is often incorporated into LRMD/LRTD methods to eliminate noise. Notable methods include KBR [24], BM4D [54], and LLRT [55].

Given the large size of HSIs, applying regularization directly on raw data is inefficient. Subspace-based methods address this by performing low-rank decomposition, allowing for regularizers to be applied to smaller subspace factors. This reduces computation time

while preserving important features like local smoothness and non-local self-similarity. Methods such as RCTV [30], NGMeet [29], FastHyDe [26], and LRTFDFR [56] leverage these subspace characteristics for more efficient denoising. Despite their physical interpretability, model-based methods have limitations in fully capturing the unique priors in HSI data, leaving room for improvement in denoising performance.

2.2. Data-Driven Methods

Unlike model-based methods, data-driven methods directly learn a denoising mapping from large datasets. These approaches can be classified into three main categories: supervised learning, plug-and-play, and unsupervised learning. In supervised learning, deep denoising models are trained using paired clean-noisy data, such as in GRN [33], T3SC [38], QRNN3D [34], TRQ3DNet [57], HSID-CNN [35], HSI-DeNet [36], ADRN [58], 3D-ADNet [59], MAC-Net [60], and SST [61]. These methods achieve high fitting accuracy but often face challenges in real-world scenarios due to their limited generalization capability. The plug-and-play method uses a pre-trained network as a denoiser for specific sub-problems within the denoising process. This approach benefits from the robustness of the pre-trained model, resulting in good denoising performance. Representative methods include FastHyMix [40] and AdHyDe [41]. However, since the selected network may not be well matched to the data and noise types, there is still room for improvement. In unsupervised learning, networks are designed to denoise from random white noise inputs without relying on labeled clean data. Examples include DHP [42], DS2DP [43], and S2DIP [46]. While these methods do not require clean labels, they often underperform compared to supervised and plug-and-play methods. For instance, DHP [42] tries to capture spatial and spectral features of HSI, but it struggles to extract sufficient useful information in the unsupervised setting. To improve performance in unsupervised learning, it is crucial to focus on both effective HSI structure modeling and the design of suitable network modules for better prior information extraction.

3. Low-Rank Tucker Subspace Factor Parameterized Representation

3.1. Problem Formulation

Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ represent a clean hyperspectral image (HSI), where $n_1 \times n_2$ denotes the spatial dimensions and n_3 denotes the number of spectral bands. Assuming the presence of universal additive noise, the observation model can be expressed as:

$$\mathcal{Y} = \mathcal{X} + \mathcal{N} + \mathcal{E}, \quad (1)$$

where $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ represents the observed degraded HSI, $\mathcal{N} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ captures the Gaussian noise components, and $\mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ captures the sparse noise, such as impulse noise, stripe noise, data dropouts, and missing pixels [19,20].

Recovering $\mathcal{X}$ from $\mathcal{Y}$ is a standard inverse problem, and solving such problems necessitates the incorporation of prior knowledge about the data. Generally speaking, the HSI denoising problem can be written as:

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{N}, \mathcal{E}} \quad & \mathcal{R}(\mathcal{X}) + \tau \mathcal{R}(\mathcal{E}) + \beta \mathcal{R}(\mathcal{N}) \\ \text{s.t.} \quad & \mathcal{Y} = \mathcal{X} + \mathcal{E} + \mathcal{N}, \end{aligned} \quad (2)$$

where $\mathcal{R}(\mathcal{X})$, $\mathcal{R}(\mathcal{E})$, and $\mathcal{R}(\mathcal{N})$ represent the prior term of the clean-image, sparse noise term, and Gaussian noise term, respectively, and τ, β denote the trade-off parameters.

3.2. Prior Mining

Common priors for hyperspectral images (HSI) include the low-rank nature of the spectral data and the spatial structural properties of the image. Recalling Figure 1c, we can observe that HSI possesses the low-rankness along all three dimensions. To accurately capture the multi-dimensional low-rank structure of data, a series of tensor decomposition-based methods have emerged [12]. Among them, low-rank Tucker decomposition is one of the most commonly used approaches. The Tucker decomposition is defined as follows:

Definition 1 (Tucker Decomposition [12]). *Tucker decomposition factorizes a tensor into a core tensor multiplied by a matrix along each mode. For a 3rd-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, the decomposition is:*

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \quad (3)$$

where $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the core tensor, $\times_i$ is the mode product between the tensor and the matrix [12], $[r_1, r_2, r_3]$ is the Tucker rank, and $\mathbf{U}_n \in \mathbb{R}^{I_n \times r_n}$ are the factor matrices.

According to the definition of Tucker decomposition, if we need to characterize the low rank of multi-dimensional data, we only need to choose a relatively small and appropriate Tucker rank $[r_1, r_2, r_3]$.

Apart from the low-rankness, the local smoothness is another important spatial prior. The local smoothness means that adjacent pixels are likely to belong to the same class of material, and pixels of the same material typically have similar values. Therefore, the total variation (TV) regularization [62] can be used to characterize this property. The definition of total variation is as follows:

Definition 2 (Total Variation [62]). *For an image $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$, the total variation regularization is defined as:*

$$\|\mathbf{X}\|_{TV} = \|\nabla \mathbf{X}\|_1 = \sum_{i,j} (|\mathbf{X}_{i+1,j} - \mathbf{X}_{i,j}| + |\mathbf{X}_{i,j+1} - \mathbf{X}_{i,j}|), \quad (4)$$

where $\nabla(\cdot)$ is the difference operator. Or, in the continuous case:

$$\|\mathbf{X}\|_{TV} = \int_{\Omega} |\nabla \mathbf{X}| dx, \quad (5)$$

where Ω represents the integration space. For images, Omega is the integration interval in the horizontal and vertical directions.

As seen in Figure 1d–f, each factor matrix exhibits varying degrees of continuity across its columns, which originates from the local smoothness of the image. In fact, HSI can be viewed as a stack of multiple images along the spectral dimension, and these images inherently possess local smoothness;

When an image is further decomposed into a series of vectors, this local smoothness is reflected as continuity among the vectors. A natural approach to modeling vector continuity is to apply TV regularization. However, as shown in Figure 1d–f, the degree of continuity varies across different columns, while TV regularization implicitly assumes uniform weighting at every point of the vector. This uniform assumption fails to capture the varying continuity levels in the factor matrices accurately. Therefore, how to precisely characterize the varying continuity becomes a crucial problem for HSI denoising. Under the low-rank Tucker decomposition, the HSI denoising problem based on the subspace prior mining can be reformulated as:

$$\begin{aligned}
& \min_{\mathcal{G}, \mathcal{N}, \mathcal{E}, \{\mathbf{U}_i\}_{i=1}^3} \sum_{i=1}^3 \mathcal{R}(\mathbf{U}_i) + \tau \mathcal{R}(\mathcal{E}) + \beta \mathcal{R}(\mathcal{N}) \\
& \text{s.t. } \mathcal{Y} = \mathcal{X} + \mathcal{E} + \mathcal{N}, \\
& \mathcal{X} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3.
\end{aligned} \tag{6}$$

3.3. Parametric Representation

In model (6), how to mine the prior information of each subspace factor and encode them into $\mathcal{R}(\mathbf{U}_i)$ is the key to obtaining high-performance denoising performance.

Different from low-rank matrix decomposition, the subspace factor is still an image [28,30,63]. Under Tucker decomposition, the subspace factor is composed of a series of one-dimensional vectors, that is, $\mathbf{U}_i = [\mathbf{U}_i(:,1), \mathbf{U}_i(:,2), \dots, \mathbf{U}_i(:,r_i)]$, $i = 1, 2, 3$. Therefore, to characterize the prior of $\mathbf{U}_i$ is to characterize the continuity prior presented in Figure 1d–f. From Figure 1, we can observe that there is a significant difference in the continuity between different factor matrices and their respective columns. To better capture the degree of continuity between the columns of different factor matrices, we need to establish a data-adaptive prior representation mechanism. Inspired by the implicit neural representation [64,65], we consider using implicit neural representations to parameterize each factor matrix $\mathbf{U}_i$ in the Tucker decomposition. The definition of the implicit neural representation is as follows:

Definition 3 (Implicit Neural Representation [64]). *An implicit neural representation is defined as a continuous function parameterized by a neural network with a sine activation function:*

$$f_{\theta} : \mathbf{x} \in \mathbb{R}^d \longrightarrow \mathbf{y} \in \mathbb{R}^k \tag{7}$$

where $\mathbf{x}$ denotes an input coordinate, $\mathbf{y}$ is the corresponding signal value, and d denotes the dimensions of the input; for example, $d = 1$ for a one-dimensional signal and $d = 2$ for an image since the image has two dimensions. The function f_{θ} is typically implemented as a multilayer perceptron (MLP) with parameters θ , enabling continuous and differentiable modeling of complex signals.

