An Optimization-Driven Network With Knowledge Prior Injection for HSI Denoising

Yajie Li, Jie Li, *Member, IEEE*, Jiang He, *Graduate Student Member, IEEE*, Xinxin Liu, *Member, IEEE*, and Qiangqiang Yuan, *Member, IEEE*

***Abstract*— Due to the limitations of sensor hardware devices, the hyperspectral image (HSI) often suffers from various types of noise, such as Gaussian noise, impulse noise, stripe noise, and deadlines, which can significantly degrade their quality. Although many data-driven methods have been proposed to deal with complex noise, few of them consider the structural characteristics of noise. This not only leads to a lack of interpretability but also results in poor performance when dealing with structural noise in practical applications. To address this issue, this article proposes KPInet, a convolutional neural network (CNN) driven by the structural knowledge of noise for HSI denoising. First and foremost, the knowledge optimization-driven module (KODM) utilizes the deep unrolling method to unfold a total variation (TV) algorithm that considers the structural characteristics of noise. This approach improves the network's interpretability and results in better performance on structural noise, while maintaining the effect of removing Gaussian noise. Second, the statistical feature injection module (SFIM) extracts more features by utilizing spectral gradients, medians, and means of the HSI. Third, the multiscale degradation guidance module (MDGM) utilizes a dual-stream decoder with a low-resolution upsampling guidance branch to better distinguish the real structure and noise structure in the HSI. Experimental results on simulated and real datasets indicate that the approach achieves favorable denoising performance, as evidenced by both quantitative evaluation metrics and visual results. Furthermore, it also demonstrates the robustness and generalization capacity of the proposed KPInet.**

***Index Terms*— Convolutional neural network (CNN), deep unrolling, hyperspectral image (HSI) denoising, mixed noise, total variation (TV) model.**

I. Introduction

HYPERSPECTRAL image (HSI), due to its rich spectral information, has been widely applied in dealing with various tasks, such as classification [1], pan-sharpening [2], [3], object detection and tracking [4], and damage detection [5]. However, the imaging energy of the spectrometer is attenuated, which inevitably corrupts the observed HSIs with complex noise such as Gaussian noise, impulse noise, stripes, and deadlines. These noise significantly degrade the visual quality of HSIs and restrict their further applications [6], [7].

In the beginning, it is crucial to comprehend the distinct characteristics and sources of noise in order to develop effective denoising techniques. Gaussian noise, arising from random variations in the intensity of light, conforms to a normal distribution. Impulse noise, caused by errors in the imaging sensor or transmission system, manifests as isolated pixels with extremely high- or low-intensity values. Stripes, caused by the nonuniformity of the sensors or errors in the calibration process, appear as a series of bright or dark lines running horizontally or vertically across the image. Finally, deadlines represent regions where no data are captured and are typically observed as black lines in the image. Among these, stripes and deadlines are more difficult to remove as they are highly structured and often intertwined with the underlying image structure. In comparison to Gaussian and impulse noise, these two forms of structural noise exhibit intricate spatial distributions that cannot be easily modeled by simple statistical models. Hence, there is an urgent need for an efficient denoising method capable of effectively suppressing Gaussian and impulse noise, while also adeptly handling structural noise.

In the past few decades, various methods have been proposed to address the problem of the HSI denoising. These methods can be broadly categorized into two types: model-driven and data-driven methods. Model-driven methods, which are based on mathematical models, often have a clear physical interpretation, allowing for better understanding of the denoising process. Examples of such methods include filter [8], [9], [10], [11], [12], [13], sparse representation [14], [15], low rank (LR) [16], [17], [18], [19], [20], and total variation (TV) regularization [21], [22], [23], [24]. This category of methods for HSI denoising has several advantages. They are often based on physical models that can capture the basic process of the degradation and the characteristics of noise, which ensures stronger generalization across different sensors. Most importantly, these models perform well at removing structural noise by considering the physical characteristics of the noise, as demonstrated by some destriping methods [21], [25]. Although the great interpretability and generalization ability of model-driven methods make them attractive options for many denoising tasks in practical applications,

Manuscript received 1 June 2023; revised 15 August 2023 and 27 September 2023; accepted 27 October 2023. Date of current version 29 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62071341 and in part by the Key Research and Development Program of Hubei Province under Grant 2023BAB066. *(Corresponding authors: Jie Li; Jiang He.)*

Yajie Li and Jiang He are with the School of Geodesy and Geomatics, Wuhan University, Wuhan 430072, China (e-mail: l_anyways@163.com; hej96.work@gmail.com).

Jie Li and Qiangqiang Yuan are with the School of Geodesy and Geomatics and the Hubei Luojia Laboratory, Wuhan University, Wuhan 430072, China (e-mail: jli89@sgg.whu.edu.cn; yqiang86@gmail.com).

Xinxin Liu is with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China (e-mail: liuxinxin@hnu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2023.3329887

these advantages come with a cost. The inclusion of complex mathematical models results in numerous hyperparameters that require manual adjustment for different scenarios, making their usage inconvenient. Finally, model-driven approaches struggle to handle complex noise due to the inability of existing physical models to accurately represent it.

