

# NSR-Net: Representation Model-Inspired Interpretable Deep Unfolding Network for Hyperspectral Image Classification

Xiangyu Nie<sup>ID</sup>, Zhaohui Xue<sup>ID</sup>, Member, IEEE, Hongjun Su<sup>ID</sup>, Senior Member, IEEE, and Jun Li<sup>ID</sup>, Fellow, IEEE

**Abstract**—Deep learning-based methods have demonstrated promising performance in hyperspectral image (HSI) classification. However, the black-box nature of deep learning poses a significant challenge in designing effective network architectures for HSI classification. To overcome this issue, this article presents a representation model-inspired interpretable deep unfolding network (NSR-Net). First, we formulate a deep-constrained nonnegative sparse representation (NSR) model with enhanced generalization ability to address the limitations of the prior-constrained NSR, i.e., its reliance on manual priors and specific assumptions. Second, the solving process for deep-constrained NSR is unfolded into a deep network, with each component of the network corresponding directly to a specific step. Finally, following the principle of representation model-based classification, a subdictionary reconstruction module (SDRM) is designed to determine the class label. In SDRM, each subdictionary is learned through a context-integrated training process, resulting in superior discriminative capability. In addition, to better guide NSR-Net optimization, we introduce a new composite loss function, which consists of constraint loss and residual loss, aiming to effectively recover representation coefficients and reconstruct data from the subdictionary. Experiments conducted on four distinct HSI datasets illustrate the superiority and generalization performance of the proposed method compared with advanced representation model-based and deep learning-based methods, with overall accuracy (OA) improvements of 0.72%–9.80%, 1.39%–8.09%, 0.40%–5.85%, and 0.56%–6.84% for Indian Pines, Salinas, LongKou, and Loukia, respectively. The source code will be available at: <https://github.com/ZhaohuiXue/NSR-Net>.

**Index Terms**—Classification, deep unfolding, hyperspectral image (HSI), interpretability, nonnegative sparse representation (NSR).

## I. INTRODUCTION

WITH the emergence of the new generation of satellite missions, comprising constellations featuring short

Received 12 September 2024; revised 5 November 2024 and 10 December 2024; accepted 17 December 2024. Date of publication 20 December 2024; date of current version 3 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 42271324, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20221506, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20240282. (*Corresponding author: Zhaohui Xue.*)

Xiangyu Nie is with the School of Earth Sciences and Engineering, Hohai University, Nanjing 211100, China.

Zhaohui Xue and Hongjun Su are with the College of Geography and Remote Sensing, Hohai University, Nanjing 211100, China (e-mail: zhaohui.xue@hhu.edu.cn).

Jun Li is with the School of Computer Science and Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430078, China.

Digital Object Identifier 10.1109/TGRS.2024.3520639

revisit time, high-spatial resolution, and high-spectral resolution sensors, the amount of remote sensing images available has increased significantly [1]. In particular, hyperspectral image (HSI), which contains rich spectral information, serves as a reliable data source for precise ground object classification. The task of classifying HSI involves assigning a class label to each pixel. By analyzing the response of each pixel across hundreds of narrow spectral bands, HSI offers more refined detail and higher precision compared with traditional multispectral imaging [2]. Currently, research on HSI classification has garnered significant interest across various fields, including agricultural management, forestry monitoring, mineral exploration, water resource management, and more [3].

In past decades, the rapid evolution of pattern recognition technology and machine learning algorithms, coupled with continuous hardware updates, has driven a steady improvement in the efficiency and accuracy of HSI classification [4], [5], [6], [7]. In the early stages, research on HSI classification mainly depended on traditional machine learning algorithms such as support vector machine (SVM) [8], extreme learning machine (ELM) [9], and random forest (RF) [10]. These methods were popular for their straightforward classification principles and ease of implementation. However, they often relied on manually designed features, leading to limited classification performance and making it challenging to handle high-dimensional data. Later, the emerging paradigm of compressed sensing has garnered significant attention in research on HSI classification using representation-based models [11]. These models focus on exploring the inherent low-dimensional structures embedded within the data, addressing the challenges of high-dimensional data while providing a more robust approach to HSI classification.

The representation models utilized for HSI classification mainly include collaborative representation (CR) [12], low-rank representation (LRR) [13], and the most popular sparse representation (SR) [14]. In a given overcomplete dictionary, the purpose of SR is to represent the target sample using as few atoms from the dictionary as possible. To achieve this, SR-related models generally incorporate a regularization term with either  $L_0$ -norm or  $L_1$ -norm constraint, aiming to minimize the number of nonzero elements in the representation coefficients. Numerous studies have demonstrated the effectiveness of SR for HSI classification [15], [16], [17]. For example, Liu et al. [18] proposed a class-specific sparse multiple kernel learning (CS-SMKL) framework, which combines SR with multikernel learning (MKL) for spatial–spectral joint classification of HSI.

Then, in [19], a class-guided coupled dictionary learning strategy was introduced into the SR model, which encourages pixels from the same class to have similar SR coefficients, thereby enhancing the discriminative power of SR. Furthermore, our previous works have significantly expanded SR by incorporating incremental dictionary learning [20] and introducing structural prior constraint [21], both of which have shown remarkable performance improvements. However, these representation model-based methods have certain drawbacks.

- 1) The solving process based on greedy algorithms or iterative optimization is highly time-consuming, especially when the dictionary contains a large number of atoms, which limits its practical application scenarios.
- 2) The dictionary constructed from labeled samples may not provide reliable discriminative capability, whereas traditional dictionary learning methods offer only limited performance improvements.
- 3) The model contains a series of parameters that need to be set manually, and the optimal parameter combinations may vary between different datasets, making the process of parameter tuning quite challenging.

In recent years, with the rapid development of deep learning in computer vision and data processing, researchers have applied it to the analysis and processing of HSI data, achieving impressive results [22], [23], [24], [25], [26]. For classification, deep learning approaches, such as convolutional neural network (CNN) [27], [28], [29], recurrent neural network (RNN) [30], [31], graph convolutional network (GCN) [32], [33], [34], and Transformer [35], [36], [37] have also been widely adopted. Specifically, in [38], a multistage information complementary fusion network was proposed to effectively regularize different data combinations, achieving competitive classification performance. To address the reliance of deep learning networks on a large number of labeled samples, Lv et al. [39] proposed a nonparametric sample augmentation algorithm that mines potential samples around each initial sample, significantly enhancing the training performance and stability of the network. To fully exploit the advantages of different data sources and effectively utilize the mutual information among them, Gao et al. [40] designed a spatial–spectral perception network that jointly learns spatial and spectral information distribution. In addition, considering the limitations of single-source data, a multiscale interactive fusion network (MIFNet) was presented in [41] for the joint classification of HSI data and synthetic aperture radar (SAR) data. Furthermore, Wang et al. [42] introduced the representation-enhanced status replay network (RSRNet) to mitigate the accumulation of representation bias and classifier bias during the learning process, thereby improving the efficiency of utilizing complementary information from multisource data. More related research works can also be found in [43], [44], and [45]. Compared with representation model-based methods, the data-driven characteristics of deep learning provide significant advantages in processing nonlinear high-dimensional data, automatically extracting discriminative features, improving classification performance, and reducing the need for manual intervention [46]. For deep learning-based methods, the architecture of the network has a significant impact on the final classification performance. However, as a black-box technology, determining an interpretable and effective network

architecture for HSI classification remains a critical research challenge.

Recently, deep unfolding, also known as deep unrolling, has emerged as a promising method for designing interpretable deep networks [47], [48], [49]. In this study, to explore the potential of deep unfolding in HSI classification and to bridge the gap between representation models and deep learning, a representation model-inspired interpretable deep unfolding network (NSR-Net) is proposed. We first formulate a new deep-constrained nonnegative SR (NSR) model based on the prior-constrained NSR framework. Thanks to the powerful nonlinear learning capabilities of deep learning, it provides a generalization constraint for NSR without relying on specific assumptions. Subsequently, the learned iterative shrinkage–thresholding algorithm (LISTA) [50] is utilized to solve the model. The network architecture is designed based on the solving process, with each computational step strictly corresponding to a specific component within the network. Next, to ensure accurate class determination while maintaining network interpretability, we designed a subdictionary reconstruction module (SDRM) based on the principle of representation models classification. In SDRM, each subdictionary is optimized using training samples from the corresponding class, and the context-integrated learning approach enhances its discriminative capability. In addition, a new composite loss function incorporating constraint loss and residual loss is utilized to guide the optimization of model parameters.

The main contributions of our work can be summarized as follows.

- 1) A new NSR-Net is proposed for HSI classification. The network features a transparent architecture and achieves superior classification performance compared with existing representation model-based and deep learning-based methods.
- 2) We propose a deep-constrained NSR model, which provides a data-driven approach to establishing a generalization prior constraint for solving representation coefficients without depending on specific assumptions. Meanwhile, the iterative solution process of the model guides the architecture design of the network.
- 3) We design the SDRM based on the classification principle of representation models to build a complete end-to-end classification network. In SDRM, each subdictionary is learned from training samples with integrated contextual information, ensuring the discriminability of the dictionary.
- 4) A new composite loss function that combines constraint loss and residual loss is introduced to guide network parameters optimization, enabling effective representation coefficient recovery and subdictionary-based data reconstruction.

## II. RELATED WORKS

The main idea behind the deep unfolding framework is to guide the design of network architectures using physical models. It achieves this by unfolding the iterative optimization process of traditional algorithms into a layerwise structure within deep learning networks, allowing parameters to be automatically learned through training [51]. This approach has

gained attention because of its ability to provide interpretable, efficient, and tailored solutions.

- 1) Since each layer corresponds to an iteration of a traditional algorithm, deep unfolding networks maintain strong interpretability, explaining the decision-making process of the model.
- 2) Deep unfolding significantly reduces the number of iterations required to reach optimal solutions, improving the speed of convergence compared to traditional model-based methods.
- 3) The learnable parameters allow the model to adapt to specific tasks, balancing between the structured nature of optimization algorithms and the adaptability of deep learning networks.

In view of this, substantial studies based on deep unfolding framework have been exploited for HSI processing. For pansharpening, a deep interpretable iterative generative dual adversarial network (iGDNNet) was proposed in [52], which unfolds the iterative steps of the half-quadratic splitting (HQS) algorithm for the designed pansharpening model through deep unfolding technique. For denoising, in [53], a multitask SR (MTSR) model was first developed, and its iterative process was then unfolded into a deep denoising network consisting of three distinct submodules. Each submodule is connected to specific operations in the iterative optimization algorithm to maintain the interpretability of the network. For anomaly detection, Li et al. [54] introduced an interpretable HSI anomaly detection network called LRR-Net. The architecture of LRR-Net is guided by the iterative process of solving the LRR model using the alternating direction method of multipliers (ADMMs). In addition, deep unfolding techniques have also been utilized for multispectral image and HSI fusion [55], as well as for HSI super-resolution [56] and unsupervised change detection [57], achieving outstanding performance across these applications.

However, research on deep unfolding techniques in the field of HSI classification remains quite limited. Currently, the few existing classification methods mainly rely on additional information provided by light detection and ranging (LiDAR) data [58], while specifically designed deep unfolding methods for HSI data classification are relatively rare. To this end, this article combines the NSR model with deep learning through the deep unfolding approach, proposing NSR-Net for HSI classification.

### III. METHODOLOGY

In this section, a novel deep unfolding network NSR-Net is designed for HSI classification, motivated by the need to combine the interpretability of the NSR model with the powerful learning capabilities of deep learning. Our method enhances the generalization ability of the NSR model by introducing deep learning techniques. The solving process for this model is systematically unfolded into a deep network, where each layer corresponds directly to a specific step in the optimization process. To further improve the class determination, we introduce SDRM, which leverages a context-integrated training process to enhance the discriminative ability of the learned subdictionaries.

For an HSI dataset with  $C$  different classes, the great number of dimensions and significant redundancy among adjacent

bands result in decreased model classification performance and increased computational complexity [59]. Therefore, we first preprocess the original high-dimensional HSI data through principal component analysis (PCA) and extract the top  $b$  components for classification. Let  $\mathbf{y}_i \in \mathbb{R}^b$  be the  $i$ th pixel in HSI, and  $\mathbf{Y}_i \in \mathbb{R}^{b \times T}$  represent the patch data constructed with a patch size of  $T = t \times t$  centered at  $\mathbf{y}_i$ . Moreover,  $K$  represents the unfolding iterations of the network. Specifically, for  $\mathbf{y}_i$  and  $\mathbf{Y}_i$ , they can be regarded as 3-D tensor data with dimensions  $\mathbb{R}^{b \times 1 \times 1}$  and  $\mathbb{R}^{b \times t \times t}$  involved in convolutional operations. The graphical representation of NSR-Net is illustrated in Fig. 1.

#### A. Prior-Constrained NSR Model

From the perspective of data reconstruction, the objective function of the traditional NSR model is formulated as follows:

$$\min_{\mathbf{x}_i} \frac{1}{2} \|\mathbf{D}\mathbf{x}_i - \mathbf{y}_i\|_2^2 + \lambda \|\mathbf{x}_i\|_1, \text{ s.t. } \min(\mathbf{x}_i) \geq 0 \quad (1)$$

where  $\mathbf{D} \in \mathbb{R}^{b \times m}$  is the dictionary containing  $m$  atoms, typically constructed from labeled samples.  $\mathbf{x}_i$  stands for the coefficient vector and  $\lambda$  denotes the regularization parameter for the  $L_1$ -norm term. Specifically, the presence of the nonnegative constraint limits the expressive capability of heterogeneous atoms, ensuring that  $\mathbf{y}_i$  can be better approximated by homogeneous atoms within the dictionary [60]. Meanwhile, it effectively prevents different dictionary atoms from canceling each other out due to addition and subtraction operations.