Based on the definition of INR (i.e., Definition 3), we assign an implicit neural representation network to each factor matrix $\mathbf{U}_i$. Through experiments, we found that a three-layer fully connected network with sine activation functions (using a frequency of 1) yields the best performance. The input to the network is a single scalar, and the number of hidden units is set to four times the height of the data to be reconstructed (e.g., for data of size $h \times w \times p$, the neural number of each hidden layer is set to $4h$). The output is a vector, whose length corresponds to the rank of the respective factor matrix—specifically, the output lengths are r_1 , r_2 , and r_3 for the three-factor matrices $\{\mathbf{U}\}_{i=1}^3$. By feeding all coordinate inputs into the network and concatenating the outputs row-by-row, we can obtain the parameterized representation of each factor matrix under Tucker decomposition, as illustrated in Figure 2. Then, we can further obtain the parameterized representation of the needed-to-be-repaired clean data as follows:

$$\begin{aligned}
\mathcal{X} &:= \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \\
\mathbf{U}_1 &:= f_{\mathbf{x}}(\mathbf{x}, \Theta_1) \in \mathbb{R}^{n_1 \times r_1}, \\
\mathbf{U}_2 &:= f_{\mathbf{y}}(\mathbf{y}, \Theta_2) \in \mathbb{R}^{n_2 \times r_2}, \\
\mathbf{U}_3 &:= f_{\mathbf{z}}(\mathbf{z}, \Theta_3) \in \mathbb{R}^{n_3 \times r_3},
\end{aligned} \tag{8}$$

where $f_{\mathbf{x}}(\cdot; \Theta_1)$, $f_{\mathbf{y}}(\cdot; \Theta_2)$ and $f_{\mathbf{z}}(\cdot; \Theta_3)$ are INR network.

For Equation (8), the INR network is implemented as a fully connected neural network with sine activation functions. This architectural choice ensures that the mapping function $f(\cdot)$ is inherently continuous and smooth. When the input to the network consists of continuous coordinate values, the output—typically a high-dimensional vector representing image intensity, spectral signature, or other data features—varies continuously with respect to those input coordinates. This guarantees that the INR can preserve continuity in the encoded data along the row-wise (spatial or spectral) direction. More concretely, when coordinate inputs vary smoothly, the corresponding output vectors also change smoothly in each of their dimensions. This continuity ensures that the INR does not introduce artificial discontinuities or artifacts in the encoded representation, which is particularly important for hyperspectral or natural image data, where spatial and spectral coherence are essential. This design ensures that the factor matrices represented by the INR maintain continuity along the row direction. Moreover, as the factor matrices are parameterized by the network, the optimization process enables the network to learn the degree of continuity at each point directly from the data.

3.4. Models

Based on the above analysis, we ultimately adopt Tucker decomposition to model the multi-dimensional correlations of the data to be restored. We use implicit neural representations combined with total variation regularization to capture the continuity of the factor matrices under Tucker decomposition. The Frobenius norm is employed to model Gaussian noise, while the ℓ_1 -norm is used to characterize sparse noise. As a result, we formulate the following denoising model:

$$\begin{aligned} \min_{\mathcal{G}, \mathcal{N}, \mathcal{E}, \{\Theta_i\}_{i=1}^3} \quad & \tau \sum_{i=1}^3 \|\nabla(\mathbf{U}_i)\|_1 + \beta \|\mathcal{N}\|_F^2 + \|\mathcal{E}\|_1 \\ \text{s.t.} \quad & \mathcal{Y} = \mathcal{X} + \mathcal{E} + \mathcal{N}, \\ & \mathcal{X} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \\ & \mathbf{U}_1 = f_{\mathbf{x}}(\mathbf{x}, \Theta_1) \in \mathbb{R}^{n_1 \times r_1}, \\ & \mathbf{U}_2 = f_{\mathbf{y}}(\mathbf{y}, \Theta_2) \in \mathbb{R}^{n_2 \times r_2}, \\ & \mathbf{U}_3 = f_{\mathbf{z}}(\mathbf{z}, \Theta_3) \in \mathbb{R}^{n_3 \times r_3}, \end{aligned} \quad (9)$$

where $\mathbf{x} = [1, 2, \dots, M]$, $\mathbf{y} = [1, 2, \dots, N]$ and $\mathbf{z} = [1, 2, \dots, P]$ are coordinate vectors.

From the perspective of effectiveness, both INR and TV regularization are used to characterize the continuity of the factor matrix. However, their effects are different. The TV regularizer assumes that the continuity of the original data between all adjacent rows/columns is consistent, which is obviously inconsistent with the actual situation. Therefore, INR is needed to more finely characterize the continuity of the data. However, model (9) is an unsupervised model, and the parameter update of INR depends on the design of the loss function. In order to enable the network to converge to a better solution, TV regularization is added to Equation (9) to increase the robustness of the model.

3.5. Optimization

Reviewing model (9), it is a constrained optimization problem. Here, we consider directly incorporating the constraint $\mathcal{E} = \mathcal{Y} - \mathcal{X} - \mathcal{N}$ into the optimization objective. Therefore, model (9) can be further expressed as the following optimization model:

$$\begin{aligned} \min_{\mathcal{G}, \mathcal{N}, \{\Theta_i\}_{i=1}^3} \quad & \mathcal{J}(\mathcal{G}, \mathcal{N}, \{\Theta_i\}_{i=1}^3) = \|\mathcal{Y} - \mathcal{G} \times_1 f_{\mathbf{x}}(\mathbf{x}, \Theta_1) \times_2 f_{\mathbf{y}}(\mathbf{y}, \Theta_2) \times_3 f_{\mathbf{z}}(\mathbf{z}, \Theta_3) - \mathcal{N}\|_1 \\ & + \beta \|\mathcal{N}\|_F^2 + \tau \{ \|\nabla(f_{\mathbf{x}}(\mathbf{x}, \Theta_1))\|_1 \\ & + \|\nabla(f_{\mathbf{y}}(\mathbf{y}, \Theta_2))\|_1 + \|\nabla(f_{\mathbf{z}}(\mathbf{z}, \Theta_3))\|_1 \} \end{aligned} \quad (10)$$

For Equation (10), the parameters that need to be updated in the entire optimization objective are the core tensor $\mathcal{G}$, Gaussian noise term $\mathcal{N}$, and the weights in the network, namely, $\{\Theta_i\}_{i=1}^3$. We adopt an iterative method to solve the above model (10). In each iteration, the Gaussian noise component $\mathcal{N}$ and clean data $\mathcal{X}$ can be updated using the following formula:

$$\begin{cases} \mathcal{N}^{t+1} = \mathcal{Y} - \mathcal{X}^t - \mathcal{S}_{1/(2\beta)}(\mathcal{Y} - \mathcal{X}^t) \\ \mathcal{X}^t = \mathcal{G} \times_1 f_{\mathbf{x}}(\mathbf{x}, \Theta_1^t) \times_2 f_{\mathbf{y}}(\mathbf{y}, \Theta_2^t) \times_3 f_{\mathbf{z}}(\mathbf{z}, \Theta_3^t) \end{cases} \quad (11)$$

where $\mathcal{S}_{\lambda}(\cdot)$ is the soft-threshold operator [53].

For the network parameters $\{\Theta_i\}_{i=1}^3$ and the core tensor $\mathcal{G}$, we can directly update them using gradient descent. Giving the learning rate α , the update rule is as follows:

$$\begin{cases} \mathcal{G}^{t+1} = \mathcal{G}^t - \alpha \frac{\partial \mathcal{J}(\mathcal{G}^t, \mathcal{N}^{t+1}, \{\Theta_i^t\}_{i=1}^3)}{\partial \mathcal{G}} \\ \Theta_1^{t+1} = \Theta_1^t - \alpha \frac{\partial \mathcal{J}(\mathcal{G}^t, \mathcal{N}^{t+1}, \{\Theta_i^t\}_{i=1}^3)}{\partial \Theta_1} \\ \Theta_2^{t+1} = \Theta_2^t - \alpha \frac{\partial \mathcal{J}(\mathcal{G}^t, \mathcal{N}^{t+1}, \{\Theta_i^t\}_{i=1}^3)}{\partial \Theta_2} \\ \Theta_3^{t+1} = \Theta_3^t - \alpha \frac{\partial \mathcal{J}(\mathcal{G}^t, \mathcal{N}^{t+1}, \{\Theta_i^t\}_{i=1}^3)}{\partial \Theta_3} \end{cases} \quad (12)$$

For convenience of viewing the entire optimization process and describing more details, we have placed the entire optimization process in Algorithm 1.