With the increasing availability of large datasets, the advancements in machine learning algorithms have led to the rapid development of data-driven methods for denoising. These methods have gained popularity in recent years due to their superior performance and faster speed. Data-driven methods primarily include deep-learning-based algorithms, such as convolutional neural network (CNN)-based methods [26], [27], [28], [29], attention-based methods [30], [31], [32], and recurrent network-based methods [33], [34]. Data-driven methods achieve several advantages over model-driven methods. First of all, compared with model-driven methods, they have superior nonlinear fitting ability and can learn complex nonlinear input–output relationship from a large number of the HSI data. This advantage allows them to capture the underlying features of complex noise through powerful learning abilities, enabling the construction of effective denoising models. Second, with a sufficient number of training samples, data-driven methods can achieve state-of-the-art denoising performance through an end-to-end approach. This also means that these methods do not require manual adjustment of additional hyperparameters during usage, delivering excellent results and extremely fast processing speeds. However, data-driven methods rely on black-box neural networks with numerous uncertain parameters, resulting in highly complex models that are challenging to interpret. When processing HSI data, they do not specifically consider the prior knowledge of noise. As a result, they lack the ability to remove structural noise.

As model-driven and data-driven methods have their respective strengths and weaknesses, we aim to combine their advantages and better overcome the limitations. Data-driven methods offer powerful modeling capabilities, enabling efficient removal of mixed noise. On the other hand, model-driven methods can provide valuable physical priors that assist data-driven approaches in effectively addressing structural noise, especially wide ones. Therefore, we propose the utilization of a model-driven algorithm that incorporates structural noise features to guide the construction of the deep learning network. By doing so, each step of the end-to-end network acquires the physical meaning of learning structural priors, facilitating the integration of model-driven and data-driven approaches.

Based on the above discussion, this article proposes an optimization-driven network with knowledge prior injection (KPInet) for HSI denoising. The network consists of three modules, including the knowledge optimization-driven module (KODM), the statistical feature injection module (SFIM), and the multiscale degradation guidance module (MDGM). The main contributions of the proposed approach can be summarized as follows.

1) To enhance the interpretability of the network and improve its performance in removing structural noise, we designed a module driven by knowledge optimization called the KODM. The KODM incorporates the characteristics of the noise structure, considering factors such as the smoothness of the image structure in both spatial and spectral directions, as well as the discontinuities caused by structural noise.
2) The SFIM is introduced to extract statistical features from the HSI, including spectral gradients, medians, and means. These prior knowledge-driven features aid the network in capturing relevant information more effectively.
3) The MDGM uses a dual-stream decoder to restore the HSI. It is guided by information from low-resolution images during the upsampling process. This architecture enhances the differentiation between real structure and noise structure, resulting in more effective noise removal.
4) We conduct extensive experiments to validate the effectiveness of our proposed method. We compare its performance with several methods, using both quantitative and qualitative evaluations. The results clearly demonstrate the superior performance of our approach.

The rest of this article is organized as follows. Section II provides a review of the related work on HSI denoising. In Section III, we present our model formulation and network details. In Section IV, we demonstrate the effectiveness of KPInet through both simulated and real-world experiments. Finally, we summarize the article and draw conclusions based on the experimental results in Section V.

II. Related Works

In this section, we review HSI denoising methods in two categories: model-driven and data-driven.

A. Model-Driven Methods

Model-driven methods are based on optimization models or mathematical theories. Earlier algorithms for 2-D grayscale images were used band by band. For instance, Buades et al. [10] proposed the nonlocal means (NLM) filter, Dabov et al. [8] proposed the block-matching and 3-D (BM3D) filter, Gu et al. [35] proposed the weighted nuclear norm minimization (WNNM), and Donoho [12] proposed the wavelet transform. However, these denoising methods often result in greater spectral distortion because they ignore the correlation of spectral information. To address this limitation, many authors tried to propose various spatial–spectral methods for HSI. Heckel and Hand [9] proposed the 4-D (BM4D) filter utilizing 4-D data by extending BM3D filter. Likewise, Qian and Ye [11] proposed 3-D NLM, and Othman and Qian [13] proposed 3-D wavelet, both based on previous methods. These methods take into consideration adjacent bands, which are particularly valuable for HSI recovery.