It is widely acknowledged that the inherent spectral-spatial relationships in HSI can serve as prior information, effectively enhancing model classification performance [16], [17], [21]. Consequently, the NSR model considering the prior constraint can be formulated as follows:

$$\min_{\mathbf{x}_i} \frac{1}{2} \|\mathbf{D}\mathbf{x}_i - \mathbf{y}_i\|_2^2 + \lambda \|\mathbf{W}\mathbf{x}_i\|_1 + \mathcal{N}_+(\mathbf{x}_i). \quad (2)$$

In this equation,  $\mathbf{W}$  represents the transformation matrix constructed based on specific prior information and  $\mathcal{N}_+(\cdot)$  denotes the nonnegative constraint.

For HSI, pixels within a certain local range usually belong to the same type of land cover and exhibit similar spectral characteristics. Based on this, most existing methods tend to integrate contextual information to make better classification decisions [61], [62], [63]. Thus, considering patch data  $\mathbf{Y}_i$  as the fundamental processing unit, (2) can be described as follows:

$$\min_{\mathbf{X}_i} \frac{1}{2} \|\mathbf{D}\mathbf{X}_i - \mathbf{Y}_i\|_F^2 + \lambda \|\mathbf{W}\mathbf{X}_i\|_1 + \mathcal{N}_+(\mathbf{X}_i) \quad (3)$$

where  $\mathbf{X}_i \in \mathbb{R}^{m \times T}$  is the coefficient matrix. Through the algorithm of iterative shrinkage-thresholding (ISTA) [64], in the  $k$ th iteration, the objective function in (3) can be solved according to the following two steps:

$$\mathbf{R}_i^{(k)} = \mathbf{X}_i^{(k-1)} - \eta \mathbf{D}^T (\mathbf{D}\mathbf{X}_i^{(k-1)} - \mathbf{Y}_i) \quad (4)$$

$$\mathbf{X}_i^{(k)} = \min \frac{1}{2} \|\mathbf{X}_i - \mathbf{R}_i^{(k)}\|_F^2 + \lambda \|\mathbf{W}\mathbf{X}_i\|_1 + \mathcal{N}_+(\mathbf{X}_i) \quad (5)$$

where  $\mathbf{R}_i^{(k)} \in \mathbb{R}^{m \times T}$  represents the gradient updating result corresponding to the fidelity term  $(1/2)\|\mathbf{X}_i - \mathbf{R}_i^{(k)}\|_F^2$ , and  $\eta$  denotes the step size, which usually needs to be manually

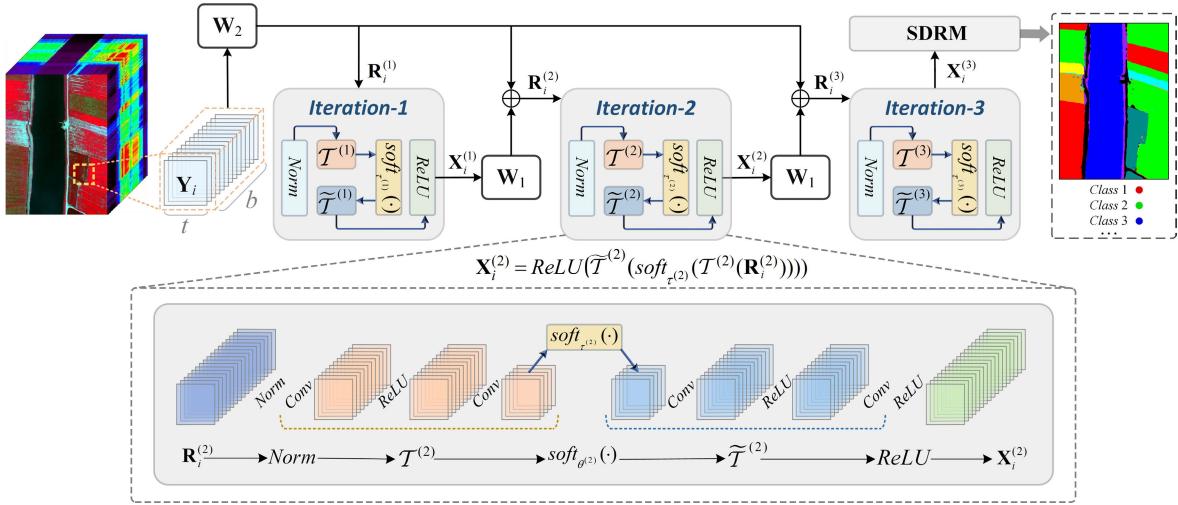


Fig. 1. Graphical illustration of the proposed NSR-Net for HSI classification (taking three unfolding iterations as an example).

configured. For (5), it can be solved by the soft thresholding operation.

However, for HSI classification, the above representation model-based approach has some drawbacks. First, it typically requires hundreds of iterations to solve, consuming a considerable amount of computation time. Second, some labeled samples may be affected by noise or represent mixed pixels, which may undermine the discriminative capability of the dictionary. Third, the transform matrix  $\mathbf{W}$  is a linear constraint designed under specific assumptions, lacking generalizability. Furthermore, determining the optimal combination of regularization parameters presents a significant challenge due to their reliance on manual empirical settings.

### B. Deep Unfolding Process

In this section, we describe how to leverage the prior-constrained NSR model to guide the fundamental architecture design of a deep network, which involves three main parts. The first part involves constructing a depth-constrained NSR model. The second part incorporates the concept of LISTA to optimize the model by introducing learnable variables with global parameter sharing. The third part aims to recover the coefficient matrix under the nonlinear transformation constraint.

1) *Deep-Constrained NSR Model*: We first establish a generalized nonlinear transformation  $T(\cdot)$  based on CNN to replace the manually designed transform matrix  $\mathbf{W}$  in (3). With this premise, the depth-constrained NSR model as follows:

$$\min_{\mathbf{X}_i} \frac{1}{2} \|\mathbf{D}\mathbf{X}_i - \mathbf{Y}_i\|_F^2 + \lambda \|\mathcal{T}(\mathbf{X}_i)\|_1 + \mathcal{N}_+(\mathbf{X}_i). \quad (6)$$

For each iteration, there exists its own  $\mathcal{T}(\cdot)$ , and  $\mathcal{T}^{(k)}(\cdot)$  represents the nonlinear transformation corresponding to the  $k$ th iteration. As shown in Fig. 1,  $\mathcal{T}^{(k)}(\cdot)$  comprises two bias-free convolutional layers, and a non-linear activation layer  $\text{ReLU}(\cdot)$ . Regarding the coefficient matrix  $\mathbf{X}_i$ , it can be regarded as a tensor with dimensions  $\mathbb{R}^{m \times t \times t}$ , directly involved in the convolutional operation.

2) *LISTA for Model Solving*: In the  $k$ th iteration, to solve the objective equation in (6),  $\mathbf{R}_i^{(k)}$  can still be obtained by (4).

Furthermore, the solving process in (4) can be further elaborated as follows:

$$\mathbf{R}_i^{(k)} = (\mathbf{I} - \eta \mathbf{D}^T \mathbf{D}) \mathbf{X}_i^{(k-1)} + \eta \mathbf{D}^T \mathbf{Y}_i \quad (7)$$

where  $\mathbf{I}$  is the identity matrix. Then, following the LISTA concept, the matrices  $\mathbf{I} - \eta \mathbf{D}^T \mathbf{D}$  and  $\eta \mathbf{D}^T$  in (7) can be substituted with two trainable transform matrices,  $\mathbf{W}_1 \in \mathbb{R}^{m \times m}$  and  $\mathbf{W}_2 \in \mathbb{R}^{m \times b}$

$$\mathbf{R}_i^{(k)} = \mathbf{W}_1 \mathbf{X}_i^{(k-1)} + \mathbf{W}_2 \mathbf{Y}_i. \quad (8)$$

In the proposed method, two bias-free convolutional layers with a kernel size of  $1 \times 1$  are utilized to construct  $\mathbf{W}_1$  and  $\mathbf{W}_2$ , and the parameters are shared across different iterations. Furthermore, to effectively accelerate the convergence of the network, a normalization layer  $\text{Norm}(\cdot)$  without any learnable parameters is introduced

$$\mathbf{R}_i^{(k)} = \text{Norm}(\mathbf{W}_1 \mathbf{X}_i^{(k-1)} + \mathbf{W}_2 \mathbf{Y}_i). \quad (9)$$

After normalization, the mean of elements within  $\mathbf{R}_i^{(k)}$  is 0, and the standard deviation is 1.

Subsequently, to obtain the coefficient matrix  $\mathbf{X}_i^{(k)}$ , (5) needs to be further refinement as follows:

$$\mathbf{X}_i^{(k)} = \min_{\mathbf{X}_i} \frac{1}{2} \left\| \mathbf{X}_i - \mathbf{R}_i^{(k)} \right\|_F^2 + \lambda \|\mathcal{T}^{(k)}(\mathbf{X}_i)\|_1 + \mathcal{N}_+(\mathbf{X}_i). \quad (10)$$

However, due to the presence of  $\mathcal{T}^{(k)}(\cdot)$ , it is not feasible to apply the soft threshold operation in (10). Thanks to the work in [65], which has demonstrated that under the nonlinear transformation  $\mathcal{T}(\cdot)$ , the following approximate linear mapping relationship can be established:

$$\|\mathcal{T}(\mathbf{X}_i) - \mathcal{T}(\mathbf{R}_i)\|_F^2 \approx \alpha \|\mathbf{X}_i - \mathbf{R}_i\|_F^2 \quad (11)$$

where  $\alpha$  is a constant determined by the parameters in  $\mathcal{T}(\cdot)$ . At this stage, combining (11), (10) can be further optimized as follows:

$$\begin{aligned} \mathcal{T}^{(k)}(\mathbf{X}_i^{(k)}) &= \min_{\mathcal{T}^{(k)}(\mathbf{X}_i)} \frac{1}{2} \left\| \mathcal{T}^{(k)}(\mathbf{X}_i) - \mathcal{T}^{(k)}(\mathbf{R}_i^{(k)}) \right\|_F^2 \\ &\quad + \alpha^{(k)} \lambda \|\mathcal{T}^{(k)}(\mathbf{X}_i)\|_1 + \mathcal{N}_+(\mathbf{X}_i). \end{aligned} \quad (12)$$

In (12), the regularization parameter  $\alpha^{(k)} \lambda$  can be considered as a whole, denoted by  $\tau^{(k)}$ . Furthermore, we seek for  $\tau^{(k)}$  to

TABLE I  
PARAMETER INFORMATION FOR EACH COMPONENT IN THE  $k$ TH UNFOLDING ITERATION

Iteration ( $k$ th)	Input Size	Operation	Output Size
$\text{Norm}(\cdot)$	$(m, t, t)$	$\text{Normalization}(\text{avg} = 0, \text{std} = 1)$	$(m, t, t)$
$\mathcal{T}^{(k)}(\cdot)$	$(m, t, t)$	$\text{Convolution}(\text{kernel\_size} = 3, \text{padding} = 1, \text{bias} = \text{false})$	$(128, t, t)$
	$(128, t, t)$	$\text{Activation}(\text{Function} = \text{ReLU})$	$(128, t, t)$
	$(128, t, t)$	$\text{Convolution}(\text{kernel\_size} = 5, \text{padding} = 2, \text{bias} = \text{false})$	$(64, t, t)$
$\text{soft}_{\tau^{(k)}}(\cdot)$	$(64, t, t)$	$\text{Soft-Thresholding}(\text{threshold} = \tau^{(k)})$	$(64, t, t)$
$\tilde{\mathcal{T}}^{(k)}(\cdot)$	$(64, t, t)$	$\text{Convolution}(\text{kernel\_size} = 5, \text{padding} = 2, \text{bias} = \text{false})$	$(128, t, t)$
	$(128, t, t)$	$\text{Activation}(\text{Function} = \text{ReLU})$	$(128, t, t)$
	$(128, t, t)$	$\text{Convolution}(\text{kernel\_size} = 3, \text{padding} = 1, \text{bias} = \text{false})$	$(m, t, t)$
$\text{ReLU}(\cdot)$	$(m, t, t)$	$\text{Activation}(\text{Function} = \text{ReLU})$	$(m, t, t)$

dynamically adjust its value across different iterations, instead of being manually fixed. Thus, we designate  $\tau^{(k)}$  for each iteration as an independent and trainable parameter

$$\mathcal{T}^{(k)}(\mathbf{X}_i^{(k)}) = \min_{\mathcal{T}^{(k)}(\mathbf{X}_i)} \frac{1}{2} \left\| \mathcal{T}^{(k)}(\mathbf{X}_i) - \mathcal{T}^{(k)}(\mathbf{R}_i^{(k)}) \right\|_F^2 + \tau^{(k)} \left\| \mathcal{T}^{(k)}(\mathbf{X}_i) \right\|_1 + \mathcal{N}_+(\mathbf{X}_i). \quad (13)$$

Here,  $\mathbf{R}_i^{(k)}$  can also be regarded as a tensor with dimension  $\mathbb{R}^{m \times t \times t}$  to participate in the convolutional operation.