Algorithm 1 LRTSINR for HSI Denoising.

Input: Observed image $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$; The core tensor size (r_1, r_2, r_3) ; maximum iteration number T_max = 2000; relative error tolerance $\epsilon = 0.01$; $\mathbf{x} = [1, \dots, n_1]$, $\mathbf{y} = [1, \dots, n_2]$ and $\mathbf{z} = [1, \dots, n_3]$, the learning rate $\alpha = 1e^{-4}$.

Initialization: Set iteration number $t = 1$; , randomly initialize network parameters $f_{\mathbf{x}}(\mathbf{x}; \Theta_1^{(0)})$, $f_{\mathbf{y}}(\mathbf{y}; \Theta_2^{(0)})$, $f_{\mathbf{z}}(\mathbf{z}; \Theta_3^{(0)})$.

- 1: **while** $t \leq \text{T_max}$ **and** $\text{RelErr}^{(t)} < \epsilon$ **do**
- 2: Update $\mathcal{N}^t$ and recovered data $\mathcal{X}^t$ via Equation (11);
- 3: Carry out the gradient descent algorithm to update the core tensor $\mathcal{G}^t$ and the network weights $\{\Theta_i^t\}_{i=1}^3$ according to Equation (12).
- 4: Check the relative error $\text{RelErr}^{(t)} := \frac{\|\mathcal{X}^t - \mathcal{X}^{t-1}\|_F^2}{\|\mathcal{X}^t\|_F^2}$;
- 5: Update $t \rightarrow t + 1$;
- 6: **end while**

Output: The final restored HSI data $\mathcal{X}^t$.

3.6. Time Complexity Analysis

There are two variables that need to be updated in LRTSINR, namely, the Gaussian noise component $\mathcal{N} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and clean data $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. According to the low-rank Tucker decomposition, assuming that the ranks of the clean data $\mathcal{X}$ in the three directions are $[r_1, r_2, r_3]$, then $\mathcal{X}$ can be modeled by $\mathcal{X} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$, where the core tensor $\mathcal{G}$ has a size of $[r_1, r_2, r_3]$ and each factor matrix $\{\mathbf{U}_i\}_{i=1}^3$ has a size of $[n_i, r_i]$. In this paper, a fully connected network with two hidden layers is selected. The number of nodes in the middle layer is n_1 , so the number of parameters of each sub-network is $\mathcal{O}(n_1(1 + r_i) + 2n_1^2)$. For neural networks, gradient descent is the most commonly used update method. Therefore, in each step of the update, the update parameters required for the three sub-networks are $\mathcal{O}(n_1(3 + \sum_{i=1}^3 r_i) + 6n_1^2)$. The number of parameters in $\mathcal{G} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is $\prod_{i=1}^3 r_i$, so the time complexity of updating $\mathcal{X}$ is $\mathcal{O}(n_1(3 + \sum_i r_i) + 6n_1^2 + \prod_{i=1}^3 r_i)$. For the Gaus-

sian noise component $\mathcal{N}$, it can be solved by the soft threshold operator. For a tensor with size of $n_1 \times n_2 \times n_3$, the time complexity for updating $\mathcal{N}$ is $\mathcal{O}(\prod_{i=1}^3 n_i)$. Finally, integrating the time complexity of all variables, the time complexity of Algorithm 1 is $\mathcal{O}(n_1(3 + \sum_i r_i) + 6n_1^2 + \prod_{i=1}^3 r_i + \prod_{i=1}^3 n_i)$.

From the perspective of time complexity, the time complexity of the LRTSINR single step is not high. However, since LRTSINR is an unsupervised method, it should require more iterations to converge to a good result. Therefore, its running time is a little longer. However, with GPU acceleration, the running time of the LRTSINR algorithm is still comparable.

4. Experiments

In this section, we perform extensive experiments on both semi-real noise and real noise data to demonstrate the effectiveness of the proposed model more intuitively.

To evaluate the effectiveness of the proposed model, we compare it against fourteen state-of-the-art (SOTA) methods on both synthetic and real noisy HSI datasets. These competing methods are categorized into two groups: eight model-based methods and six data-driven methods. The model-based approaches include the following: low-rank regularization methods (e.g., LRMR [14]); low-rank combined with total variation regularization (e.g., LRTV [19], LRTDTV [20], CTV [53], RCTV [30]); low-rank combined with non-local self-similarity (e.g., NGMeet [29], GLF [26]); and plug-and-play methods (e.g., FHyMix [40]). The data-driven methods include one unsupervised deep learning method (e.g., S2DIP [46]) and one supervised deep learning method (e.g., RCILD [32]).

To ensure a more comprehensive comparison, we selected two clean datasets for simulated experiments: *DC mall* and *Cloth*. The *DC mall* dataset (200×200 pixels, 160 bands) used in [17] is a synthetic hyperspectral image simulating an urban shopping area, featuring relatively low spatial resolution but rich spectral information in the visible to near-infrared range. This dataset is particularly suitable for evaluating spectral fidelity preservation in denoising algorithms. In contrast, the *Cloth* (these data are available at <https://ieee-dataport.org/documents/cave-hsi#files> (accessed on 25 July 2025).) dataset (512×512 pixels, 31 bands) provides high-resolution textile imagery with complex spatial textures but fewer spectral bands, making it ideal for testing spatial detail recovery in denoising tasks. These complementary characteristics allow for comprehensive assessment of denoising algorithms across both spectral and spatial domains. Both datasets serve as clean references for controlled experiments where various noise types can be artificially introduced to simulate real-world scenarios. Furthermore, we select *Urban* used in [20] and *ZY01* used in [66] as the real hyperspectral datasets to evaluate the denoising performance under practical scenarios. The *Urban* dataset, captured by the AVIRIS sensor over an urban area, has a spatial resolution of 307×307 pixels and contains 210 spectral bands. It includes diverse ground materials such as rooftops, grass, and roads, making it a challenging dataset due to high spectral variability and noise levels. The *ZY01* dataset is acquired from the Chinese ZY-1 satellite and has a spatial resolution of 300×300 pixels with 166 spectral bands. It primarily captures natural scenes and agricultural regions, and is characterized by severe noise caused by atmospheric and environmental interference. These datasets are widely used benchmarks for evaluating the robustness of denoising algorithms on real noisy data.

All experiments are performed on a PC with an Intel(R) Core(TM) i5-10600KF CPU @ 4.10 GHz, an NVIDIA RTX 3080 GPU, and 10 GB of RAM.

4.1. Synthetic Data Experiments

To comprehensively evaluate the performance of the proposed method, we conduct simulation experiments by comparing it with several competing approaches using both

quantitative metrics and visual assessments. The clean hyperspectral image (HSI) dataset *DC mall* and the multispectral image (MSI) dataset *Cloth* are used as the ground truth for generating noisy scenarios.

We employ four widely used quantitative metrics to assess image restoration quality as follows: Mean Peak Signal-to-Noise Ratio (MPSNR), Mean Structural Similarity Index (MSSIM), Erreur Relative Globale Adimensionnelle de Synthèse (ERGAS), and Mean Spectral Angle Mapper (MSAM). Then, by adding different degrees of noise (derived from our simulation of real noise), we can calculate the restoration quality by comparing the difference between the denoised data and the original data. Higher MPSNR and MSSIM values, along with lower ERGAS and MSAM scores, indicate better denoising performance. In addition to quantitative evaluation, we qualitatively assess spatial structure preservation and spectral curve recovery to ensure a holistic evaluation.

Since real-world HSI data are often affected by complex noise patterns beyond simple i.i.d. Gaussian noise—including non-i.i.d. Gaussian noise, impulse noise, and deadlines—we simulate five different noise cases as follows:

- (a) Each band is corrupted by zero-mean Gaussian noise with a fixed standard deviation of 0.1.
- (b) Same as (a), with an additional deadline introduced in *DC mall* bands 91–130 and *Cloth* bands 21–30. The width of the deadline is randomly selected from 1, 2, and 3.
- (c) Each band is corrupted by Gaussian noise with standard variance 0.075 and salt-and-pepper noise with a missing fraction of 0.15.
- (d) Same as (c), with an added deadline in *DC mall* bands 91–130 and *Cloth* bands 21–30. The deadline width is randomly chosen from 1, 2, 3.
- (e) Each band is corrupted by Gaussian noise with standard deviation randomly sampled from $[0, 0.1]$, and salt-and-pepper noise with missing fraction randomly selected from $[0, 0.2]$. A deadline is also added in *DC mall* bands 91–130 and *Cloth* bands 21–30, with randomly chosen widths from 1, 2, and 3.