Then, to make full use of the prior information of the HSI, TV and LR regularization methods are commonly employed. TV regularization primarily smooths the image by minimizing its gradient while preserving the edge information to the greatest extent possible. Its advantage lies in its effectiveness

in removing structural noise and preserving edge information. For example, the spectral–spatial adaptive hyperspectral TV (SSAHTV) proposed by Yuan et al. [23] considered both spectral and spatial differences, the hybrid spatial–spectral TV (HSSTV) proposed by Takeyama et al. [22] could better remove artifacts, the anisotropic spectral-spatial TV (ASSTV) proposed by Chang et al. [21] achieved great results in the destriping by constraining the smoothness of image structure and the discontinuity caused by noise. However, the TV-based model often loses the detail information of the image, causing the image to look fuzzy. The LR method mainly uses the LR property of the image matrix to remove noise and can maintain the structural information and details of the image. There are three important and common tensor decompositions for characterizing the LR features of the HSI: T-SVD [36], TUCKER3 [37], and CANDECOMP (CP) [38]. Zhang et al. [17] proposed an LR matrix recovery (LRMR) model for the mixed noise removal in the HSI. And He et al. [20] proposed the nonlocal meets global (NGmeet) model by exploiting the similarities among full-band patches. Additionally, some methods combine LR and TV with promising results. On the one hand, the LR-based methods can be used to efficiently separate the LR clean image and the sparse noise. On the other hand, the TV-based method can be adopted to effectively remove the Gaussian noise [19]. The TV regularized LR matrix factorization (LRTV) model proposed by He et al. [19] introduced TV regularization into the LR matrix decomposition model. The TV regularized LR tensor decomposition (LRTDTV) proposed by Chen et al. [18] introduced ASSTV to consider both spatial and spectral smoothness.

In recent years, many fast denoising methods based on subspace representation have been proposed, which could typically leverage the LR and sparse properties of the HSI to separate noise from HSIs. These methods are usually faster, more scalable, and more practical than model-driven methods. FastHyDe proposed by Zhuang and Bioucas-Dias [39] was an LR tensor approximation-based method that used tensor decomposition and nuclear norm regularization to remove noise while preserving the spatial and spectral structure of the HSI. FastHyMix proposed by Zhuang and Ng [40] was a fast mixed noise removal method that utilized LR tensor decomposition and matrix shrinkage techniques to remove multiple types of noise. Inspired by t-SVD, Lin et al. [41] proposed the TenSRDe, which developed a tensor space representation that can faithfully convey the intrinsic structure of the HSI tensor.

However, because of the complex mathematical structure, the parameters in model-driven methods require manual setting and adjustment. Moreover, there are a large number of priors for different properties that result in high computational complexity and significant time costs. In addition to efficiency and running speed issues, the nonlinear fitting ability of model-driven methods is limited, thus these methods can hardly handle complex noise.

B. Data-Driven Methods

Data-driven methods have gained increasing popularity for HSI denoising in recent years due to their capability to automatically learn complex nonlinear mappings from data. Numerous deep neural network architectures have been proposed for HSI denoising.

CNNs have been widely used for HSI denoising due to their ability to extract spatial features from high-dimensional data. The spatial–spectral deep residual CNN (HSID-CNN) proposed by Yuan et al. [26] is an early residual network that considers both spatial and spectral feature extraction. Zhao et al. [27] proposed the attention-based deep residual network (ADRN), which utilized convolution layers with different filter sizes to extract multiscale features and employed shortcut connections to incorporate multilevel information. With the increase in computing power, 3-D convolutions have been widely used in CNNs for HSI processing to more effectively expand the convolutional receptive field and better fit the HSI. 3-D atrous CNN (3DADCNN) proposed by Liu and Lee [28] combined a spatial–spectral deep architecture with 3-D atrous convolution to better extract multiscale features. Recurrent neural networks (RNNs) have also been used in the HSI denoising, particularly for their strong ability to model spectral correlation in the spectral domain. For example, the 3-D quasi-RNN (QRNN3D) proposed by Wei et al. [33] used a 3-D convolutional architecture to learn spatial–spectral features from the HSI and has achieved impressive results in denoising. Due to the limitations of convolutional design, the ability of CNN and RNN models to capture global context is limited. In contrast, self-attention mechanism enables the model to attend to different regions of the input, allowing it to better capture long-range dependencies and global contexts. Nonlocal self-similarity neural network (NSSNN) proposed by Fu et al. [34] integrates spatial–spectral relationship, global spectral correlation, and nonlocal spatial correlation to extract features with more precise structures. The local–global feature-aware transformer-based residual network (FATR) proposed by Wang et al. [30] utilized a spectral embedding operation and a multiscale windows partitioning scheme to extract spectral–spatial features from the HSI. Besides, Chen et al. [32] proposed a transformer with spatial–spectral constrains to achieve HSI denoising. With a lot of training data, the above methods are greatly capable of nonlinear mapping. However, they only rely on feature extraction and do not consider the specific HSI priors.

Recently, several methods have been proposed to incorporate image priors into neural networks. These methods can be categorized into four groups. First, some methods utilize the principles of physical models to construct neural networks [42]. For instance, Zhang et al. [42] proposed an LR spatial–spectral network (LR-Net), which integrated the LR matrix decomposition model into a deep CNN with 3-D atrous convolution. Second, certain methods replace a specific part of the algorithm with neural networks [43], [44], [45], [46], [47], [48]. As an example, Xiong et al. [43] proposed a model-aided nonlocal neural network (MAC-Net), which first built a spectral LR model and then integrated a nonlocal U-Net into the model. Third, some researchers have utilized different loss functions to capture the HSI priors. For example, Aetesam et al. [49] constructed a loss function by designing a discriminative learning framework with a Bayesian viewpoint.