In the absence of considering the nonnegative constraint  $\mathcal{N}_+(\cdot)$ , the closed-form solution for (13) can be expressed as follows:

$$\begin{aligned} & \mathcal{T}^{(k)}(\mathbf{X}_i^{(k)}) \\ &= \text{soft}_{\tau^{(k)}}(\mathcal{T}^{(k)}(\mathbf{R}_i^{(k)})) \\ &= \max(\mathcal{T}^{(k)}(\mathbf{R}_i^{(k)}) - \tau^{(k)}, 0) + \min(\mathcal{T}^{(k)}(\mathbf{R}_i^{(k)}) + \tau^{(k)}, 0) \end{aligned} \quad (14)$$

where  $\text{soft}_{\tau^{(k)}}(\cdot)$  represents the soft thresholding operation with  $\tau^{(k)}$  as the threshold at the  $k$ th iteration.

3) *Recovery of Representation Coefficients*: It is worth noting that the output of (14) is influenced by  $\mathcal{T}^{(k)}(\cdot)$  and cannot be utilized for data reconstruction. To recover the coefficient matrix, a nonlinear inverse transformation  $\tilde{\mathcal{T}}^{(k)}(\cdot)$  is defined, and a specific constraint loss is employed to ensure that it satisfies the reversibility condition  $R_i^{(k)} = \tilde{\mathcal{T}}^{(k)}(\mathcal{T}^{(k)}(R_i^{(k)}))$ . The definition of constraint loss can be found in Section III-D. As shown in Fig. 1,  $\tilde{\mathcal{T}}^{(k)}(\cdot)$  has a structure symmetric to  $\mathcal{T}^{(k)}(\cdot)$ , which also includes two bias-free convolutional layers and an ReLU( $\cdot$ ) activation layer. Hence, the representation coefficients can be recovered as follows:

$$\mathbf{X}_i^{(k)} = \tilde{\mathcal{T}}^{(k)}(\text{soft}_{\tau^{(k)}}(\mathcal{T}^{(k)}(\mathbf{R}_i^{(k)}))). \quad (15)$$

For the nonnegative constraint  $\mathcal{N}_+(\cdot)$ , which can be addressed by applying an additional ReLU( $\cdot$ ) layer

$$\mathbf{X}_i^{(k)} = \text{ReLU}(\tilde{\mathcal{T}}^{(k)}(\text{soft}_{\tau^{(k)}}(\mathcal{T}^{(k)}(\mathbf{R}_i^{(k)})))). \quad (16)$$

At this point, referring to Fig. 1, it can be observed that each component within the iteration module strictly corresponds to the solving steps described by (16). Thus, the forward computation process of each iteration in NSR-Net corresponds directly to solving the coefficient matrix in (6). In addition, the parameter information for each component within the  $k$ th iteration is detailed in Table I, and the pseudocode for the depth-constrained NSR model solving is outlined in Algorithm 1.

### Algorithm 1 Deep-Constrained NSR Model Solving

```

Input:  $\mathbf{D}, \mathbf{Y}_i, \lambda, \eta, K$ 
1 Define:  $\mathbf{W}_1 = \mathbf{I} - \eta \mathbf{D}^T \mathbf{D}, \mathbf{W}_2 = \mathbf{D}^T, \tau = \eta \lambda$ 
2 Initialize:  $\mathbf{X}_i = 0, \mathbf{R}_i = 0, k = 1$ 
3 while  $k \leq K$  do
    4 Step1: Calculate  $\mathbf{R}_i^{(k)}$  by Eq. (8)
    5 Step2: Normalize  $\mathbf{R}_i^{(k)}$  by Eq. (9)
    6 Step3: Calculate  $\mathcal{T}^{(k)}(\mathbf{X}_i^{(k)})$  by Eq. (14)
    7 Step4: Recover  $\mathbf{X}_i^{(k)}$  by Eq. (15)
    8 Step5: Solve  $\mathcal{N}_+(\cdot)$  constraint by Eq. (16)
    9 Step6: Update  $k$  by  $k = k + 1$ 
10 end
Output:  $\mathbf{X}_i^{(K)}$ 

```

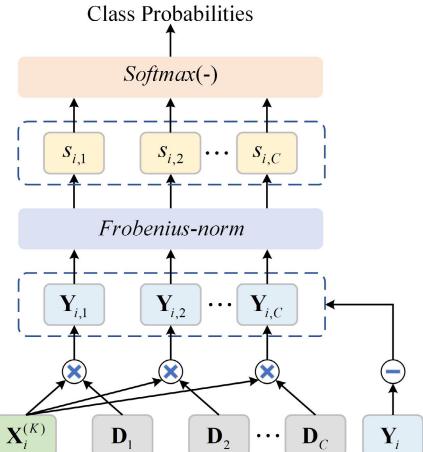


Fig. 2. Graphical illustration of the proposed SDRM.

### C. Subdictionary Reconstruction Module

In HSI analysis, the fundamental principle behind classification tasks employing SR or LRR-related methods is to reconstruct the HSI data using subdictionaries, and then calculate the corresponding residuals, also known as the reconstruction errors. The class associated with the minimum residual is then chosen as the label for the unknown sample. Leveraging this concept, SDRM is designed to establish a complete end-to-end classification network. A graphical illustration of the proposed SDRM can be found from Fig. 2.

First of all, we define the subdictionary set as  $\{\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_C\}$ . Then, the patch data  $\mathbf{Y}_{i,c}$  reconstructed

based on the  $c$ th class subdictionary can be obtained by

$$\mathbf{Y}_{i,c} = \mathbf{D}_c \mathbf{X}_i^K \quad (17)$$

where  $\mathbf{X}_i^K$  represents the coefficient matrix obtained after a maximum of  $K$  unfolding iterations. At this point, the residual matrix for the  $c$ th class can be expressed as  $\mathbf{Y}_i - \mathbf{Y}_{i,c}$ . Then, the residual value  $s_{i,c}$  is obtained by applying the Frobenius norm ( $\|\cdot\|_F$ ) to the residual matrix

$$s_{i,c} = \|\mathbf{Y}_i - \mathbf{Y}_{i,c}\|_F^2 = \|\mathbf{Y}_i - \mathbf{D}_c \mathbf{X}_i^K\|_F^2. \quad (18)$$

The residuals for each class can be represented as a vector  $[s_{i,1}, s_{i,2}, \dots, s_{i,C}]$ , denoted by  $\mathbf{s}_i$ . For  $s_{i,c}$  in  $\mathbf{s}_i$ , the smaller its value indicates that  $\mathbf{D}_c$  performs better in reconstructing  $\mathbf{Y}_i$ , implying a greater likelihood that  $\mathbf{Y}_i$  belongs to the class of  $c$ . To better align with the design rules of existing deep networks for HSI classification, we employ the softmax function to normalize the vector  $\mathbf{s}_i$

$$s'_{i,c} = \text{Softmax}(-s_{i,c}) = \frac{e^{-s_{i,c}}}{\sum_{j=1}^C e^{-s_{i,j}}} \quad (19)$$

where  $e$  is the natural constant and the result of the softmax operation on  $\mathbf{s}_i$  is represented as  $\mathbf{s}'_i$ .

It is important to note that in (19), the class probabilities are calculated using the negative residual values ( $-s_{i,c}$ ). The reason behind this is to make smaller residual values correspond to higher class probability outputs. In this case, the assigned class label for the central pixel of  $\mathbf{Y}_i$  can be determined as

$$\text{class}(y_i) = \max_{\text{index}}([s'_{i,1}, s'_{i,2} \dots s'_{i,C}]) \quad (20)$$

where  $\max_{\text{index}}(\cdot)$  represents the operation of returning the index corresponding to the maximum probability and  $s'_{i,c}$  is the  $c$ th element in  $\mathbf{s}'_i$ .

In SDRM, the parameters within each subdictionary are learnable. Consequently, during the training phase, samples of a specific class can individually optimize their corresponding subdictionaries. Furthermore, adopting patch-based data organization as the network input effectively combines the contextual information of the central pixel. This mitigates the negative impact on classification performance caused by some noisy or mixed labeled pixels and promotes the learning of more discriminative subdictionaries.

#### D. Loss Function of NSR-Net

For the proposed NSR-Net, its loss function  $\mathcal{L}_{\text{total}}$  consists of two parts including residual loss  $\mathcal{L}_{\text{res}}$  and constraint loss  $\mathcal{L}_{\text{const}}$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{res}} + \xi \mathcal{L}_{\text{const}} \quad (21)$$

where  $\xi$  is a weight utilized to balance the two losses.

The constraint loss is utilized to ensure the reversibility constraint between  $\mathcal{T}(\cdot)$  and  $\tilde{\mathcal{T}}(\cdot)$  within each iteration block of NSR-Net. Assuming that the number of training samples is  $N_{\text{tr}}$ , the constraint loss is

$$\mathcal{L}_{\text{const}} = \frac{1}{N_{\text{tr}}} \frac{1}{K} \sum_{i=1}^{N_{\text{tr}}} \sum_{k=1}^K \left\| \mathbf{R}_i^{(k)} - \tilde{\mathcal{T}}^{(k)} \left( \mathcal{T}^{(k)} \left( \mathbf{R}_i^{(k)} \right) \right) \right\|_F^2. \quad (22)$$

The residual loss is employed to minimize the feature discrepancy between the original data and its reconstructed data

corresponding to the consistent class. Moreover, as indicated in (19), the residuals for each class have been transformed into their respective probability outputs. Consequently, the residual loss can be formulated in the form of cross entropy

$$\mathcal{L}_{\text{res}} = -\frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} \sum_{c=1}^C \Phi(c) \log(s'_{i,c}) \quad (23)$$

where  $\Phi(\cdot)$  represents the one-hot encoding function, which outputs either 0 or 1.

## IV. EXPERIMENTAL RESULTS

### A. Hyperspectral Datasets

1) *Indian Pines*: The first dataset was acquired from the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor deployed over the Indian Pines test site in Northwestern Indian. It comprises  $145 \times 145$  pixels with a spatial resolution of 20 m and encompasses 220 bands ranging from 0.4 to 2.5  $\mu\text{m}$ . To mitigate the influence of noise and water absorption, only 200 bands were employed for subsequent experiments.

2) *Salinas*: The second dataset was acquired using the AVIRIS sensor over Salinas Valley, CA, USA. This dataset features a spatial resolution of 3.7 m, encompassing  $512 \times 217$  pixels. It comprises 224 bands within the wavelength range of 0.4–2.5  $\mu\text{m}$ . To account for water absorption effects, 20 bands were excluded, resulting in 204 bands employed for further analysis.

3) *LongKou*: The third dataset was captured by an 8-mm focal length Headwall Nano-Hyperspec imaging sensor mounted on a DJI Matrice 600 Pro (DJI M600 Pro) unmanned aerial vehicle (UAV) platform deployed over Longkou, Hubei, China. The dataset encompasses 270 spectral bands within the wavelength range of 0.4–1  $\mu\text{m}$ . It features a high spatial resolution of approximately 0.463 m and comprises  $550 \times 400$  pixels.

4) *Loukia*: The fourth dataset was obtained from the Hyperion sensor, carried by the National Aeronautics and Space Administration (NASA) Earth Observing 1 (EO-1) satellite. This dataset has a spatial resolution of 30 m, encompassing  $945 \times 176$  pixels. It comprises 176 bands within the wavelength range of 0.4–2.5  $\mu\text{m}$ .

The false-color composite images and ground-truth maps corresponding to the four datasets are presented in Fig. 3.

### B. Experimental Settings

1) *Comparison Methods*: The compared classification methods can be divided into two categories. The first category is the representation model-based methods, which include fusing correlation coefficient and joint SR (CCJSR) [66], locality-constrained SR (LSRC) [17], generalized tensor regression based on spatially derived features (GTR-SD) [67], tensor low-rank and SR (TLRSR) [20], and complementary spectral-spatial method (CSSM) [61]. The second category is the deep learning-based methods, which include deep feature fusion network (DFFN) [27], random patches network (RPNet) [68], spectral-spatial feature tokenization Transformer (SSFTT) [35], group-aware hierarchical Transformer (GAHT) [36], spatial-spectral ConvNeXt (SSCNX) [28].