Tables 1 and 2 report the quantitative evaluation results of all competing denoising methods on both hyperspectral (HSI) and multispectral (MSI) datasets across five noise scenarios (cases (a) to (e)). In the LRTSINR model (9), there are three hyperparameters, namely, $\tau, \beta, [r_1, r_2, r_3]$. For multi-rank $[r_1, r_2, r_3]$, we use the following settings: $[r_1, r_2, r_3] = (0.9 * n_1, 0.9 * n_2, r_3)$, where r_3 needs to be fine-tuned and n_1 and n_2 denote the spatial dimensions (height and width) of the dataset. Since the spectral low rank of hyperspectral images is relatively strong, r_3 can generally be selected from 3–15. As for τ , it controls the continuity of the factor matrix. The smaller the value, the weaker the restriction on smoothness. Therefore, if the texture of an image is complex (that is, there are not so many flat areas), τ should take a relatively small value. In our experiments, we found that for LRTSINR, the τ value range of most remote sensing images is $[0.001, 0.2]$. Therefore, readers can choose the appropriate τ value according to this range and the role of τ . As for β , it controls the ratio between Gaussian noise and sparse noise. If the image contains a lot of Gaussian noise, one can take a smaller beta value to make the Gaussian noise term have a larger value. For the *Cloth* dataset, the hyperparameter configurations for noise case (a)–(b) and noise case (c)–(e) are $r_3 = 5, \tau = 0.01, \beta = 5$, and $r_3 = 6, \tau = 0.01, \beta = 100$. Similarly, for the *DC mall* dataset, the hyperparameter configurations for noise case (a)–(b) and noise case (c)–(e) are $r_3 = 6, \tau = 0.01, \beta = 5$ and $r_3 = 6, \tau = 0.01, \beta = 100$. From these two tables, we can observe the following phenomena. In the simpler settings involving Gaussian noise (cases (a) and (b)), the proposed LRTSINR method does not always outperform state-of-the-art non-local self-similarity (NLSS)-based approaches, such as NG-Meet and FHyMix. This outcome is expected, as NLSS-based methods excel at leveraging repetitive patterns across image patches, which aligns well with the spatial properties

of Gaussian noise. Therefore, for LRTSINR, a feasible approach is to aggregate similar patches together to form a fourth-order tensor, and then parameterize it using the LRTSINR method, so that LRTSINR can further utilize non-local similarity to improve the Gaussian noise denoising performance. While our method may not achieve the highest MPSNR scores in these specific cases, it still delivers competitive results—consistently surpassing most low-rank and local smoothness-based methods, as well as those requiring supervised training. This highlights LRTSINR as a robust and effective unsupervised alternative even in settings where NLSS techniques dominate.

Table 1. Quantitative comparison of all competing methods on *DC mall* dataset. The best and the second best results are highlighted in bold and italic underlined, respectively.

Noise Types	Metrics	Noisy	LRMR	LRTV	LRTD TV	CTV	RCTV	NG Meet	FHy Mix	RCI LD	S2DIP	LRT SINR
Case (a)	MPSNR	20.00	34.63	33.47	32.42	33.43	36.61	<u>38.05</u>	37.65	38.23	35.01	35.68
	MSSIM	0.5146	0.9627	0.9456	0.9318	0.9478	0.9736	<u>0.9833</u>	0.9811	0.9838	0.9637	0.9669
	MSAM	0.4754	0.1007	0.0883	0.0836	0.1147	0.0803	<u>0.0586</u>	0.0661	0.0579	0.0820	0.0767
	ERGAS	376.01	67.07	76.11	85.79	77.86	53.59	<u>45.03</u>	47.36	44.86	58.54	56.18
	Time(s)	–	21.78	79.19	116.80	58.26	4.07	35.10	0.49	<u>1.76</u>	15380	68.22
Case (b)	MPSNR	19.86	34.32	33.49	32.31	33.25	36.03	<u>37.05</u>	36.65	37.73	34.87	35.53
	MSSIM	0.5093	0.9615	0.9444	0.9307	0.9465	0.9714	<u>0.9799</u>	0.9771	0.9808	0.9625	0.9666
	MSAM	0.4810	0.1027	0.0909	0.0851	0.1175	0.0858	0.0666	0.0716	<u>0.0679</u>	0.0843	0.0772
	ERGAS	381.49	69.63	78.66	87.02	79.40	59.87	<u>53.32</u>	54.15	46.86	61.25	57.24
	Time(s)	–	22.50	78.48	112.60	55.36	3.84	34.81	0.30	<u>1.76</u>	15420	66.48
Case (c)	MPSNR	12.38	34.17	34.29	32.60	<u>34.72</u>	33.73	24.47	23.72	26.75	34.01	35.50
	MSSIM	0.2206	0.9566	0.9525	0.9357	<u>0.9615</u>	0.9216	0.8392	0.8263	0.8719	0.9371	0.9658
	MSAM	0.7334	0.0993	0.0992	0.1069	<u>0.0956</u>	0.0826	0.1817	0.1834	0.1556	0.0909	0.0825
	ERGAS	925.37	72.94	76.78	84.12	<u>67.38</u>	79.90	241.84	239.24	166.51	78.34	62.70
	Time(s)	–	22.25	89.24	113.63	56.90	3.75	36.33	0.29	<u>1.76</u>	15428	66.48
Case (d)	MPSNR	12.38	<u>34.63</u>	34.17	32.50	34.53	33.83	24.61	23.58	25.98	34.00	35.33
	MSSIM	0.2196	<u>0.9627</u>	0.9511	0.9345	0.9603	0.9288	0.8368	0.8231	0.8607	0.9399	0.9666
	MSAM	0.7351	<u>0.0904</u>	0.1027	0.1011	0.0975	0.0973	0.1803	0.1852	0.1584	0.1000	0.0788
	ERGAS	924.36	<u>68.71</u>	80.96	85.07	68.85	82.56	238.53	241.24	172.83	81.76	62.07
	Time(s)	–	22.33	78.84	113.10	56.05	3.71	35.14	0.29	<u>1.84</u>	15444	66.48
Case (e)	MPSNR	15.20	36.76	36.29	34.86	<u>37.86</u>	36.99	27.13	24.09	29.43	36.64	38.42
	MSSIM	0.3688	0.9757	0.9680	0.9627	<u>0.9819</u>	0.9691	0.8786	0.8328	0.9012	0.9685	0.9826
	MSAM	0.6674	0.0714	0.1305	<u>0.0659</u>	0.0656	0.0737	0.1730	0.3924	0.1323	0.1021	0.0655
	ERGAS	758.52	47.87	141.25	<u>65.47</u>	44.27	55.09	192.90	378.72	164.42	89.55	<u>47.53</u>
	Time(s)	–	27.58	81.36	120.01	58.26	3.82	36.06	0.28	<u>1.84</u>	15328	66.48

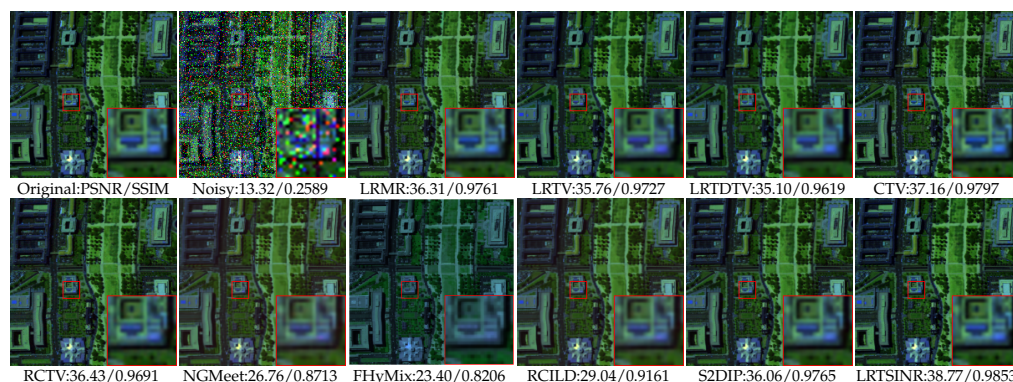
However, the performance landscape changes significantly under more complex and realistic noise conditions, such as impulse noise and deadlines—commonly present in practical HSI acquisition scenarios (cases (c) to (e)). In these challenging conditions, NLSS-based methods often struggle to simultaneously remove intricate noise patterns while preserving fine structural details. In contrast, some total variation (TV)-based techniques (e.g., LRTV, LRTDTV, E3DTV, RCTV) exhibit greater resilience to such noise. Most notably, LRTSINR demonstrates clear advantages in these scenarios, achieving the highest MPSNR scores among all competitors and preserving image structure more effectively. Compared to other unsupervised deep learning approaches such as S2DIP, our method shows superior capacity in modeling and mitigating complex noise while maintaining spectral–spatial fidelity.