Furthermore, HSI priors are utilized to extract mathematical features during the preprocessing stage, thereby strengthening the model's ability to capture the spectral priors. For example, Dou et al. [50] proposed a new data augmentation method (PatchMask), which combined the characteristics of the HSI for feature extraction and guided the network to focus on to the noisy region.

Overall, both of data-driven and model-driven methods have a long history of development and have achieved promising results, but they still have their own limitations. The KPInet aims to combine the two methods and leverage their complementary strengths.

III. Methodology

In this section, we will first derive the necessary formulas for the model and then present the overall framework of the KPInet, followed by a detailed description of three proposed modules: the SFIM, the MDGM, and the KODM.

A. Model Formulation

The HSI captured by remote sensing satellites is often contaminated by various types of noise, including Gaussian noise, impulse noise, deadline, and stripe noise. The noise model can be formulated as

$$Y = X + N \tag{1}$$

where $Y \in \mathbb{R}^{W\times H\times C}$ represents the noisy image, $X \in \mathbb{R}^{W\times H\times C}$ denotes the original image, and $N \in \mathbb{R}^{W\times H\times C}$ means noise, including Gaussian noise, impulse noise, deadline, and stripe noise.

Aiming to address denoising as an ill-posed inverse problem, the objective of this article is to estimate a latent clear image from a given degraded image with multiple sources of random noise. Theoretically, our task is to estimate a potential clear image X from given image Y containing multiple types of random noise. To solve this problem, we adopt an optimization model that includes data fidelity and prior terms. The data fidelity terms measure the similarity between the degraded image and the desired clear image, while the prior terms impose constraints on the image

$$\hat{X} = \underset{X}{\operatorname{argmin}} \|Y-X\|_2^2 + \lambda \mathcal{R}(X) \tag{2}$$

where $\hat{X}$ represents the estimated X, argmin_X means that this optimization problem is a minimization problem, $||\cdot||_2^2$ denotes the $L2$ norm, and Y is the noisy input. λ is a weight parameter for the prior term $\mathcal{R}(X)$, which maintains the tradeoff between the data fidelity term and the prior term. The key to effective noise reduction through an optimization model is to construct an appropriate prior term.

Due to the filter-like structure of CNNs, they are better suited for dealing with pixel-level noise such as Gaussian noise and impulse noise than structural noise like deadline and stripe noise. To better remove the structural noise, the characteristics of the structure must be taken into account. For a specific noisy HSI, all the structural noise extends in the same direction and breaks the spatial and spectral continuity of the HSI. Along the direction of the structural noise, adjacent pixels often exhibit minimal differences, indicating local oversmoothing caused by structural noise. Across the direction of the structural noise, noise and image information appear alternatively, resulting in noticeable discontinuities in the image. Similar discontinuities are also observed in the spectral direction. Recognizing these structural priors, the KPInet imposes constraints on the network in both the spectral and spatial domains. Specifically, the KPInet treats the denoising process as an optimization problem and further refines the prior term

$$\begin{aligned}\hat{X} = \underset{X}{\operatorname{argmin}} &\|Y-X\|_2^2 + \lambda_1\|\nabla_v X\|_2^2 + \lambda_2\|\nabla_z X\|_2^2 \\ &+ \lambda_3\|\nabla_h Y - \nabla_h X\|_2^2 + \mu\mathcal{J}(X)\end{aligned} \tag{3}$$

where $\nabla_v X$ and $\nabla_h X$ represent the gradients of X across the structural noise direction and along the structural noise direction. Meanwhile, $\nabla_z X$ denotes the gradient of X along the spectral dimension. The gradient across the structural noise direction of the output Y is represented by $\nabla_h Y$. There are three regularization parameters, λ_1, λ_2, and λ_3, each associated with a specific prior. The weight parameter for the regularization term $\mathcal{J}(X)$ is denoted as μ. The first term in the equation ensures that the restored image $\hat{X}$ contains the most relevant information from the observed image Y. The second term enforces smoothness in the restored image by suppressing the gradient across the direction of the structural noise. This helps to mitigate the discontinuities caused by the structural noise. The third term penalizes spectral gradients to ensure spectral continuity. Finally, the fourth term maintains the gradient along the structural noise direction by constraining the difference between the gradient of the restored image $\hat{X}$ and that of the noisy input Y. In this way, both the spectral consistency in the spectral domain and the directional information of structural noise in the spatial domain are effectively utilized.