TABLE II  
NUMBER OF TRAINING AND TESTING SAMPLES FOR FOUR HYPERSPECTRAL DATASETS

No.	Indian Pines			Salinas			LongKou			Loukia						
	Class	Train	Test	Total	Class	Train	Test	Total	Class	Train	Test	Total				
1	Alfalfa	3	43	46	Brocoli-weeds-1	11	1,998	2,009	Corn	70	34,441	34,511	Dense Urban Fabric	15	273	288
2	Corn-notill	72	1,356	1,428	Brocoli-weeds-2	19	3,707	3,726	Cotton	17	8,357	8,374	Mineral Extraction Sites	4	63	67
3	Corn-mintill	42	788	830	Fallow	10	1,966	1,976	Sesame	7	3,024	3,031	Non Irrigated Arable Land	28	514	542
4		12	225	237	Grass-pasture	7	1,387	1,394	Broad-leaf soybean	127	63,085	63,212	Buildings	4	75	79
5	Grass/pasture	25	458	483	Fallow-smooth	14	2,664	2,678	Narrow-leaf soybean	9	4,142	4,151	Olive Groves	71	1,330	1,401
6	Grass/trees	37	693	730	Stubble	20	3,939	3,959	Rice	24	11,830	11,854	Broad-leaved Forest	12	211	223
7	Grass/pasture-mowed	2	26	28	Celery	18	3,561	3,579	Water	135	66,921	67,056	Coniferous Forest	25	475	500
8	Hay-windrowed	24	454	478	Grapes-untrained	57	11,214	11,271	Roads and houses	15	7,109	7,124	Mixed Forest	54	1,018	1,072
9	Oats	1	19	20	Soil-vinyard-develop	32	6,171	6,203	Mixed weed	11	5,218	5,229	Dense Sclerophyllous Veg.	190	3,603	3,793
10	Soybean-notill	49	923	972	Corn-senesced-weeds	17	3,261	3,278					Space Sclerophyllous Veg.	141	2,662	2,803
11	Soybean-mintill	123	2,332	2,455	Lettuce-romaine-4wk	6	1,062	1,068					Sparsely Vegetated Areas	21	383	404
12	Soybean-clean	30	563	593	Lettuce-romaine-5wk	10	1,917	1,927					Rock & Sand	25	462	487
13	Wheat	11	194	205	Lettuce-romaine-6wk	5	911	916					Water	70	1,323	1,393
14	Woods	64	1,201	1,265	Lettuce-romaine-7wk	6	1,064	1,070					Coastal Water	23	428	451
15	Buildings-Grass-Trees-Drive	20	366	386	Vinyard-untrained	37	7,231	7,268								
16	Stone-Stell-Towers	5	88	93	Vinyard-vertical-trellis	10	1,797	1,807								
Total	-	520	9,729	10,249	-	279	53,850	54,129	-	415	204,127	204,542	-	683	12,820	13,503

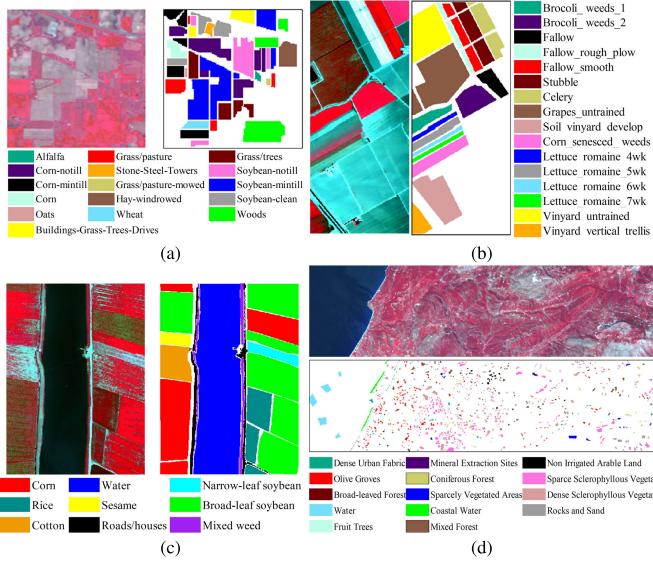


Fig. 3. Hyperspectral datasets. (a) False-color composite image (R:50, G:27, B:17) and ground-truth map for Indian Pines. (b) False-color composite image (R:50, G:27, B:17) and ground-truth map for Salinas. (c) False-color composite image (R:174, G:118, B:57) and ground-truth map for LongKou. (d) False-color composite image (R:40, G:22, B:14) and ground-truth map for Loukia.

2) *Training Settings*: We set training sample ratios according to the characteristics of each dataset to ensure fair and reliable results. As listed in Table II, Indian Pines and Loukia use 5% of labeled samples to construct the training set, while Salinas and LongKou use 0.5% and 0.2%, respectively. The training hyperparameters, including learning rate, number of epochs, and batch size, are set to  $5e-5$ , 200, and 8, respectively. The number of subdictionary atoms is set to 60. The balance parameter  $\xi$ , which is utilized to adjust the relative weights of the two losses in the total loss function, is empirically set to  $1e-2$ . Moreover, the optimizer is configured as Adam, and no data augmentation techniques are utilized during the training phase.

3) *Quantitative Metrics*: Three quantitative accuracy evaluation metrics, including overall accuracy (OA), average accuracy (AA), and kappa coefficient ( $\kappa$ ), are employed to assess the classification performance of each method. In addition, all presented experimental data represent the average of ten independent runs, with the training set for each run generated through random sampling.

4) *Running Platform*: All implementations are carried out using PyTorch-2.1.2 with CUDA-12.1 on a desktop PC equipped with Intel Core i7-13700KF CPU, NVIDIA GeForce RTX 4070ti GPU, and 64-GB RAM.

### C. Parameter Sensitiveness

In this section, we adopt a step-by-step optimization approach, where only one parameter is adjusted at a time, and other parameters are fixed at empirically determined or previously optimized values.

1) *Input Patch Size*: Input patch data with different sizes correspond to the utilization of spatial information within distinct local ranges, which has a significant influence on accuracy. Consequently, we first analyze the impact of patch size on classification performance, while dimension, learning rate, and unfolding depth are empirically set to 35,  $5e-5$ , and 2, respectively. As illustrated in Fig. 4(a), the range of patch sizes tested is from  $3 \times 3$  to  $17 \times 17$ . It is evident that as the patch size continues to increase, the OA of four datasets exhibits a trend of initially sharp improvement followed by a gradual decline. For LouKou and LouKia, the best performance is achieved when the patch size is  $10 \times 10$ . In contrast, Indian Pines and Salinas require consideration of spatial information from a larger range to achieve optimal results, with the highest accuracy achieved at patch sizes of  $12 \times 12$  and  $15 \times 15$ , respectively.

2) *Input Dimension*: Input patch data with different dimensions represent diverse degrees of utilization for spectral information, which also greatly influences the classification results. In this section, the number of dimensions ranged from 5 to 60, and the variation in classification accuracy as the input dimension increases is demonstrated in Fig. 4(b). Meanwhile, the patch size of each data is set to the optimal value, and the learning rate and unfolding depth remain set at  $5e-5$  and 2, respectively. Similar to the analysis results observed for Fig. 4(a), as the number of dimensions increases, the accuracy improves continuously. However, once the dimensions reach a certain threshold, the accuracy begins to decline. From the trends of the curves, it is evident that the optimal input dimensions for Indian Pines and Salinas are 50 and 15, respectively. For LongKou and LouKia, the optimal classification performance is achieved with input dimensions of 25 and 20, respectively.

3) *Learning Rate*: The learning rate controls the learning progress and convergence speed of deep learning-based

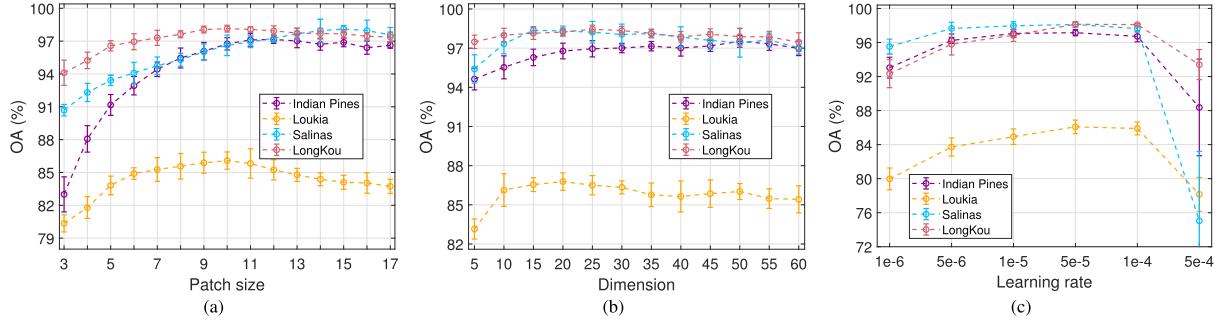


Fig. 4. Influence of (a) patch size, (b) dimension, and (c) learning rate on classification accuracy for four datasets.

TABLE III

IMPACT OF DIFFERENT UNFOLDING ITERATIONS ON CLASSIFICATION ACCURACY FOR FOUR DATASETS

Depth	Indian Pines	Salinas	LongKou	Loukia
2	97.52±0.45	<b>98.37±0.28</b>	<b>98.45±0.31</b>	<b>86.78±0.67</b>
3	97.25±0.50	96.65±1.06	98.05±0.35	85.18±0.92
4	<b>97.59±0.35</b>	97.72±0.54	97.89±0.55	85.64±0.71
5	97.16±0.45	97.63±0.55	98.30±0.44	85.95±1.24
6	97.30±0.55	97.27±0.80	97.98±0.28	85.99±0.73
7	97.41±0.42	97.37±0.68	98.30±0.50	86.64±0.58
8	97.12±0.64	97.88±0.55	98.42±0.35	86.31±0.91

methods, significantly impacting the performance and stability of NSR-Net. In this section, we analyze the impact of different learning rate settings on classification performance, and the values of learning rate are taken from a set of  $\{1e-6, 5e-6, 1e-5, 5e-5, 1e-4, 5e-4\}$ . At this stage, patch size and dimension for each HSI are set to their respective optimal values, while unfolding depth remains at 2. From Fig. 4(c), we can observe that a smaller learning rate prevents the method from converging within a limited number of epochs, while an excessively large learning rate seriously affects the stability of the method. Furthermore, under the condition of 200 epochs, four datasets achieve the optimal classification accuracy when the learning rate is set to  $5e-5$ . Evidently, this value serves as a good choice for the proposed NSR-Net.

4) *Unfolding Depth*: In this section, we aim to analyze the impact of different iterations of unfolding on the classification results. During the experiment, we gradually increase the number of unfolding iterations from 2 to 8, and other parameters are set according to the optimal values. The corresponding classification results are listed in Table III. Overall, the performance of NRS-Net is not positively correlated with the unfolding depth. For Salinas, LongKou, and Loukia, the optimal result is achieved when the unfolding depth is 2. Notably, while Indian Pines achieves its best performance at an unfolding depth of 4, the difference in accuracy compared to a depth of 2 is only 0.07%. The results in the table further demonstrate that the proposed classification method can achieve optimal discriminative performance without repeatedly stacking specific modules multiple times. This is mainly attributed to the effectiveness of the network architecture. Based on this observation, in subsequent experiments, the unfolding depth of NSR-Net is set to 2.

The parameters involved in this experiment, including patch size, dimension, learning rate, and unfolding depth, are independent of each other. Based on the parameter sensitivity

TABLE IV

OPTIMAL PARAMETER SETTINGS FOR DIFFERENT DATASETS

Parameter	Indian Pines	Salinas	LongKou	Loukia
Input Patch Size	12×12	15×15	10×10	10×10
Input Dimension	50	15	25	20
Learning Rate	5e-5	5e-5	5e-5	5e-5
Unfolding Depth	2	2	2	2

analysis, the optimized settings are summarized in Table IV. The experimental results indicate that while each parameter has an impact on classification performance, the sensitivity of the proposed method to different parameters varies significantly. In practical applications, the learning rate and unfolding depth can be set according to the corresponding experimental results, and the dimension can be uniformly set to 20. Notably, patch size has the greatest influence on the performance of NSR-Net, requiring adaptive adjustments based on different application scenarios.

#### D. Classification Results

1) *Indian Pines*: The classification accuracies for this dataset are in Table V. With 5% labeled samples, NSR-Net outperforms other methods. OA improved by 0.72%–9.80%, AA by 0.89%–18.83%, and  $\kappa$  by 0.81%–11.26%. Furthermore, NSR-Net achieves the highest accuracy in five classes, surpassing methods such as SSCNX, GAHT, and CSSM. The classification maps for each method on this dataset are displayed in Fig. 5. From these classification maps, it is evident that LSRC and RPNet perform poorly. In LSRC, numerous noise-like pixels are present, while RPNet is significantly affected by the class imbalance problem, resulting in a bias toward the class *Soybean-mintill*. Overall, NSR-Net achieves the highest accuracy and produces the most accurate classification map, demonstrating its effectiveness.