Furthermore, while supervised learning-based methods offer fast inference and decent denoising quality, they do not achieve top performance in any noise setting. This limitation is likely due to their reliance on specific training data distributions, which may not generalize well to the diverse noise patterns in real-world HSI data. Nonetheless, their efficiency makes them valuable in time-sensitive applications.

Table 2. Quantitative comparison of all competing methods on *Cloth* dataset. The best and the second best results are highlighted in bold and italic underlined, respectively.

Noise Types	Metrics	Noisy	LRMR	LRTV	LRTD TV	CTV	RCTV	NG Meet	FHy Mix	RCI LD	S2DIP	LRT SINR
Case (a)	MPSNR	20.00	29.56	29.13	31.18	30.69	31.00	<u>35.56</u>	33.29	35.89	32.01	32.42
	MSSIM	0.7248	0.9426	0.9155	0.9430	0.9385	0.9414	<u>0.9820</u>	0.9670	0.9842	0.9427	0.9509
	MSAM	0.4159	0.1299	0.1111	0.0998	0.1103	0.1151	<u>0.0530</u>	0.0787	0.0512	0.1023	0.0841
	ERGAS	362.81	117.90	126.35	98.76	104.60	101.03	<u>58.76</u>	77.70	55.52	94.32	83.20
	Time(s)	–	52.19	242.00	228.35	88.04	10.84	226.72	<u>2.47</u>	2.32	6340	101.23
Case (b)	MPSNR	19.85	29.46	29.23	31.12	30.65	30.73	<u>34.71</u>	32.96	35.12	31.24	31.70
	MSSIM	0.7171	0.9405	0.9161	0.9424	0.9373	0.9361	<u>0.9769</u>	0.9633	0.9832	0.9378	0.9431
	MSAM	0.4226	0.1320	0.1044	0.1005	0.1110	0.1203	<u>0.0615</u>	0.0812	0.0542	0.1023	0.0873
	ERGAS	366.73	118.89	123.68	99.43	104.94	103.77	<u>65.77</u>	80.24	60.52	94.32	86.23
	Time(s)	–	47.25	241.36	213.33	77.77	9.10	218.92	<u>2.33</u>	2.32	6340	102.12
Case (c)	MPSNR	12.45	28.75	29.41	31.41	<u>31.61</u>	30.67	22.00	28.93	26.85	30.97	32.26
	MSSIM	0.3840	0.9262	0.9265	0.9478	<u>0.9524</u>	0.9334	0.7583	0.9210	0.8722	0.9432	0.9576
	MSAM	0.6848	0.1468	0.0991	0.0968	<u>0.0970</u>	0.1050	0.2605	0.1183	<u>0.0939</u>	0.1609	0.0899
	ERGAS	886.36	134.38	122.56	97.03	<u>94.12</u>	98.87	322.18	127.13	185.64	91.45	88.79
	Time(s)	–	52.60	257.13	212.85	78.15	8.64	229.11	2.21	<u>2.46</u>	6345	99.42
Case (d)	MPSNR	12.43	28.48	29.44	31.29	<u>31.58</u>	30.58	22.05	28.66	26.68	31.14	32.11
	MSSIM	0.3814	0.9227	0.9270	0.9467	<u>0.9522</u>	0.9363	0.7566	0.9136	0.8665	0.9406	0.9559
	MSAM	0.6888	0.1519	0.0970	0.0987	<u>0.0971</u>	0.1062	0.2633	0.1249	<u>0.0944</u>	0.1665	0.0908
	ERGAS	886.87	137.35	121.38	<u>98.38</u>	94.40	102.87	323.14	131.81	189.21	102.62	99.15
	Time(s)	–	50.81	276.65	279.24	125.40	15.19	390.59	2.14	<u>2.53</u>	6366	99.23
Case (e)	MPSNR	14.59	28.97	31.29	33.46	<u>34.80</u>	32.51	24.70	33.05	30.38	34.02	35.44
	MSSIM	0.4881	0.9152	0.9510	0.9687	<u>0.9739</u>	0.9547	0.8260	0.9626	0.9202	0.9698	0.9816
	MSAM	0.6275	0.1704	0.0943	0.0784	<u>0.0633</u>	0.0704	0.2457	0.1015	0.1418	0.0647	0.0616
	ERGAS	802.14	182.94	105.82	78.50	<u>59.79</u>	74.23	279.87	90.06	150.80	55.05	52.43
	Time(s)	–	115.25	402.63	264.09	79.45	9.38	245.58	2.02	<u>2.67</u>	6384	101.23

We present the pseudo-RGB visualizations of the restored HSI/MSI data under case (e) in Figures 3 and 4 to demonstrate the spatial denoising capabilities of all competing methods. A restoration result is considered superior if it aligns more closely with the ground truth in terms of spatial structure, texture detail, and color fidelity. In each subfigure, an identical subregion is highlighted and magnified four times to facilitate visual comparison. From these visual results, several observations can be made. First, all methods demonstrate the ability to reduce complex mixed noise to some extent. Second, beyond quantitative metrics, our method delivers notable visual advantages. It effectively removes impulse noise artifacts, and deadlines while preserving the original spatial texture and structure of the HSI, exhibiting minimal artifacts and color distortions.

**Figure 3.** Recovered false-color images (R: 130, G: 95, B: 14) on the *DC mall* dataset, with a spatial resolution of $200 \times 200 \times 160$, under noise case (e). The corresponding PSNR and SSIM values are shown at the bottom of each subfigure. For better visual comparison, a selected region of interest is enlarged by a factor of four.

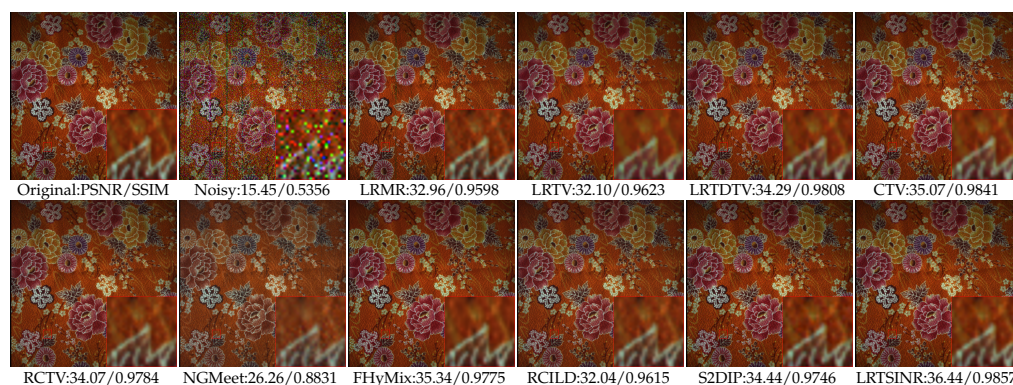


Figure 4. Recovered false images (R: 30, G: 21, B: 8) on *Cloth* of the ICVL dataset with a size of $512 \times 512 \times 31$ under noise case (e). The corresponding PSNR and SSIM values are shown at the bottom of each subfigure. For better visual comparison, a selected region of interest is enlarged by a factor of four.

Additionally, we display the restored spectral signatures at selected locations for all methods in case (e), as shown in Figures 5 and 6. In each spectral subfigure, the ground-truth spectral curve of the clean pixel is plotted alongside the curves recovered by different denoising methods. By comparing these curves, we observe that most methods are capable of effectively suppressing noise while maintaining the overall spectral shape. However, among all the methods, only our proposed approach achieves the closest alignment with the clean spectral curve, clearly demonstrating its superior performance in accurately restoring spectral information.