To solve the constrained optimization problem in (3), the variable splitting technique is commonly used [51], [52]. This technique introduces an auxiliary variable to decompose the original problem into several subproblems that are easier to solve. With the help of variable splitting technique, (4) introduces an auxiliary variable H and reformulates the constrained optimization problem in (3). Formulation can be defined as

$$\begin{aligned}\hat{X} = \underset{X}{\operatorname{argmin}} &\|Y-X\|_2^2 + \lambda_1\|\nabla_v X\|_2^2 + \lambda_2\|\nabla_z X\|_2^2 \\ &+ \lambda_3\|\nabla_h Y - \nabla_h X\|_2^2 + \mu\mathcal{J}(H), \ \text{s.t. } H = X\end{aligned} \tag{4}$$

where H is equal to X.

According to the half-quadratic splitting method, the cost function is then transformed into

$$\begin{aligned}L(X, H) = &\|Y - X\|_2^2 + \lambda_1\|\nabla_v X\|_2^2 + \lambda_2\|\nabla_z X\|_2^2 \\ &+ \lambda_3\|\nabla_h Y - \nabla_h X\|_2^2 + \lambda_4\|X - H\|_2^2 + \mu\mathcal{J}(H)\end{aligned} \tag{5}$$

where λ_4 is a weight parameter that guarantees the consistency of X and H.

Through variable splitting technique, (5) will be further addressed by solving the following two subproblems

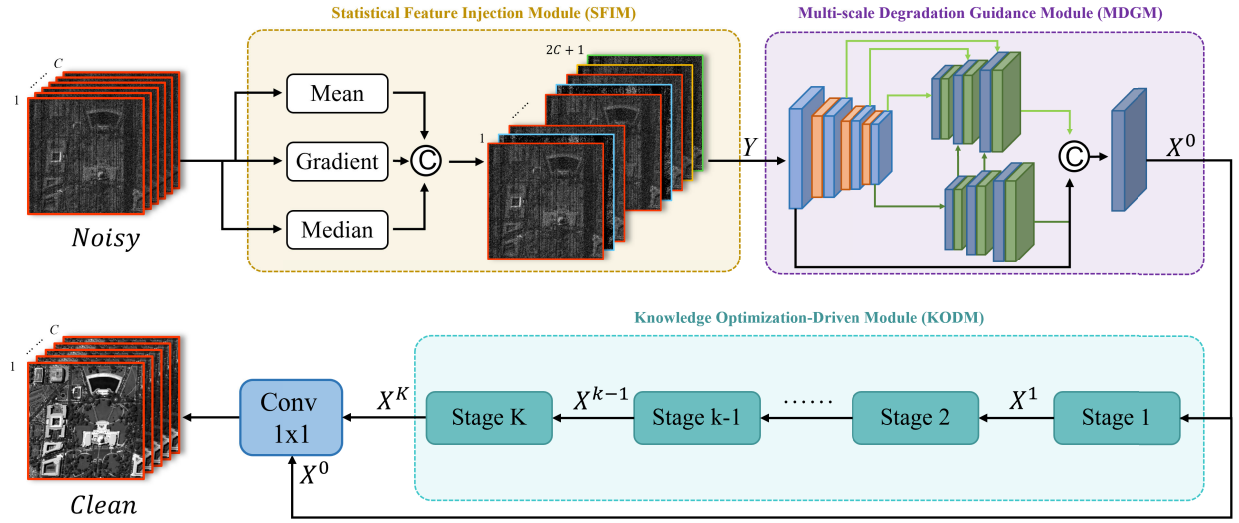

Fig. 1. Overall architecture of the KPInet consists of three modules: SFIM (the orange box), MDGM (the purple box), and KODM (the cyan box). The noisy HSI is fed into these three modules to obtain the estimated clean HSI.