2) *Salinas*: Using 0.5% labeled samples, the performance of various methods is shown in Fig. 6 and Table VI. NSR-Net achieves the highest OA, AA, and  $\kappa$ , with improvements of approximately 1%–8%, 0.5%–6%, and 1.5%–9%, respectively. NSR-Net achieves the highest accuracy in three classes, while GTR-SD performs best in five classes. Although GTR-SD outperforms our method in more classes, the difference in accuracy between the two methods is negligible. (e.g., less than 0.3% for *Celery* and *Soil-vineyard-develop*). From the classification maps, it is evident that the predicted map generated by NSR-Net is smoother and more accurate compared with others,

TABLE V  
CLASSIFICATION ACCURACIES OF VARIOUS METHODS FOR INDIAN PINES DATASET (5% LABELED SAMPLES FOR TRAINING)

Class	Representation model-based					Deep learning-based					Ours	NSR-Net
	CCJSR [66]	LSRC [17]	GTR-SD [67]	TLRSR [20]	CSSM [61]	DFFN [27]	RPNet [68]	SSFTT [35]	GAHT [36]	SSCNX [28]		
Alfalfa	80.47±6.51	24.55±6.33	90.00±10.09	93.72±7.21	80.00±40.00	52.73±23.57	62.56±12.24	75.68±21.35	25.23±14.32	59.77±22.95	<b>97.91±1.93</b>	
Corn-notill	89.79±2.17	85.31±2.58	93.64±2.52	92.12±1.84	91.45±2.47	89.17±2.46	92.60±2.11	95.20±2.75	91.90±1.98	96.01±1.44	<b>97.68±0.89</b>	
Corn-mintill	89.01±3.60	73.45±2.06	93.29±2.78	88.67±3.76	94.04±3.33	91.23±3.35	88.02±4.30	94.63±2.99	94.70±3.03	96.35±2.31	<b>97.54±1.03</b>	
Corn	87.73±1.79	58.36±7.35	89.73±7.48	91.91±5.53	93.92±2.93	90.36±4.93	66.00±12.31	88.36±7.36	91.02±4.74	95.50±4.58	<b>96.80±2.09</b>	
Grass/pasture	92.58±1.42	93.12±1.54	92.25±5.66	94.89±1.82	95.50±2.32	94.03±2.91	90.72±4.44	94.16±4.15	<b>95.51±1.92</b>	95.22±1.36	93.78±4.01	
Grass/trees	93.94±1.49	97.99±0.81	98.67±1.14	98.77±0.96	97.49±1.23	96.13±1.22	96.08±2.01	<b>99.00±1.53</b>	98.76±0.61	98.56±0.92	98.12±1.56	
Grass/pasture-mowed	72.31±21.95	46.67±17.93	<b>92.69±8.68</b>	85.38±14.16	0.00±0.00	48.15±17.05	86.54±3.94	75.19±20.49	27.78±17.78	64.80±15.78	88.85±14.94	
Hay-windrowed	99.03±0.80	99.91±0.15	98.63±3.74	99.91±0.18	92.98±2.92	98.96±1.46	95.46±3.05	99.85±0.46	<b>100.00±0.00</b>	98.00±1.44	99.89±0.18	
Oats	24.21±18.71	20.53±9.25	24.74±14.52	<b>94.74±8.15</b>	0.00±0.00	55.26±27.58	33.68±10.04	68.42±16.48	12.11±9.13	53.39±27.59	63.68±20.31	
Soybean-notill	89.95±3.19	85.22±2.05	92.57±3.27	91.86±2.94	92.57±3.40	87.82±4.51	88.85±3.50	92.84±4.08	92.24±2.46	95.59±1.75	<b>96.74±2.12</b>	
Soybean-mintill	93.80±0.38	91.10±1.39	96.79±0.67	95.36±1.62	94.61±1.03	94.46±2.04	<b>98.80±0.71</b>	95.34±1.80	96.58±1.53	98.14±0.73	98.18±0.92	
Soybean-clean	84.94±3.96	83.82±1.73	91.81±2.67	88.21±2.61	90.87±5.04	81.92±6.24	82.08±5.60	87.83±5.30	91.07±2.40	<b>95.98±1.93</b>	94.53±2.83	
Wheat	85.57±4.28	95.44±4.24	95.67±2.30	98.76±2.42	<b>99.49±0.73</b>	97.13±2.35	89.07±12.00	98.82±2.26	95.74±3.68	99.36±0.69	97.63±3.57	
Woods	97.84±0.43	97.52±0.78	<b>99.79±0.18</b>	99.40±0.40	99.33±0.82	97.23±1.86	97.13±1.57	97.06±2.20	97.45±2.27	98.33±0.75	99.28±0.72	
Buildings-Grass-Trees-Drives	85.36±5.46	73.60±4.83	96.01±2.59	90.66±8.16	<b>99.49±0.53</b>	91.88±6.64	85.52±6.28	94.44±3.75	91.77±4.37	94.45±2.70	97.79±1.69	
Stone-Steel-Towers	82.05±7.27	83.64±7.26	95.00±3.10	92.95±4.42	<b>98.66±1.31</b>	92.95±8.02	40.34±13.18	91.70±7.82	82.73±8.67	96.04±1.32	93.07±6.36	
OA (%)	91.69±0.44	87.72±0.49	95.34±0.37	94.31±0.60	94.69±0.85	92.21±1.34	91.91±0.89	94.86±1.36	94.29±0.69	96.80±0.64	<b>97.52±0.45</b>	
AA (%)	84.28±2.08	75.64±2.14	90.08±1.30	93.58±1.67	82.53±2.79	84.96±2.38	80.84±1.31	90.53±2.84	80.29±2.07	89.72±3.09	<b>94.47±1.80</b>	
$\kappa \times 100$	90.52±0.50	85.91±0.58	94.68±0.42	93.51±0.68	93.93±0.98	91.12±1.53	90.68±1.04	94.15±1.55	93.48±0.78	96.36±0.73	<b>97.17±0.52</b>	

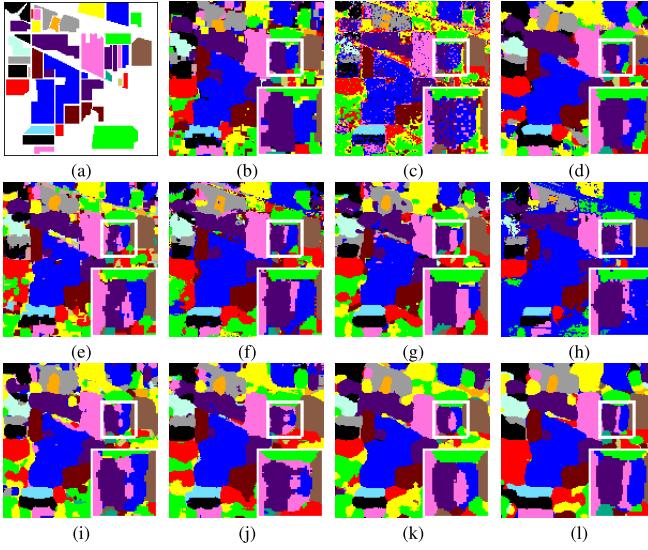


Fig. 5. Classification maps for Indian Pines with 5% labeled samples for training. (a) Ground truth, (b) CCJSR (91.69%), (c) LSRC (87.72%), (d) GTR-SD (95.34%), (e) TLRSR (94.31%), (f) CSSM (94.69%), (g) DFFN (92.21%), (h) RPNet (91.91%), (i) SSFTT (94.86%), (j) GAHT (94.29%), (k) SSCNX (96.80%), and (l) NSR-Net (ours, 97.52%).

especially for *Grapes-untrained* and *Vinyard-untrained*, which are prone to confusion.

3) *LongKou*: Table VII shows accuracy performance with 0.2% labeled data. NSR-Net again achieves the best results, with OA and AA improving by 0.40%–5.85% and 0.38%–21.98%, and  $\kappa$  by 0.53%–7.62%. In addition, NSR-Net achieves the highest accuracy in four of the nine classes. From the classification maps in Fig. 7, it can be observed that representation model-based methods generally produce less smooth classification maps compared to deep learning-based methods. This indicates that deep learning-based methods have a stronger ability to utilize contextual information.

4) *Loukia*: Table VIII shows classification accuracies for this dataset. Notably, NSR-Net outperforms other methods in OA and  $\kappa$ , with improvements of 0.56%–6.84% and 0.8%–8.13%, respectively. It is worth highlighting that CSSM achieves an impressive accuracy of 84.86% in AA and demonstrates the best accuracy in six different classes, showcasing its

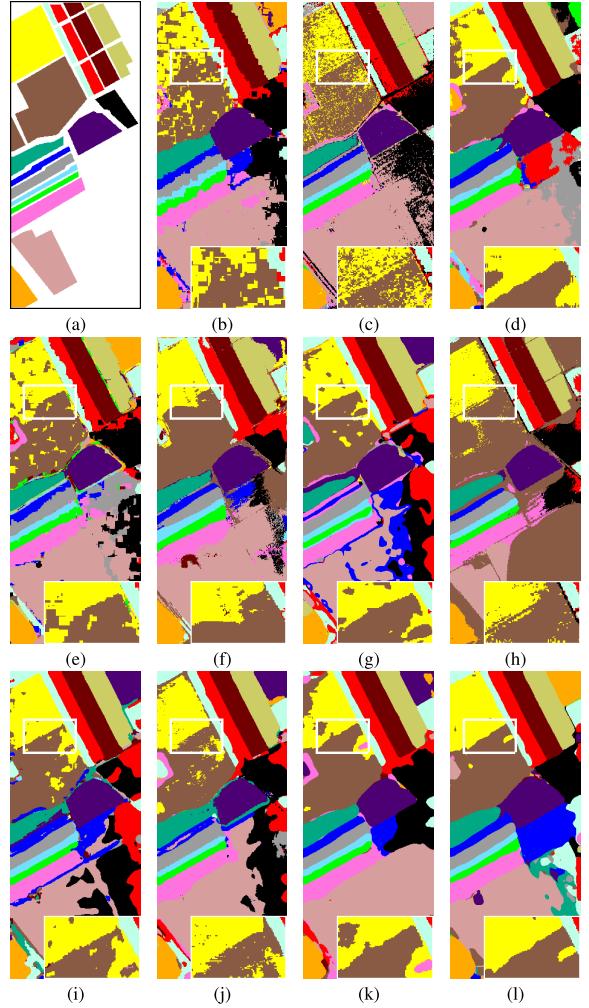


Fig. 6. Classification maps for Salinas with 0.5% labeled samples for training. (a) Ground truth, (b) CCJSR (90.32%), (c) LSRC (90.28%), (d) GTR-SD (94.34%), (e) TLRSR (94.66%), (f) CSSM (96.98%), (g) DFFN (91.84%), (h) RPNet (95.92%), (i) SSFTT (93.74%), (j) GAHT (94.96%), (k) SSCNX (95.79%), and (l) NSR-Net (ours, 98.37%).

competitiveness. The superior performance of complementary spectral–spatial method (CSSM) in achieving a high AA is attributed to its robust discriminative ability for the classes of



TABLE VIII  
CLASSIFICATION ACCURACIES OF VARIOUS METHODS FOR LOUKIA DATASET (5% LABELED SAMPLES FOR TRAINING)

Class	Representation model-based						Deep learning-based					Ours
	CCJSR [66]	LSRC [17]	GTR-SD [67]	TLRSR [20]	CSSM [61]	DFFN [27]	RPNNet [68]	SSFTT [35]	GAHT [36]	SSCNX [28]	NSR-Net	
Dense Urban Fabric	57.58±6.43	56.82±7.41	56.30±3.07	78.32±2.91	<b>78.79±8.51</b>	40.36±15.49	54.99±8.43	68.54±7.40	63.72±8.94	70.90±7.34	70.22±5.48	
Mineral Extraction Sites	98.41±2.46	73.44±8.89	93.97±6.50	98.10±3.46	<b>100.00±0.00</b>	88.59±10.10	32.96±10.69	84.38±11.9	65.31±24.86	90.24±6.80	97.78±3.34	
Non Irrigated Arable Land	61.98±4.10	83.17±3.94	81.09±2.66	79.16±3.96	<b>89.86±5.38</b>	75.96±4.65	74.07±5.45	85.94±4.01	84.41±4.93	86.19±3.52	84.65±4.70	
Fruit Trees	43.47±7.57	30.4±11.12	1.73±1.69	<b>53.33±12.95</b>	37.18±39.35	18.00±13.52	38.67±11.81	27.6±12.86	17.73±14.37	18.89±21.68	38.53±16.14	
Olive Groves	78.86±2.78	89.20±1.93	<b>93.29±1.65</b>	84.52±2.87	84.06±4.17	89.68±2.93	89.49±2.44	90.31±2.94	93.23±3.26	90.20±1.02	92.82±2.04	
Broad-leaved Forest	38.96±6.95	52.92±8.29	14.60±4.76	56.49±5.76	<b>89.10±15.54</b>	26.04±13.77	46.45±8.70	52.92±6.01	35.42±7.02	39.84±14.57	44.22±9.75	
Coniferous Forest	59.71±3.60	58.93±6.25	47.07±4.95	69.31±5.27	<b>88.81±6.65</b>	60.32±7.18	69.51±6.30	67.07±9.18	63.28±3.87	74.34±3.88	67.47±6.83	
Mixed Forest	81.22±4.15	63.52±2.65	68.97±3.38	83.52±2.88	82.07±2.41	67.47±6.30	82.23±3.71	78.93±3.72	71.49±5.90	73.16±3.98	<b>85.80±3.92</b>	
Dense Sclerophyllous Veg.	83.55±1.85	76.68±1.58	86.48±1.84	81.48±1.42	75.28±2.07	81.98±2.69	<b>89.34±1.62</b>	85.95±2.17	84.37±2.88	79.39±1.99	87.86±2.62	
Sparse Sclerophyllous Veg.	81.97±2.55	81.67±3.01	76.76±2.44	81.13±2.04	79.83±2.00	82.91±3.13	<b>88.00±1.69</b>	84.80±3.61	83.58±2.21	80.44±1.31	83.42±3.68	
Sparcely Vegetated Areas	81.88±3.55	69.06±5.76	57.28±8.29	85.25±5.51	<b>90.61±5.82</b>	63.20±9.36	81.93±5.05	69.11±7.49	75.78±8.14	79.46±5.45	84.41±4.86	
Rocks and Sand	79.00±1.11	91.04±1.94	89.57±2.50	92.90±1.63	92.52±5.87	92.38±1.42	85.46±4.32	93.89±2.39	91.79±2.67	<b>94.33±1.44</b>	92.25±3.33	
Water	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.98±0.07	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	
Coastal Water	85.00±6.63	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.37±0.47	99.98±0.07	99.72±0.84	99.25±1.80	99.74±0.29	99.74±0.37	99.74±0.54	99.95±0.09	
OA (%)	80.89±0.46	79.94±0.85	80.90±0.39	83.81±0.92	82.86±1.04	80.89±1.70	86.22±0.79	85.33±0.69	83.72±0.76	83.00±1.58	<b>86.78±0.67</b>	
AA (%)	73.68±0.73	73.35±1.41	69.08±0.64	81.63±1.45	<b>84.86±3.27</b>	70.47±3.73	73.74±1.71	77.80±1.61	73.56±2.93	76.94±3.54	80.67±1.56	
$\kappa \times 100$	77.10±0.57	76.11±1.00	76.94±0.48	80.83±1.11	79.35±1.29	77.14±2.11	83.44±0.97	82.54±0.83	80.57±0.90	79.70±1.90	<b>84.24±0.80</b>	