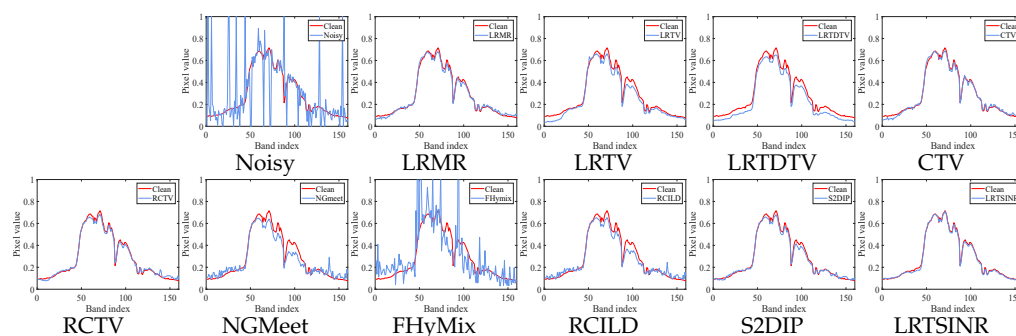


Figure 5. Recovered spectral curves under noise case (e) of all the competing methods at point (100, 100) on *DC mall* dataset. There are two curves in each sub-graph, one is the clean spectrum curve, and the other is the spectrum curve restored by denoising methods.

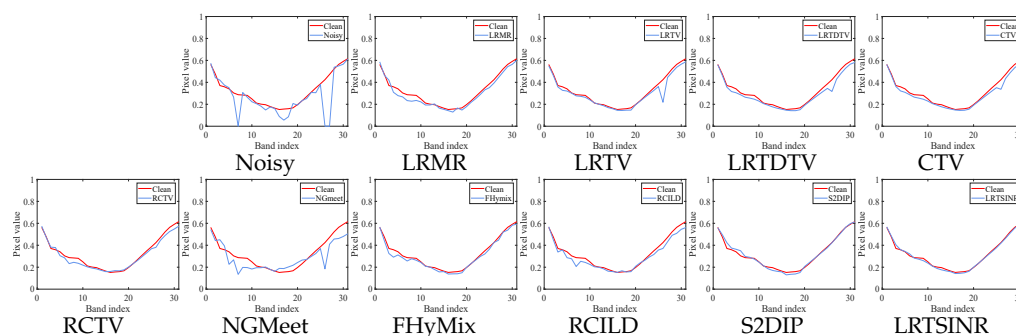


Figure 6. Recovered spectral curves under noise case (e) of all the competing methods at point (200, 200) on *Cloth* dataset. There are two curves in each sub-graph, one is the clean spectrum curve, and the other is the spectrum curve restored by denoising methods.

As observed in Tables 1 and 2, our LRTSINR algorithm does not achieve the highest performance metrics across all cases compared to supervised deep learning. Specifically, it cannot outperform supervised algorithms like RCILD or the plug-and-play method like FHyMix, primarily because these methods leverage pre-trained networks, eliminating the need for inference-time training and often resulting in faster execution. However, when the pre-trained model faces a distribution different from that for its training, its denoising performance drops significantly. This shows that although the supervised method has fast test performance, its results do not have better denoising performance due to generalization issues. Furthermore, our algorithm maintains a relatively fast processing speed when compared to other DIP-based unsupervised methods. A key advantage of LRTSINR becomes evident when contrasted with S2DIP. Our subspace-based modeling significantly reduces the computational burden by decreasing the size of the factors that need to be solved during the optimization process.

4.2. Real Data Experiments

To further verify the adaptability of our proposed method on real noise scenes, we executed denoising experiments on two real-world HSI datasets *Urbanpart* and *ZY01*. In addition to the methods used in the simulation experiments, we also introduced the GLF method in real-world experiments, which, like NGMeet and FHyMix, is based on the nonlocal similarity approach. Parts of the spectra of the two HSIs are heavily polluted by atmosphere, water, and some other physical factors, bringing in severe complex noise, including impulse noise, stripe noise, and deadlines. Given the lack of ground truth, it is impractical to quantitatively evaluate all competing methods. Instead, we qualitatively assess the denoising results by showcasing the reconstruction images as pseudo-RGB images of severely affected bands and displaying the spectral curve recovery.

The recovery results for heavily degraded spectral bands are presented in Figures 7 and 8. From these visualizations, it is evident that although all competing methods alleviate real noise to some extent, our proposed approach achieves the most visually appealing results—recovering sharper and more well-defined edge structures. The enlarged subregions further highlight that our method not only effectively suppresses impulse noise and stripe artifacts but also preserves intricate spatial textures, a task that remains challenging for many traditional methods, especially those that do not fully leverage the inherent local smoothness of HSI data.

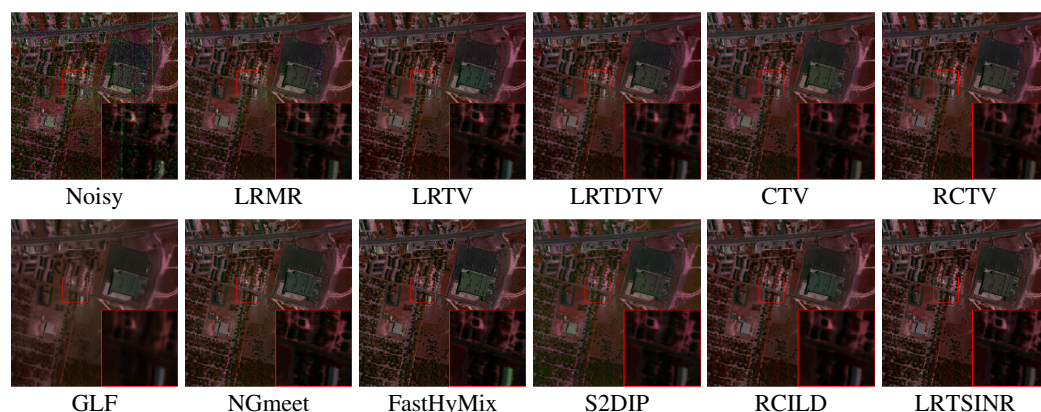


Figure 7. Recovered false images (R: 132, G: 207, B: 155) on *Urbanpart* dataset with a size of $307 \times 307 \times 210$. For better visual comparison, a selected region of interest is enlarged by a factor of three.

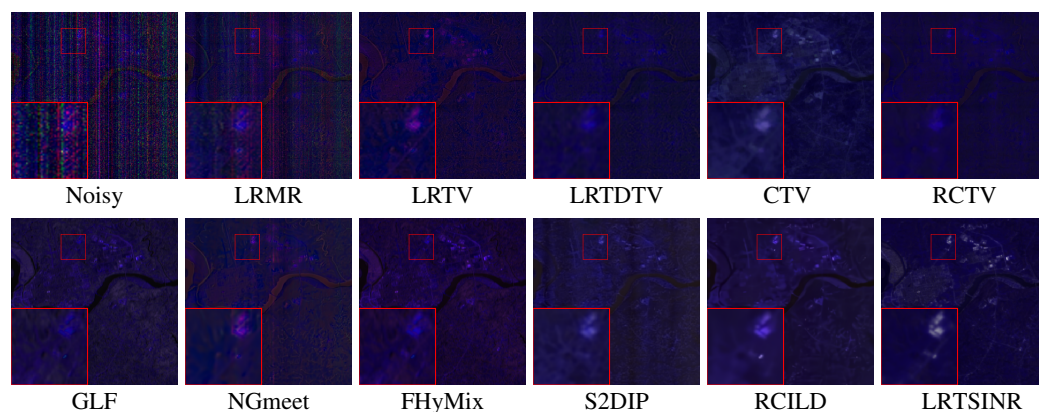


Figure 8. Recovered false images (R: 165, G: 166, B: 132) on ZY01 dataset with a size of $300 \times 300 \times 166$. For better visual comparison, a selected region of interest is enlarged by a factor of three.

In addition to spatial reconstruction, preserving the intrinsic trend of the spectral signature is a crucial criterion for evaluating denoising performance. An effective denoising algorithm should maintain essential spectral priors, such as local continuity, even in the presence of severe noise-induced distortions. As illustrated in Figures 9 and 10, the spectral curves reconstructed by our method exhibit strong local continuity in heavily degraded regions while closely aligning with the reference spectra in relatively clean bands. These results underscore the strength of our approach in maintaining spectral fidelity and demonstrate its robustness against complex, real-world noise.

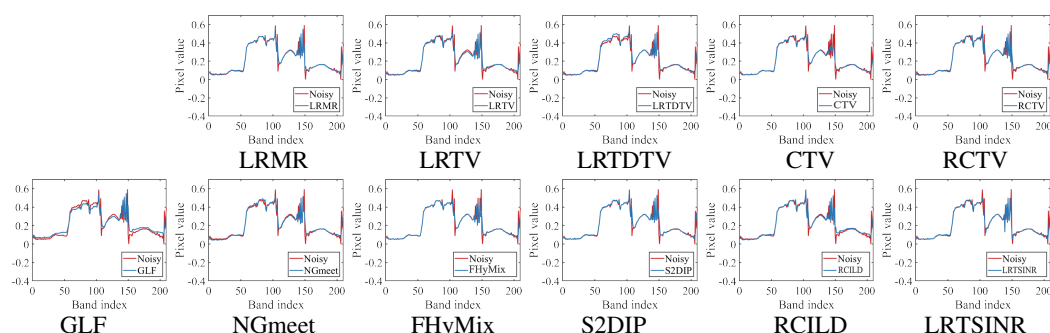


Figure 9. Recovered spectral curves under noise case (e) of all the competing methods at point (80, 120) on Urbanpart dataset. There are two curves in each sub-graph, one is the noisy spectrum curve, and the other is the spectrum curve restored by denoising methods.