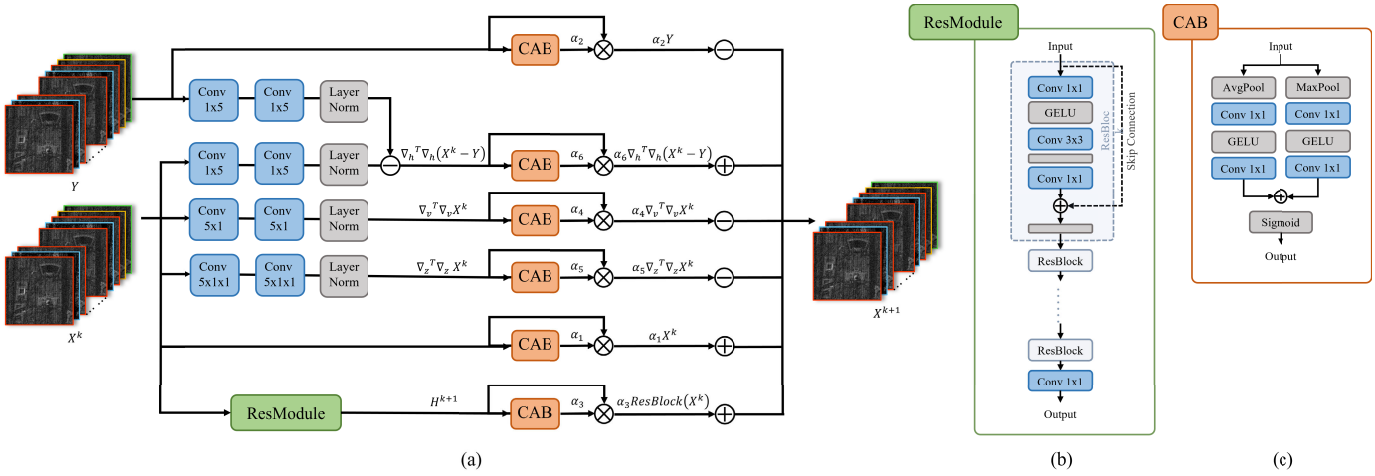

Fig. 2. Knowledge optimization-driven module (KODM), which is based on (10). (a) Overall flow of the module. (b) Residual module (ResModule). (c) CAM.

iteratively:

$$\begin{cases} \hat{X} = \underset{X}{\operatorname{argmin}} \|Y - X\|_2^2 + \lambda_1 \|\nabla_v X\|_2^2 + \lambda_2 \|\nabla_z X\|_2^2 \\ \qquad +\lambda_3 \|\nabla_h Y - \nabla_h X\|_2^2 + \lambda_4 \|X - H\|_2^2, \\ \hat{H} = \underset{H}{\operatorname{argmin}} \|H - X\|_2^2 + \mu \mathcal{J}(H) \end{cases} \tag{6}$$

where $\hat{H}$ represents the estimated H, and argmin_H means that this optimization problem is a minimization problem.

For the X-subproblem, we use the gradient descent algorithm to solve it, and the iterative formula is expanded as

$$\begin{aligned} \hat{X}^{k+1} &= X^k - \epsilon \big[X^k - Y + \lambda_1 \nabla_v^T \nabla_v X^k + \lambda_2 \nabla_z^T \nabla_z X^k \\ &\qquad + \lambda_3 \nabla_h^T \nabla_h (X^k - Y) + \lambda_4 (X^k - H^k) \big] \\ &= (1 - \epsilon - \epsilon\lambda_4) X^k + \epsilon Y + \epsilon\lambda_4 H^k - \epsilon\lambda_1 \nabla_v^T \nabla_v X^k \\ &\quad - \epsilon\lambda_2 \nabla_z^T \nabla_z X^k - \epsilon\lambda_3 \nabla_h^T \nabla_h (X^k - Y) \end{aligned} \tag{7}$$

where ϵ is the optimization stride, and $k \in [0, K)$ is the number of iteration times.

The H-subproblem is an optimization problem that requires finding the value of H that minimizes the objective function. Here, $\mathcal{J}(H)$ implicitly extracts the prior knowledge of H by imposing certain constraints. From a mathematical perspective, the H-subproblem is an optimization problem with both a differentiable function $\|H - X\|_2^2$ and implicit prior knowledge $\mathcal{J}(H)$. Therefore, we use a proximal operator $\operatorname{Prox}(\cdot)$ to solve the H-subproblem

$$\begin{aligned} \hat{H}^{k+1} &= \operatorname{Prox}(X^{k+1}) \\ &= \underset{H}{\operatorname{argmin}} \left\| H - X^{k+1} \right\|_2^2 + \mu \mathcal{J}(H). \end{aligned} \tag{8}$$

With the help of the half-quadratic splitting and the gradient descent algorithm, the problem of solving (3) is transformed into the process of alternately updating (7) and (8).

B. Knowledge-Driven CNN With Prior Injection

In Section III-A, we derived formulations that can better deal with structured noise, which is based on (7) and (8).

For the complex implicit prior in the H-subproblem, it can be learned by the CNN. However, for the X-subproblem, it contains multiple parameters that require manual intervention. Therefore, directly solving the two subproblems alternately according to the formula is equivalent to embedding the data-driven algorithm into the model-driven algorithm. This approach will not only decrease the computational efficiency but also greatly limit the optimization capability of CNN. To solve this problem, the proposed KPInet is dedicated to unfold the mathematical model into the end-to-end CNN by replacing the operational process in the X-subproblem with the convolutional process and the tensor operation.

Next, the overall structure of the proposed KPInet is presented in Fig. 1, and details of the network will be presented in Sections III-B1–III-B3.

1) Knowledge Optimization-Driven Module: For the H-subproblem, the proximal operator $\mathrm{Prox}(\cdot)$ in (8) can be replaced by any deep learning network [29] because of the mathematical equivalence of regularized denoising. The KODM uses the residual module (ResModule) instead, which has become an important part of CNNs and has been widely used in various tasks [53], [54], [55]. In the ResModule, the input data is first fed into a convolutional layer and activation function and then be added into the original input. This design strengthens the flow of information and avoids the problem of gradient vanishing, which also improves the accuracy and stability. After substitution, the formula is written as