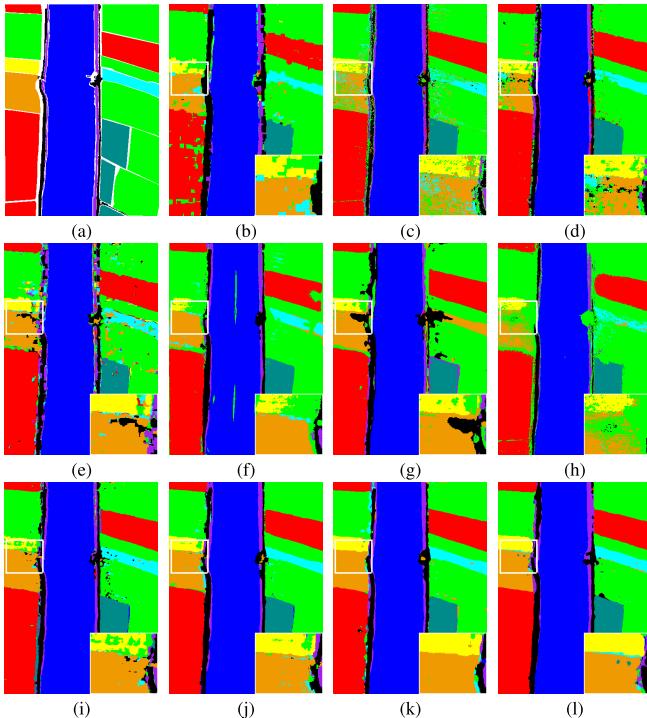


Fig. 7. Classification maps for LongKou with 0.2% labeled samples for training. (a) Ground truth, (b) CCJSR (80.89%), (c) LSRC (79.94%), (d) GTR-SD (80.90%), (e) TLRSR (83.81%), (f) CSSM (82.86%), (g) DFFN (80.89%), (h) RPNNet (86.22%), (i) SSFTT (85.33%), (j) GAHT (83.72%), (k) SSCNX (83.00%), and (l) NSR-Net (ours, 98.45%).

further demonstrating the reliability and effectiveness of the proposed method.

### C. Subdictionary Analysis

In this section, we aim to investigate the impact of the number of learnable atoms in the subdictionary on classification performance, as well as to evaluate the feature representation and discriminative ability of each subdictionary.

1) *Number of Learnable Atoms*: We set the range of learnable atoms in each class subdictionary to be 10–100. The classification performance of the four datasets under different numbers of atoms is illustrated in Fig. 10. From this figure, it is evident that when the number of learnable atoms is

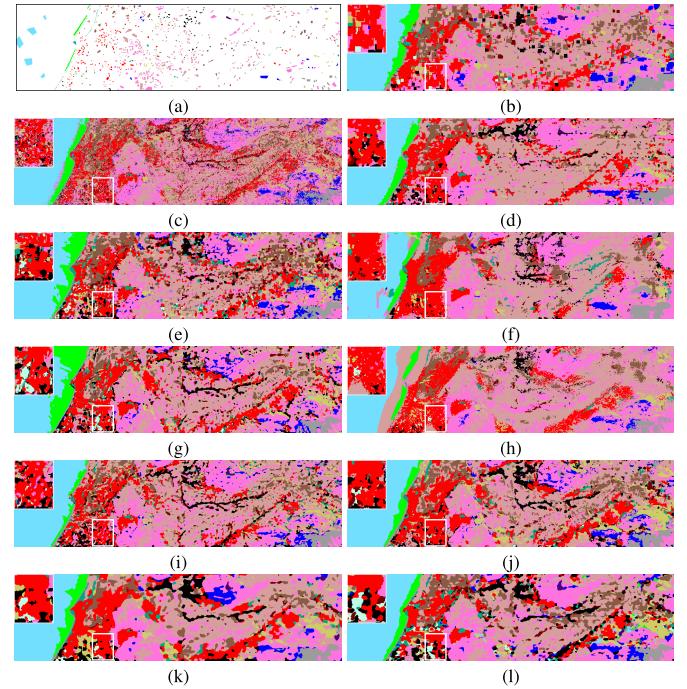


Fig. 8. Classification maps for Loukia with 5% labeled samples for training. (a) Ground truth, (b) CCJSR (80.89%), (c) LSRC (79.94%), (d) GTR-SD (80.90%), (e) TLRSR (83.81%), (f) CSSM (82.86%), (g) DFFN (80.89%), (h) RPNNet (86.22%), (i) SSFTT (85.33%), (j) GAHT (83.72%), (k) SSCNX (83.00%), and (l) NSR-Net (ours, 86.78%).

limited, the classification performance of the network cannot be effectively utilized. For example, with only ten atoms per class, the classification accuracy on all four datasets is at its lowest. The main reason is that the representation ability of the subdictionary is limited by the number of atoms, making it ineffective in capturing discriminative information from specific class data during training, thus resulting in poorer accuracy. However, more learnable atoms do not necessarily lead to better accuracy performance. From the figure, we can also observe that when the number of atoms reaches a certain threshold, the model faces risks of overfitting and decreased stability. For instance, for Salinas and LongKou datasets [as shown in Fig. 10(b) and (c)], it can be observed from the change in error bars that the standard deviation increases dramatically once the number of atoms reaches 70.

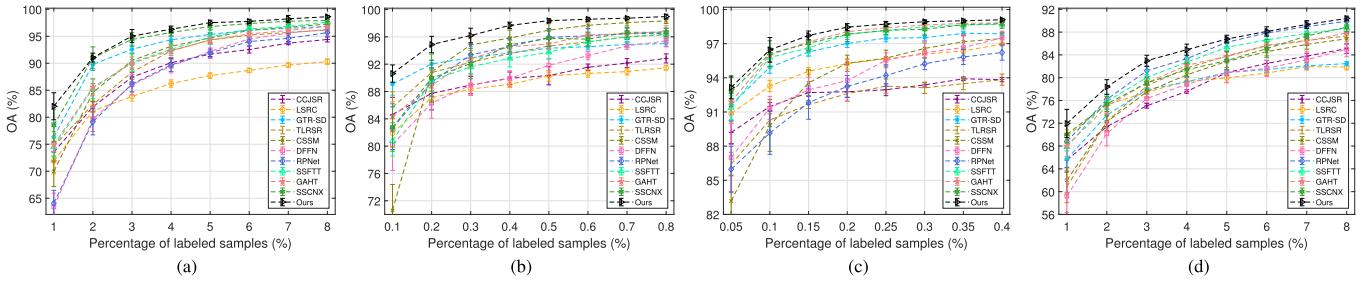


Fig. 9. Variation of OAs achieved by various methods across different percentages of training samples. (a) Indian Pines, (b) Salinas, (c) LongKou, and (d) Loukia.

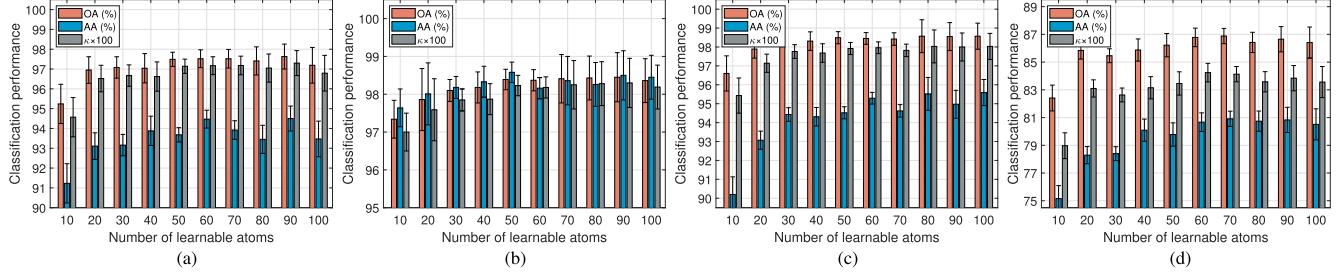


Fig. 10. Impact of the number of learnable atoms in each subdictionary on classification performance for four datasets. (a) Indian Pines, (b) Salinas, (c) LongKou, and (d) Loukia.

TABLE IX

COMPARISON OF ACCURACY AMONG DIFFERENT VARIANTS  
OF NSR-NET FOR FOUR DATASETS

Variants	Metrics	Indian Pines	Salinas	LongKou	Loukia
V1	OA (%)	96.36±0.80	96.98±0.52	97.22±0.79	85.30±0.92
	AA (%)	91.02±2.43	97.40±0.72	91.24±2.92	79.53±1.39
	$\kappa \times 100$	95.84±0.91	96.64±0.58	96.35±1.04	82.54±1.08
V2	OA (%)	89.38±1.86	89.49±5.15	88.96±6.94	81.20±1.72
	AA (%)	78.35±5.72	86.35±4.48	77.07±3.94	73.88±3.11
	$\kappa \times 100$	87.89±2.11	88.34±5.64	85.93±8.25	77.66±2.02
V3	OA (%)	93.12±1.26	95.59±0.84	95.39±0.81	83.09±0.78
	AA (%)	85.88±3.63	94.78±1.66	85.19±3.49	76.08±2.22
	$\kappa \times 100$	92.16±1.43	95.09±0.93	93.95±1.06	79.88±0.95
V4	OA (%)	96.29±0.65	95.00±0.90	96.55±0.47	81.99±0.54
	AA (%)	93.41±2.00	93.56±1.64	89.44±1.77	77.74±1.39
	$\kappa \times 100$	95.77±0.74	94.43±1.00	95.44±0.62	78.56±0.65
V5	OA (%)	94.81±1.01	95.14±1.23	96.54±0.74	83.21±1.13
	AA (%)	89.04±2.59	94.02±1.65	89.36±3.41	75.97±2.83
	$\kappa \times 100$	94.08±1.15	94.59±1.37	95.44±0.98	80.01±1.38
Ours	OA (%)	<b>97.52±0.45</b>	<b>98.37±0.28</b>	<b>98.45±0.31</b>	<b>86.78±0.67</b>
Ours	AA (%)	<b>94.47±1.80</b>	<b>98.16±0.35</b>	<b>95.29±1.08</b>	<b>80.67±1.56</b>
Ours	$\kappa \times 100$	<b>97.17±0.52</b>	<b>98.18±0.32</b>	<b>97.96±0.40</b>	<b>84.24±0.80</b>

<sup>1</sup> Variant-1 (V1): Removing the constraint loss in the loss function.

<sup>2</sup> Variant-2 (V2): Removing the nonlinear transformation part  $T(\cdot)$ .

<sup>3</sup> Variant-3 (V3): Removing the nonlinear inverse transformation part  $\tilde{T}(\cdot)$ .

<sup>4</sup> Variant-4 (V4): Removing parts of  $T(\cdot)$  and  $\tilde{T}(\cdot)$  simultaneously.

<sup>5</sup> Variant-5 (V5): Removing the SDRM module.

2) *Discriminative and Representation Ability*: As illustrated in Fig. 11, we utilize the Salinas dataset as a case study to examine the distinctions among various subdictionaries and the linear representation ability of atoms within each subdictionary. Specifically, Fig. 11(b) displays the first principal component features reconstructed based on each subdictionary. We observe that the edge contour information of land objects in the image is effectively preserved, but there are significant differences in the reconstructed features of the same land object across different subdictionaries. These feature differences caused by the diversity of subdictionaries serve as discriminative information in NSR-Net to achieve precise

TABLE X

COMPLEXITY ANALYSIS RESULTS OF DIFFERENT CLASSIFICATION METHODS ON THE SALINAS DATASET

Method	Training	Test	Parameters	FLOPs
DFFN (91.84%)	29.34 (s)	10.59 (s)	423.36 (k)	34.31 (m)
RPNet (95.92%)	31.17 (s)	20.31 (s)	358.54 (k)	37.67 (m)
SSFTT (93.74%)	22.89 (s)	9.66 (s)	950.28 (k)	25.82 (m)
GAHT (94.96%)	31.92 (s)	26.92 (s)	972.62 (k)	78.69 (m)
SSCNX (95.79%)	51.85 (s)	36.30 (s)	744.07 (k)	134.39 (m)
Ours (98.32%)	33.94 (s)	22.57 (s)	494.92 (k)	59.55 (m)

classification. Fig. 11(c) presents the feature map crafted based on the residual value of each pixel, with colors leaning toward red indicating smaller residuals. From this, we observe that each subdictionary minimizes the reconstruction error for pixels corresponding to its respective class, demonstrating the representation ability of atoms within subdictionaries for specific class features.