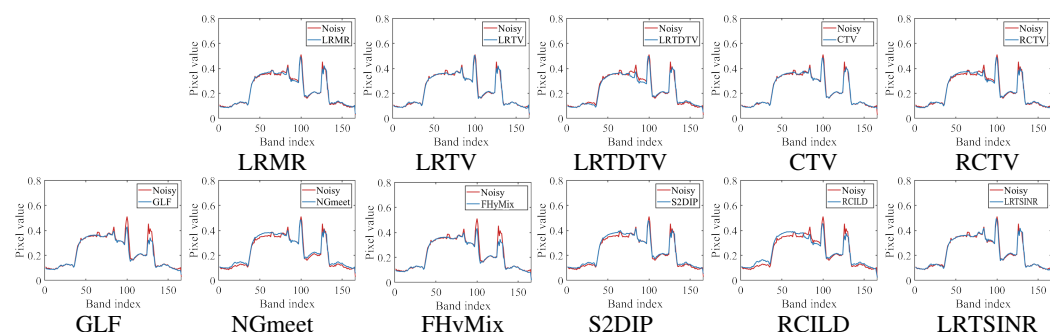


Figure 10. Recovered spectral curves under noise case (e) of all the competing methods at point (100, 100) on ZY01 dataset. There are two curves in each sub-graph, one is the noisy spectrum curve, and the other is the spectrum curve restored by denoising methods.

5. Conclusions and Future Work

This paper introduces LRTSINR, a unified framework for removing mixed noise from hyperspectral images. Unlike traditional methods that flatten HSI tensors, risking spatial structure disruption, LRTSINR directly operates on the 3D HSI data. It employs Tucker decomposition to model low-rank structures across spatial and spectral dimensions, enabling more accurate spectral prior representation. To capture continuity priors in subspace factor matrices, implicit neural representations adaptively model each factor column's continuity via coordinate-based neural networks, facilitating precise and data-adaptive spatial prior encoding. Total variation regularization is also incorporated to enhance robustness and spatial smoothness. All parameters are learned unsupervised from observed data without external ground truth.

Extensive experiments on synthetic and real-world datasets demonstrate LRTSINR's superiority over state-of-the-art methods in both quantitative metrics and visual quality. Future work will explore advanced regularization strategies and neural architectures to improve subspace prior modeling, and extend LRTSINR to broader HSI restoration tasks like inpainting and super-resolution.

Author Contributions: Conceptualization, C.C. and J.P.; Methodology, J.P.; Validation, C.C., D.S., Y.Y. and Z.G.; Formal analysis, D.S.; Investigation, C.C.; Data curation, C.C.; Writing—original draft, C.C.; Writing—review & editing, D.S. and J.P.; Visualization, D.S. and Y.Y.; Supervision, J.P.; Funding acquisition, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was in part by the National Natural Science Foundation of China under Grant 12401674, the Guangdong Basic and Applied Basic Research Foundation under Grant 2025A1515011453, the Fundamental Research Funds for the Central Universities under Grant D5000240095.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Cao, X.; Yao, J.; Xu, Z.; Meng, D. Hyperspectral image classification with convolutional neural network and active learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4604–4616. [\[CrossRef\]](#)
2. Su, Y.; Gao, L.; Jiang, M.; Plaza, A.; Sun, X.; Zhang, B. NSCKL: Normalized spectral clustering with kernel-based learning for semisupervised hyperspectral image classification. *IEEE Trans. Cybern.* **2022**, *53*, 6649–6662. [\[CrossRef\]](#)
3. Jiang, M.; Su, Y.; Gao, L.; Plaza, A.; Zhao, X.L.; Sun, X.; Liu, G. GraphGST: Graph generative structure-aware transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5504016. [\[CrossRef\]](#)
4. Su, Y.; Chen, J.; Gao, L.; Plaza, A.; Jiang, M.; Xu, X.; Sun, X.; Li, P. ACGT-Net: Adaptive cuckoo refinement-based graph transfer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5521314. [\[CrossRef\]](#)
5. Dong, W.; Fu, F.; Shi, G.; Cao, X.; Wu, J.; Li, G.; Li, X. Hyperspectral image super-resolution via non-negative structured sparse representation. *IEEE Trans. Image Process.* **2016**, *25*, 2337–2352. [\[CrossRef\]](#) [\[PubMed\]](#)
6. He, W.; Wang, M.; Chen, Y.; Zhang, H. An Unsupervised Dehazing Network with Hybrid Prior Constraints for Hyperspectral Image. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5514715. [\[CrossRef\]](#)
7. Zhang, Q.; Yuan, Q.; Li, J.; Li, Z.; Shen, H.; Zhang, L. Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning. *ISPRS J. Photogramm. Remote Sens.* **2020**, *162*, 148–160. [\[CrossRef\]](#)
8. Zhang, L.; Wei, W.; Zhang, Y.; Li, F.; Shen, C.; Shi, Q. Hyperspectral compressive sensing using manifold-structured sparsity prior. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 3550–3558.
9. Yao, J.; Meng, D.; Zhao, Q.; Cao, W.; Xu, Z. Nonconvex-sparsity and nonlocal-smoothness-based blind hyperspectral unmixing. *IEEE Trans. Image Process.* **2019**, *28*, 2991–3006. [\[CrossRef\]](#)
10. Goetz, A.F. Three decades of hyperspectral remote sensing of the Earth: A personal view. *Remote Sens. Environ.* **2009**, *113*, S5–S16. [\[CrossRef\]](#)
11. Landgrebe, D. Hyperspectral image data analysis. *IEEE Signal Process. Mag.* **2002**, *19*, 17–28. [\[CrossRef\]](#)
12. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [\[CrossRef\]](#)

13. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. Robust principal component analysis? *J. ACM (JACM)* **2011**, *58*, 1–37. [\[CrossRef\]](#)
14. 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. [\[CrossRef\]](#)
15. Lu, C.; Feng, J.; Chen, Y.; Liu, W.; Lin, Z.; Yan, S. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 925–938. [\[CrossRef\]](#)
16. Peng, Y.; Meng, D.; Xu, Z.; Gao, C.; Yang, Y.; Zhang, B. Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 2949–2956.
17. Peng, J.; Xie, Q.; Zhao, Q.; Wang, Y.; Yee, L.; Meng, D. Enhanced 3DTV regularization and its applications on HSI denoising and compressed sensing. *IEEE Trans. Image Process.* **2020**, *29*, 7889–7903. [\[CrossRef\]](#)
18. Zhang, H.; Liu, L.; He, W.; Zhang, L. Hyperspectral image denoising with total variation regularization and nonlocal low-rank tensor decomposition. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3071–3084. [\[CrossRef\]](#)
19. He, W.; Zhang, H.; Zhang, L.; Shen, H. Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration. *IEEE Trans. Geosci. Remote Sens.* **2015**, *54*, 178–188. [\[CrossRef\]](#)
20. Wang, Y.; Peng, J.; Zhao, Q.; Leung, Y.; Zhao, X.L.; Meng, D. Hyperspectral image restoration via total variation regularized low-rank tensor decomposition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *11*, 1227–1243. [\[CrossRef\]](#)
21. Xue, J.; Zhao, Y.Q.; Bu, Y.; Liao, W.; Chan, J.C.W.; Philips, W. Spatial-spectral structured sparse low-rank representation for hyperspectral image super-resolution. *IEEE Trans. Image Process.* **2021**, *30*, 3084–3097. [\[CrossRef\]](#)
22. Peng, J.; Wang, H.; Cao, X.; Jia, X.; Zhang, H.; Meng, D. Stable Local-Smooth Principal Component Pursuit. *SIAM J. Imaging Sci.* **2024**, *17*, 1182–1205. [\[CrossRef\]](#)
23. Xue, J.; Zhao, Y.; Wu, T.; Chan, J.C.W. Tensor convolution-like low-rank dictionary for high-dimensional image representation. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 13257–13270. [\[CrossRef\]](#)
24. Xie, Q.; Zhao, Q.; Meng, D.; Xu, Z. Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1888–1902. [\[CrossRef\]](#)
25. Xue, J.; Zhao, Y.; Liao, W.; Chan, J.C.W. Nonlocal low-rank regularized tensor decomposition for hyperspectral image denoising. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5174–5189. [\[CrossRef\]](#)
26. 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**, *59*, 10438–10454. [\[CrossRef\]](#)
27. 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/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6868–6877.
28. Zhuang, L.; Bioucas-Dias, J.M. 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. [\[CrossRef\]](#)
29. He, W.; Yao, Q.; Li, C.; Yokoya, N.; Zhao, Q.; Zhang, H.; Zhang, L. Non-local meets global: An iterative paradigm for hyperspectral image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 2089–2107. [\[CrossRef\]](#)
30. Peng, J.; Wang, H.; Cao, X.; Liu, X.; Rui, X.; Meng, D. Fast noise removal in hyperspectral images via representative coefficient total variation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5546017. [\[CrossRef\]](#)
31. Hong, D.; Yokoya, N.; Chanussot, J.; Zhu, X.X. An augmented linear mixing model to address spectral variability for hyperspectral unmixing. *IEEE Trans. Image Process.* **2018**, *28*, 1923–1938. [\[CrossRef\]](#)
32. Peng, J.; Wang, H.; Cao, X.; Zhao, Q.; Yao, J.; Zhang, H.; Meng, D. Learnable Representative Coefficient Image Denoiser for Hyperspectral Image. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5506516. [\[CrossRef\]](#)
33. Cao, X.; Fu, X.; Xu, C.; Meng, D. Deep spatial-spectral global reasoning network for hyperspectral image denoising. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5504714. [\[CrossRef\]](#)
34. Wei, K.; Fu, Y.; Huang, H. 3-D quasi-recurrent neural network for hyperspectral image denoising. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 363–375. [\[CrossRef\]](#)
35. Yuan, Q.; Zhang, Q.; Li, J.; Shen, H.; Zhang, L. Hyperspectral image denoising employing a spatial-spectral deep residual convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1205–1218. [\[CrossRef\]](#)
36. Chang, Y.; Yan, L.; Fang, H.; Zhong, S.; Liao, W. HSI-DeNet: Hyperspectral image restoration via convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 667–682. [\[CrossRef\]](#)
37. Xiong, F.; Zhou, J.; Tao, S.; Lu, J.; Zhou, J.; Qian, Y. SMDS-Net: Model guided spectral-spatial network for hyperspectral image denoising. *IEEE Trans. Image Process.* **2022**, *31*, 5469–5483. [\[CrossRef\]](#)
38. Bodrito, T.; Zouaoui, A.; Chanussot, J.; Mairal, J. A trainable spectral-spatial sparse coding model for hyperspectral image restoration. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 5430–5442.
39. Zhang, Q.; Zhu, J.; Dong, Y.; Zhao, E.; Song, M.; Yuan, Q. 10-minute forest early wildfire detection: Fusing multi-type and multi-source information via recursive transformer. *Neurocomputing* **2025**, *616*, 128963. [\[CrossRef\]](#)

40. Zhuang, L.; Ng, M.K. FastHyMix: Fast and parameter-free hyperspectral image mixed noise removal. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 4702–4716. [\[CrossRef\]](#)
41. Jiang, T.X.; Zhuang, L.; Huang, T.Z.; Zhao, X.L.; Bioucas-Dias, J.M. Adaptive hyperspectral mixed noise removal. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5511413.
42. Sidorov, O.; Yngve Hardeberg, J. Deep Hyperspectral Prior: Single-Image Denoising, Inpainting, Super-Resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Seoul, Republic of Korea, 27 October–2 November 2019.
43. Miao, Y.C.; Zhao, X.L.; Fu, X.; Wang, J.L.; Zheng, Y.B. Hyperspectral Denoising Using Unsupervised Disentangled Spatospectral Deep Priors. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5513916. [\[CrossRef\]](#)
44. Zhang, Q.; Yuan, Q.; Song, M.; Yu, H.; Zhang, L. Cooperated spectral low-rankness prior and deep spatial prior for HSI unsupervised denoising. *IEEE Trans. Image Process.* **2022**, *31*, 6356–6368. [\[CrossRef\]](#)
45. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Deep image prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake, UT, USA, 18–22 June 2018; pp. 9446–9454.
46. Luo, Y.S.; Zhao, X.L.; Jiang, T.X.; Zheng, Y.B.; Chang, Y. Hyperspectral mixed noise removal via spatial-spectral constrained unsupervised deep image prior. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 9435–9449. [\[CrossRef\]](#)
47. Shi, K.; Peng, J.; Gao, J.; Luo, Y.; Xu, S. Hyperspectral Image denoising via Double Subspace Deep Prior. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5531015. [\[CrossRef\]](#)
48. Gu, S.; Zhang, L.; Zuo, W.; Feng, X. Weighted nuclear norm minimization with application to image denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2862–2869.
49. Gu, S.; Xie, Q.; Meng, D.; Zuo, W.; Feng, X.; Zhang, L. Weighted nuclear norm minimization and its applications to low level vision. *Int. J. Comput. Vis.* **2017**, *121*, 183–208. [\[CrossRef\]](#)
50. Lu, C.; Feng, J.; Chen, Y.; Liu, W.; Lin, Z.; Yan, S. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5249–5257.
51. Rasti, B.; Ulfarsson, M.O.; Ghamisi, P. Automatic hyperspectral image restoration using sparse and low-rank modeling. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2335–2339. [\[CrossRef\]](#)
52. Chen, Y.; Cao, X.; Zhao, Q.; Meng, D.; Xu, Z. Denoising hyperspectral image with non-iid noise structure. *IEEE Trans. Cybern.* **2017**, *48*, 1054–1066. [\[CrossRef\]](#) [\[PubMed\]](#)
53. Peng, J.; Wang, Y.; Zhang, H.; Wang, J.; Meng, D. Exact decomposition of joint low rankness and local smoothness plus sparse matrices. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5766–5781. [\[CrossRef\]](#)
54. Maggioni, M.; Katkovnik, V.; Egiazarian, K.; Foi, A. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Trans. Image Process.* **2012**, *22*, 119–133. [\[CrossRef\]](#) [\[PubMed\]](#)
55. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 4260–4268.
56. 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. [\[CrossRef\]](#)
57. Pang, L.; Gu, W.; Cao, X. TRQ3DNet: A 3D quasi-recurrent and transformer based network for hyperspectral image denoising. *Remote Sens.* **2022**, *14*, 4598. [\[CrossRef\]](#)
58. Zhao, Y.; Zhai, D.; Jiang, J.; Liu, X. ADRN: Attention-based deep residual network for hyperspectral image denoising. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Virtually, 4–8 May 2020; pp. 2668–2672.
59. Shi, Q.; Tang, X.; Yang, T.; Liu, R.; Zhang, L. Hyperspectral image denoising using a 3-D attention denoising network. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 10348–10363. [\[CrossRef\]](#)
60. Xiong, F.; Zhou, J.; Zhao, Q.; Lu, J.; Qian, Y. MAC-Net: Model-aided nonlocal neural network for hyperspectral image denoising. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5519414. [\[CrossRef\]](#)
61. Li, M.; Fu, Y.; Zhang, Y. Spatial-spectral transformer for hyperspectral image denoising. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 1368–1376.
62. Rudin, L.I.; Osher, S.; Fatemi, E. Nonlinear total variation based noise removal algorithms. *Phys. D Nonlinear Phenom.* **1992**, *60*, 259–268. [\[CrossRef\]](#)
63. Wang, H.; Peng, J.; Cao, X.; Wang, J.; Zhao, Q.; Meng, D. Hyperspectral image denoising via nonlocal spectral sparse subspace representation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 5189–5203. [\[CrossRef\]](#)
64. Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; Wetzstein, G. Implicit neural representations with periodic activation functions. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7462–7473.

65. Luo, Y.; Zhao, X.; Li, Z.; Ng, M.K.; Meng, D. Low-rank tensor function representation for multi-dimensional data recovery. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *46*, 3351–3369. [[CrossRef](#)] [[PubMed](#)]
66. Zhang, G.; Ren, R.; Yan, X.; Zhang, H.; Zhu, Y. Effects of microplastics on dissipation of oxytetracycline and its relevant resistance genes in soil without and with *Serratia marcescens*: Comparison between biodegradable and conventional microplastics. *Ecotoxicol. Environ. Saf.* **2024**, *287*, 117235. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.