$$\hat{H}^k = \mathrm{ResModule}(X^k). \tag{9}$$

The ResModule comprises multiple residual blocks (ResBlocks) and utilizes skip connections to facilitate residual learning. The specific structure is illustrated in Fig. 2(b). Each ResBlock consists of three convolutional layers and employs the Gaussian error linear unit (GELU) activation function [56]. Initially, the input data are passed through the first convolutional layer using a 1×1 kernel. This operation increases the number of channels to enhance the representation of spatial–spectral features. Subsequently, the data are forwarded to the second convolutional layer, which utilizes a 3×3 kernel to extract more abstract features without altering the number of channels. Finally, the output is fed into the third convolutional layer, which employs a 1×1 kernel to reduce the number of channels. To introduce nonlinearity, a GELU activation function is applied after each convolutional layer. Additionally, the output from each ResBlock is combined with the original input through a skip connection, followed by another GELU activation function. This process is repeated for several ResBlocks. Ultimately, the final output is passed through a convolutional layer with a 1×1 kernel to generate the final features.

With the help of the ResModule, we can get H^k. Next, the new equation will be given by rewriting (7) and (9). For the convenience of reading, we reformulate the coefficient terms of the variables in the new equation, which is represented as

$$\begin{aligned}\hat{X}^{k+1} = {} & \alpha_1 X^k + \alpha_2 Y + \alpha_3 \mathrm{ResModule}(X^k) - \alpha_4 \nabla_v^T \nabla_v X^k \\ & - \alpha_5 \nabla_z^T \nabla_z X^k - \alpha_6 \nabla_h^T \nabla_h (X^k - Y)\end{aligned} \tag{10}$$

where $\alpha_1 = 1 - \epsilon - \epsilon\lambda_4$, $\alpha_2 = \epsilon$, $\alpha_3 = \epsilon\lambda_4$, $\alpha_4 = \epsilon\lambda_1$, $\alpha_5 = \epsilon\lambda_2$, and $\alpha_6 = \epsilon\lambda_3$.

The one-way gradient solving process can be regarded as a first-order difference operation, achieved through a specialized convolution kernel. This approach forms the mathematical foundation for implementing the operations ∇ and ∇^T using 1-D convolution. Consequently, we replace $\nabla_h^T \nabla_h$, $\nabla_v^T \nabla_v$, and $\nabla_z^T \nabla_z$ with two 5×1 convolutions, two 1×5 convolutions, and two $5 \times 1 \times 1$ convolutions, respectively. Each 1-D convolution is subsequently followed by layer normalization (LN) [57]. Additionally, we utilize pixel-to-pixel tensor multiplication, addition, and subtraction to implement the formula. By employing 1-D convolutions and tensor operations in computer implementation, we can leverage the parallel computing capabilities of GPUs to enhance computational speed and efficiency.

Upon identifying suitable alternative operators, there still exist hyperparameters $\alpha_{i\in[1,6]}$ within the formula that require definition. Moreover, since HSI exhibits distinct radiation characteristics across different channels, these hyperparameters need to be calculated separately for each channel. The channel attention module (CAM) [58], [59] can learn a channelwise attention coefficient vector that assigns varying weights to different channels based on their significance in representing the target object or context. Consequently, we update the $\alpha_{i\in[1,6]}$ values by utilizing CAM to fully exploit the spectral information present in HSI, thereby avoiding the manual setting of hyperparameters for adaptation. The specific structure of CAM is depicted in Fig. 2(c). Initially, the input feature is fed into global average pooling and global max pooling, yielding two pooling feature vectors. These vectors are then passed through two 1×1 convolutions to obtain two channel attention vectors. Subsequently, the vectors undergo activation through two GELU functions. The resulting vectors are then added together after being passed through another round of two 1×1 convolutions. Finally, a sigmoid activation function is applied to obtain a channel attention coefficient vector. The CAM enables the network to emphasize informative channels while suppressing irrelevant ones.

With the help of KODM, we can iteratively update our X^k using the initial values X^0 and Y, ultimately achieving a desirable outcome. To improve the performance of KODM in removing structural noise, we aim for the X^0 to have minimal Gaussian noise, while Y should have stronger feature representation capability. In the following parts, SFIM can obtain the enhanced Y and MDGM performs coarse reconstruction on X^0 with reduced Gaussian noise.

2) Statistical Feature Injection Module: Advanced feature extraction from input data improves the network's capacity to utilize valuable information, thereby improving its efficiency and accuracy. To enhance the feature representation ability of the Y, it is crucial to incorporate additional prior knowledge into the input noisy images through feature extraction. Our analysis of statistical information obtained from real noisy HSIs reveals that gradient features between adjacent bands contribute to enhancing sparse noise features such as impulse noise, stripes, and deadlines. Similarly, computing the median value across multiple bands effectively attenuates these sparse

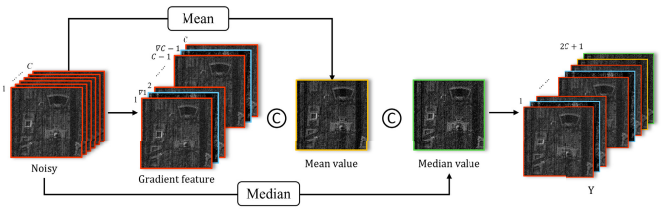

Fig. 3. Statistical feature injection module (SFIM).

noise elements. This is because they are sparse and typically manifest as extreme values resulting from sudden pixel value changes. Additionally, computing the mean values across multiple bands can slightly mitigate the impact of Gaussian noise on the image structure. Leveraging these statistical features, our proposed SFIM extracts the spectral gradient, mean, and median from the noisy images, thereby providing additional features for Y, as illustrated in Fig. 3.