#### D. Complexity Analysis

To assess the computational efficiency of different classification methods, we evaluated their performance on the Salinas dataset. Table X summarizes the key metrics, including training time, test time, model size (number of parameters), and floating-point operations (FLOPs).

For fairness, the batch size, patch size, and training epochs are uniformly set to 32, 9 × 9, and 200, respectively. In addition, 0.5% of the labeled samples are used for training, and predictions are made on the entire HSI. From Table X, we observe that while models such as SSFTT have lower computational requirements in terms of training time, test time, and FLOPs, their classification accuracy (93.74%) is slightly lower compared to other methods. Moreover, methods such as GAHT and SSCNX have high computational demands but

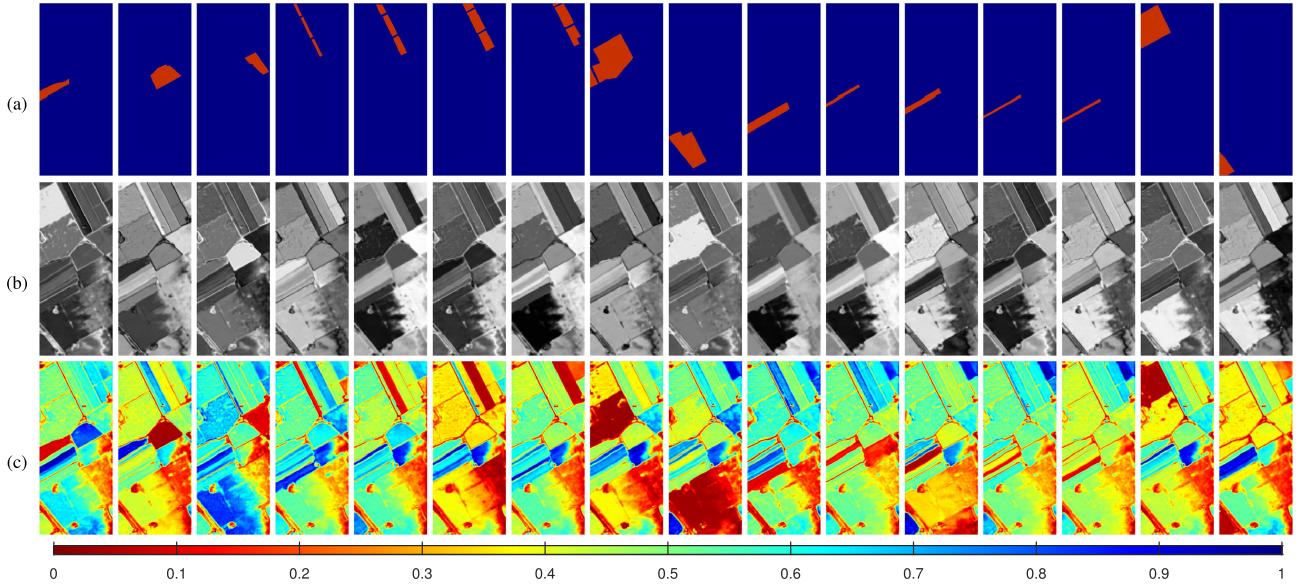


Fig. 11. Visual analysis of the subdictionary reconstruction performance for each class on the Salinas dataset. (a) Ground truth, (b) first principal component of the reconstructed data for each class, and (c) feature maps based on pixelwise residual values, where a deeper red indicates smaller residuals, reflecting more accurate reconstruction.

offer competitive classification accuracy (94.96% and 95.79%, respectively). Our method achieves the highest classification accuracy (98.32%) while maintaining a balanced complexity in terms of parameters and FLOPs. This demonstrates the effectiveness of the model in improving accuracy without significantly increasing computational burden, making it a viable solution for HSI classification tasks that demand high precision and efficiency.

#### E. Disjoint Dataset Testing

To further validate the effectiveness of the proposed method, we conducted additional classification tests on the disjoint Salinas dataset. The partitioning of the disjoint training samples and test samples is illustrated in Fig. 12, and the classification results for both the proposed and comparison methods are detailed in Table XI. Compared with other methods, our proposed method achieved improvements of 3.48%–5.93%, 4.1%–6.78%, and 3.94%–6.62% in terms of OA, AA, and  $\kappa$ , respectively. Furthermore, it demonstrated strong adaptability and robustness in class-specific accuracy, achieving the best performance in 9 out of the 16 classes. These experimental results highlight the superiority of the proposed method on the disjoint dataset, particularly in terms of overall performance and significant improvements in accuracy across the majority of classes.

#### F. Convergence Analysis

In this section, we illustrate the convergence of NSR-Net by analyzing the changes in loss with respect to the training epochs. Under the conditions of training for 200 epochs with a learning rate of  $5e-5$ , Fig. 13 illustrates the decay of loss with respect to the number of training epochs for four datasets. The evolution of the loss curves indicates a rapid decline in loss for all datasets during the initial 50 epochs of training, followed by a gradual stabilization in the subsequent training epochs. It is worth noting that, for both Indian Pines and

TABLE XI  
CLASSIFICATION ACCURACIES OF VARIOUS METHODS FOR  
THE DISJOINT DATASET OF SALINAS

Class	Comparison methods					Ours
	DFFN	RPNet	SSFTT	GAHT	SSCNX	NSR-Net
1	99.95±0.08	99.91±0.12	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>
2	98.91±0.71	94.81±1.03	<b>99.47±0.74</b>	90.30±1.77	95.86±0.00	96.45±2.61
3	54.07±1.23	62.96±3.08	57.29±2.38	61.56±3.91	62.31±2.80	<b>99.97±0.10</b>
4	94.57±3.08	<b>98.07±0.35</b>	95.31±2.27	95.73±2.43	96.80±0.79	90.97±1.73
5	88.67±5.21	96.02±2.31	94.09±2.91	90.32±1.51	92.33±1.62	<b>96.46±1.75</b>
6	99.52±0.37	<b>99.99±0.01</b>	99.80±0.31	99.99±0.02	99.59±0.41	99.89±0.18
7	92.88±1.58	98.41±1.00	98.83±0.67	<b>99.75±0.31</b>	98.51±0.38	99.65±0.38
8	89.31±0.61	83.37±1.01	86.94±4.02	82.34±8.37	87.20±2.63	<b>90.18±2.37</b>
9	98.75±1.05	98.55±0.65	97.74±0.99	96.82±1.21	97.14±1.20	<b>99.99±0.02</b>
10	64.06±2.36	68.10±2.91	70.98±2.88	75.51±2.37	72.64±3.08	<b>80.74±4.69</b>
11	92.11±2.22	90.70±2.78	98.67±0.61	99.43±0.50	98.57±0.92	<b>99.96±0.08</b>
12	96.60±5.17	<b>98.66±0.94</b>	98.11±1.27	89.44±4.99	97.82±1.59	91.49±5.02
13	97.56±1.68	98.47±1.75	99.32±0.64	<b>99.59±0.27</b>	99.03±0.85	96.80±3.31
14	97.02±0.57	99.07±0.39	99.50±0.43	<b>100.00±0.00</b>	99.04±0.05	98.59±2.63
15	81.43±1.57	80.73±0.96	87.65±3.98	91.09±2.76	91.31±0.72	<b>92.52±2.29</b>
16	78.49±0.67	76.97±0.94	80.86±2.73	84.82±1.92	78.58±0.64	<b>98.72±0.86</b>
OA (%)	89.04±0.28	88.74±0.29	90.74±0.54	89.68±1.61	91.19±0.57	<b>94.67±0.35</b>
AA (%)	88.99±0.42	90.30±0.33	91.53±0.34	91.04±0.69	91.67±0.11	<b>95.77±0.32</b>
$\kappa \times 100$	87.76±0.32	87.45±0.32	89.68±0.60	88.54±1.77	90.19±0.63	<b>94.07±0.39</b>

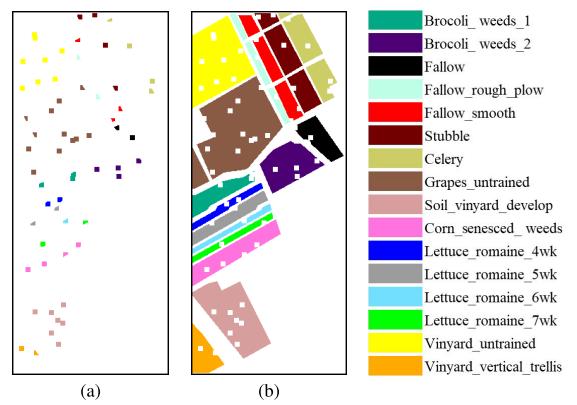


Fig. 12. Disjoint dataset of Salinas. (a) Disjoint training samples. (b) Disjoint testing samples.

Louka, although there is a slight increase in loss around epochs 85 and 125, it does not affect the overall convergence

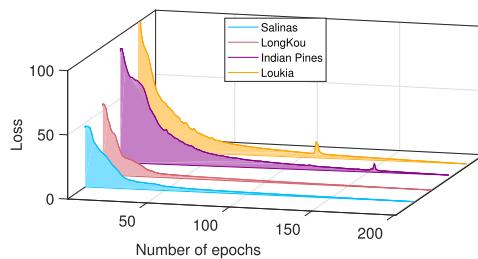


Fig. 13. Loss as a function of number of epochs in NSR-Net.

trend of the model. In conclusion, this experimental analysis demonstrates that the loss function we designed, along with the corresponding training settings, ensures the convergence of NSR-Net on different datasets.

#### G. Application and Limitation Analysis

1) *Application Analysis:* The proposed NSR-Net model, with its interpretable deep unfolding architecture and enhanced classification performance, has significant applications in remote sensing tasks, particularly HSI classification. Its ability to accurately classify HSI data in complex scenarios makes it well-suited for land cover mapping, environmental monitoring, and precision agriculture. Moreover, the interpretability of the model and reduced reliance on manual tuning of parameters increase its applicability in areas such as urban planning and disaster response, where reliable and transparent decision-making is crucial.

2) *Limitation Analysis:* While NSR-Net demonstrates strong performance in HSI classification and offers a balance between accuracy and interpretability, it does have certain limitations. On the one hand, while our experimental results demonstrate that NSR-Net exhibits a degree of robustness to parameter changes, optimal performance still often requires parameter fine-tuning. This necessity may present challenges in real-time or large-scale applications, where the capacity for repeated adjustments is limited. On the other hand, although the subdictionary reconstruction mechanism enhances classification accuracy, it may introduce scalability constraints, particularly when applied to datasets with a large number of classes. As the number of classes increases, the computational complexity associated with subdictionary learning and reconstruction also rises, which could negatively affect the efficiency of the model.

## VI. CONCLUSION

In this article, we introduce a novel deep unfolding network NSR-Net for HSI classification. Leveraging the powerful nonlinear learning ability of deep learning, NSR-Net replaces the linear constraint in the prior-constrained NSR model with a deep constraint that offers superior generalization performance. The architecture of the network is designed based on the solving process of the deep-constrained NSR model, providing clear interpretability. Moreover, the proposed SDRM learns discriminative subdictionaries through a data-driven approach, allowing it to produce accurate decision outcomes. The optimization of learnable parameters is guided by a composite loss function, enabling effective recovery of

representation coefficients and data reconstruction. Classification results on four different HSI datasets show that our method can achieve better accuracy than related representation model-based and deep learning-based methods, and a series of experimental analyses validated the superiority and effectiveness of the proposed method.

As immediate next steps, we aim to incorporate the LRR model into the network structure to capture global structural characteristics of HSI data more effectively. Furthermore, we plan to apply NSR-Net to larger and more complex real-world hyperspectral datasets to assess its scalability and explore its potential in practical applications.

## ACKNOWLEDGMENT

The authors would like to thank Prof. D. Landgrebe for making the Airborne Visible/Infrared Imaging Spectrometer Indian Pines hyperspectral dataset available to the community and Prof. Y. F. Zhong for providing the WHU-LongKou hyperspectral dataset. They would also like to thank Dr. Y. Zhang for sharing the code of locality-constrained sparse representation (LCSR), Prof. J. J. Liu for generalized tensor regression based on spatially derived features (GTR-SD), Dr. Y. H. Xu for random patches network (RPNet), Prof. L. Sun for spectral-spatial feature tokenization Transformer (SSFTT), Prof. S. H. Mei for group-aware hierarchical Transformer (GAHT), and Dr. Y. M. Zhu for spatial-spectral ConvNeXt (SSCNX), respectively.

## REFERENCES

- [1] B. Zhang et al., "Progress and challenges in intelligent remote sensing satellite systems," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1814–1822, 2022.
- [2] P. Ghamisi et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.
- [3] M. A. Moharram and D. M. Sundaram, "Land use and land cover classification with hyperspectral data: A comprehensive review of methods, challenges and future directions," *Neurocomputing*, vol. 536, pp. 90–113, Jun. 2023.
- [4] D. Hong et al., "Interpretable hyperspectral artificial intelligence: When nonconvex modeling meets hyperspectral remote sensing," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 2, pp. 52–87, Jun. 2021.
- [5] L. Zhang and L. Zhang, "Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 2, pp. 270–294, Jun. 2022.
- [6] M. Ahmad et al., "Hyperspectral image classification—Traditional to deep models: A survey for future prospects," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 968–999, 2022.
- [7] M. Wang et al., "Tensor decompositions for hyperspectral data processing in remote sensing: A comprehensive review," *IEEE Geosci. Remote Sens. Mag.*, vol. 11, no. 1, pp. 26–72, Mar. 2023.
- [8] W. Liu, X. Shen, B. Du, I. W. Tsang, W. Zhang, and X. Lin, "Hyperspectral imagery classification via stochastic HHSVMs," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 577–588, Feb. 2019.
- [9] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
- [10] J. Xia, P. Ghamisi, N. Yokoya, and A. Iwasaki, "Random forest ensembles and extended multietinction profiles for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 202–216, Jan. 2018.
- [11] J. Peng et al., "Low-rank and sparse representation for hyperspectral image processing: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 10–43, Mar. 2022.
- [12] H. Su, F. Shao, Y. Gao, H. Zhang, W. Sun, and Q. Du, "Probabilistic collaborative representation based ensemble learning for classification of wetland hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5509517.

- [13] S. Yang, J. Hou, Y. Jia, S. Mei, and Q. Du, "Superpixel-guided discriminative low-rank representation of hyperspectral images for classification," *IEEE Trans. Image Process.*, vol. 30, pp. 8823–8835, 2021.
- [14] J. Peng, L. Li, and Y. Y. Tang, "Maximum likelihood estimation-based joint sparse representation for the classification of hyperspectral remote sensing images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1790–1802, Jun. 2019.
- [15] B. Du, Y. Zhang, L. Zhang, and D. Tao, "Beyond the sparsity-based target detector: A hybrid sparsity and statistics-based detector for hyperspectral images," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5345–5357, Nov. 2016.
- [16] Y. Shao, N. Sang, C. Gao, and L. Ma, "Spatial and class structure regularized sparse representation graph for semi-supervised hyperspectral image classification," *Pattern Recognit.*, vol. 81, pp. 81–94, Sep. 2018.
- [17] Y. Zhang, Y. Ma, X. Dai, H. Li, X. Mei, and J. Ma, "Locality-constrained sparse representation for hyperspectral image classification," *Inf. Sci.*, vol. 546, pp. 858–870, Feb. 2021.
- [18] T. Liu, Y. Gu, X. Jia, J. A. Benediktsson, and J. Chanussot, "Class-specific sparse multiple kernel learning for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7351–7365, Dec. 2016.
- [19] T. Liu, Y. Gu, and X. Jia, "Class-guided coupled dictionary learning for multispectral-hyperspectral remote sensing image collaborative classification," *Sci. China Technol. Sci.*, vol. 65, no. 4, pp. 744–758, Apr. 2022.
- [20] Z. Xue, X. Nie, and M. Zhang, "Incremental dictionary learning-driven tensor low-rank and sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5544019.
- [21] X. Nie, Z. Xue, C. Lin, L. Zhang, and H. Su, "Structure-prior-constrained low-rank and sparse representation with discriminative incremental dictionary for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5506319.
- [22] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.
- [23] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019.
- [24] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, Aug. 2021.
- [25] L. Chen, J. Wu, Y. Xie, E. Chen, and X. Zhang, "Discriminative feature constraints via supervised contrastive learning for few-shot forest tree species classification using airborne hyperspectral images," *Remote Sens. Environ.*, vol. 295, Sep. 2023, Art. no. 113710.
- [26] Y. Su et al., "DAAN: A deep autoencoder-based augmented network for blind multilinear hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5512715.
- [27] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.
- [28] Y. Zhu, K. Yuan, W. Zhong, and L. Xu, "Spatial–spectral ConvNeXt for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5453–5463, 2023.
- [29] J. Zeng, Z. Xue, L. Zhang, Q. Lan, and M. Zhang, "Multistage relation network with dual-metric for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5510017.
- [30] L. Liang, S. Zhang, and J. Li, "Multiscale DenseNet meets with bi-RNN for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 5401–5415, 2022.
- [31] W. Zhou, S.-I. Kamata, Z. Luo, and H. Wang, "Multiscanning strategy-based recurrent neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5521018.
- [32] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 1559–1572, 2022.
- [33] Z. Xue, Z. Liu, and M. Zhang, "DSR-GCN: Differentiated-scale restricted graph convolutional network for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5504918.
- [34] Y. Lu, S. Mei, F. Xu, M. Ma, and X. Wang, "Separable deep graph convolutional network integrated with CNN and prototype learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5516216.
- [35] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522214.
- [36] S. Mei, C. Song, M. Ma, and F. Xu, "Hyperspectral image classification using group-aware hierarchical transformer," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5539014.
- [37] Z. Li, Z. Xue, Q. Xu, L. Zhang, T. Zhu, and M. Zhang, "SPFormer: Self-pooling transformer for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5502019.
- [38] J. Wang, M. Zhang, W. Li, and R. Tao, "A multistage information complementary fusion network based on flexible-mixup for HSI-X image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 17189–17201, Dec. 2024.
- [39] Z. Lv, P. Zhang, W. Sun, J. A. Benediktsson, and T. Lei, "Novel land-cover classification approach with nonparametric sample augmentation for hyperspectral remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4407613.
- [40] Y. Gao, W. Li, J. Wang, M. Zhang, and R. Tao, "Relationship learning from multisource images via spatial–spectral perception network," *IEEE Trans. Image Process.*, vol. 33, pp. 3271–3284, 2024.
- [41] J. Wang, W. Li, Y. Gao, M. Zhang, R. Tao, and Q. Du, "Hyperspectral and SAR image classification via multiscale interactive fusion network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10823–10837, Dec. 2023.
- [42] J. Wang, W. Li, Y. Wang, R. Tao, and Q. Du, "Representation-enhanced status replay network for multisource remote-sensing image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 15346–15358, Nov. 2024.
- [43] Z. Lv, M. Zhang, W. Sun, J. A. Benediktsson, T. Lei, and N. Falco, "Spatial-contextual information utilization framework for land cover change detection with hyperspectral remote sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4411911.
- [44] Z. Lv, J. Liu, W. Sun, T. Lei, J. A. Benediktsson, and X. Jia, "Hierarchical attention feature fusion-based network for land cover change detection with homogeneous and heterogeneous remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4411115.
- [45] B. Zhao et al., "Intermediate domain prototype contrastive adaptation for *Spartina alterniflora* segmentation using multitemporal remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5401314.
- [46] F. Ullah, I. Ullah, R. U. Khan, S. Khan, K. Khan, and G. Pau, "Conventional to deep ensemble methods for hyperspectral image classification: A comprehensive survey," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 3878–3916, 2024.
- [47] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.
- [48] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *Proc. IEEE*, vol. 111, no. 5, pp. 465–499, May 2023.
- [49] P. Wang, Z. He, B. Huang, M. D. Mura, H. Leung, and J. Chanussot, "VOGTNet: Variational optimization-guided two-stage network for multispectral and panchromatic image fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 17, 2024, doi: 10.1109/TNNLS.2024.3409563.
- [50] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Jun. 2010, pp. 399–406.
- [51] H. Li, T. Xu, X.-J. Wu, J. Lu, and J. Kittler, "LRRNet: A novel representation learning guided fusion network for infrared and visible images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11040–11052, Sep. 2023.
- [52] J. Qu, W. Dong, Y. Li, S. Hou, and Q. Du, "An interpretable unsupervised unrolling network for hyperspectral pansharpening," *IEEE Trans. Cybern.*, vol. 53, no. 12, pp. 7943–7956, Dec. 2023.
- [53] F. Xiong, J. Zhou, J. Zhou, J. Lu, and Y. Qian, "Multitask sparse representation model-inspired network for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5518515.
- [54] C. Li, B. Zhang, D. Hong, J. Yao, and J. Chanussot, "LRRNet: An interpretable deep unfolding network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5513412.
- [55] Q. Xie, M. Zhou, Q. Zhao, Z. Xu, and D. Meng, "MHF-Net: An interpretable deep network for multispectral and hyperspectral image fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1457–1473, Mar. 2022.

- [56] K. Wang, X. Liao, J. Li, D. Meng, and Y. Wang, "Hyperspectral image super-resolution via knowledge-driven deep unrolling and transformer embedded convolutional recurrent neural network," *IEEE Trans. Image Process.*, vol. 32, pp. 4581–4594, 2023.
- [57] J. Qu, J. Zhao, W. Dong, S. Xiao, Y. Li, and Q. Du, "Feature mutual representation-based graph domain adaptive network for unsupervised hyperspectral change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5501214.
- [58] W. Dong, T. Yang, J. Qu, T. Zhang, S. Xiao, and Y. Li, "Joint contextual representation model-informed interpretable network with dictionary aligning for hyperspectral and LiDAR classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 11, pp. 6804–6818, Nov. 2023.
- [59] W. Li, F. Feng, H. Li, and Q. Du, "Discriminant analysis-based dimension reduction for hyperspectral image classification: A survey of the most recent advances and an experimental comparison of different techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 1, pp. 15–34, Mar. 2018.
- [60] J. Xu, W. An, L. Zhang, and D. Zhang, "Sparse, collaborative, or nonnegative representation: Which helps pattern classification?" *Pattern Recognit.*, vol. 88, pp. 679–688, Apr. 2019.
- [61] L. Shi, C. Li, T. Li, and Y. Peng, "A complementary spectral-spatial method for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5531017.
- [62] D. Wang, J. Zhang, B. Du, L. Zhang, and D. Tao, "DCN-T: Dual context network with transformer for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 32, pp. 2536–2551, 2023.
- [63] J. Zhao, J. Wang, C. Ruan, Y. Dong, and L. Huang, "Dual-branch spectral-spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5504718.
- [64] M. Kowalski, "Thresholding RULES and iterative shrinkage/thresholding algorithm: A convergence study," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 4151–4155.
- [65] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 1828–1837.
- [66] B. Tu, X. Zhang, X. Kang, G. Zhang, J. Wang, and J. Wu, "Hyperspectral image classification via fusing correlation coefficient and joint sparse representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 340–344, Mar. 2018.
- [67] J. Liu, Z. Wu, L. Xiao, J. Sun, and H. Yan, "Generalized tensor regression for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1244–1258, Feb. 2020.
- [68] Y. Xu, B. Du, F. Zhang, and L. Zhang, "Hyperspectral image classification via a random patches network," *ISPRS J. Photogramm. Remote Sens.*, vol. 142, pp. 344–357, Aug. 2018.



**Xiangyu Nie** received the B.S. degree in geomatics engineering from Suzhou University of Science and Technology, Suzhou, China, in 2019, and the M.E. degree in remote sensing from Hohai University, Nanjing, China, in 2022, where he is currently pursuing the Ph.D. degree in surveying and mapping.

His research interests include low-rank and sparse representation, and deep unfolding for hyperspectral image classification.



**Zhaohui Xue** (Member, IEEE) received the B.S. degree in geomatics engineering from Shandong Agricultural University, Tai'an, China, in 2009, the M.E. degree in remote sensing from China University of Mining and Technology, Beijing, China, in 2012, and the Ph.D. degree in cartography and geographic information system from Nanjing University, Nanjing, China, in 2015.

He is currently a Full Professor (Ph.D. Supervisor) with the College of Geography and Remote Sensing, Hohai University, Nanjing. He has authored more than 70 scientific articles including more than 40 science citation index (SCI) articles. His research interests include hyperspectral image classification, time-series image analysis, pattern recognition, and machine learning.

Dr. Xue was a recipient of the National Scholarship for Doctoral Graduate Students granted by the Ministry of Education of China in 2014. He was awarded the Best Reviewer for the IEEE GEOSCIENCE AND REMOTE SENSING SOCIETY. He was an Editorial Board Member in *National Remote Sensing Bulletin* from 2020 to 2024. He has been a reviewer for more than ten famous remote sensing journals including *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING* and *ISPRS Journal of Photogrammetry and Remote Sensing*.



**Hongjun Su** (Senior Member, IEEE) received the Ph.D. degree in cartography and geography information system from the Key Laboratory of Virtual Geographic Environment (Ministry of Education), Nanjing Normal University, Nanjing, China, in 2011.

He is currently a Full Professor with the College of Geography and Remote Sensing, Hohai University, Nanjing. His main research interests include hyperspectral remote sensing dimensionality reduction, classification, and spectral unmixing.

Dr. Su received the 2016 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society. He is an Associate Editor of the *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*.



**Jun Li** (Fellow, IEEE) received the B.S. degree in geographic information systems from Hunan Normal University, Changsha, China, in 2004, the M.E. degree in remote sensing from Peking University, Beijing, China, in 2007, and the Ph.D. degree in electrical engineering from the Instituto de Telecomunicações, Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Lisbon, Portugal, in 2011.

From 2013 to 2021, she was a Full Professor with Sun Yat-sen University, Guangzhou, China. Since 2022, she has been with China University of Geosciences, Wuhan, China, as a Full Professor. She has received several prestigious funding grants at the national and international levels. She has authored more than 160 journal citation report (JCR) articles, 60 international conference papers, and a book chapter.

Dr. Li has been serving as an Editor-in-Chief for the *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*.