Specifically, given a noisy image, we calculate the spectral gradient image ∇_c between its bands $c \in (0, C)$ and $c+1$ and insert it between the two bands. This process expands the number of channels from C to $2C-1$. Subsequently, we compute the median and mean values of the noisy image, which are concatenated as additional bands at the end of the image. This augmentation of statistical information provides the network with richer features. After undergoing the SFIM, the number of channels expands from C to $2C+1$.

Through the extraction of statistical features from the noisy image, the SFIM achieves a more comprehensive representation of the data. The resulting Y, enriched with additional prior features, enables the network to gain a better understanding of the data, thereby enhancing its performance in the given task.

3) Multiscale Degradation Guidance Module: In HSI processing, multiscale information is very important [60], [61]. It has been observed that low-resolution images tend to have less noise compared to high-resolution images. Building upon this observation, this article proposes the MDGM which utilizes low-resolution images as priors to guide the denoising process and ultimately obtain a coarse reconstruction denoted as X^0. The structure of MDGM is shown in Fig. 4.

To incorporate both spatial and spectral information efficiently, we adopt a joint 1-D–2-D convolution approach instead of the computationally intensive 3-D convolution [42], which considers spatial and spectral dimensions simultaneously. In our architecture, the 1-D convolution with a kernel size of $5 \times 1 \times 1$ operates on the spectral dimension, while the 2-D convolution with a kernel size of $1 \times 5 \times 5$ operates on the spatial dimensions. Subsequently, a joint convolution is applied, followed by LN, and another joint convolution followed by the GELU activation function, forming the convolution block (ConvBlock).

In the encoder–decoder structure, adjacent pixel values are merged into a single pixel during the downsampling process of the encoder. This merging process effectively reduces high-frequency structured noise and produces a less noisy low-resolution image in the final layer. To leverage this characteristic, MDGM employs an encoder–dual-decoder structure, which guides the extraction of multiscale information by utilizing low-resolution information. Each downsampling layer in the encoder consists of a max-pooling layer and a ConvBlock. Conversely, each upsampling layer in the dual-decoder comprises a ConvBlock followed by a transpose convolution. The dual-decoder consists of two branches: the original upsampling branch, which retains more structural features of the original image through skip connections, and the low-resolution upsampling guidance branch, which employs the low-resolution image to guide the image restoration process. Importantly, the results of the low-resolution guidance branch at each scale are solely derived from the bottom low-resolution image. This structural design enables the network to better discern the image structure from structural noise during the upsampling process, leading to more effective noise removal.

The MDGM effectively removes a substantial amount of Gaussian and impulse noise, as well as some deadlines and stripes. With the MDGM, we can obtain better coarse reconstruction X^0 from the original noisy image. However, the obtained X^0 still exhibit a significant amount of stripe-like artifacts. This is because the structural noise that spans the entire image is challenging to remove by filter-like convolutions. Then, the X^0 is fed into the KODM to generate the final denoising result $\hat{X}$.

C. Implementation Details

1) Loss Function: Although the $L2$ loss function is commonly used in many data-driven methods, it has some limitations in the denoising problem. The squared term of $L2$ loss is very sensitive to noise and small detail changes, which may cause oversmoothing or detail loss. In contrast, the $L1$ loss function preserves details better and is more robust to noise and small changes. That is because it linearly reduces the loss when the error is small. Furthermore, the $L1$ loss helps the network to learn sparse noise more efficiently. Therefore, using $L1$ loss is more appropriate for training the KPInet.

Structural noise can significantly destroy the gradient of the image. Specifically, the gradient along the direction of the structural noise will decrease, and the gradient through the direction of the structural noise will increase. Considering these effects, in addition to using an overall $L1$ loss between the restored image $\hat{X}$ and ground truth X, we also apply $L1$ loss on the gradients of horizontal and vertical directions to better suppress structural noise. These loss functions are represented as $\mathcal{L}_1 = (1/n)\sum_{i=1}^{n}|X_i - \hat{X}_i|$, $\mathcal{L}_2 = (1/n)\sum_{i=1}^{n}|\nabla_v X_i - \nabla_v \hat{X}_i|$ and $\mathcal{L}_3 = (1/n)\sum_{i=1}^{n}|\nabla_h X_i - \nabla_h \hat{X}_i|$, respectively.

To automatically weight the contribution of each loss, we use a parameter adaptive strategy [62] which weighs multiple loss functions by considering the homoscedastic uncertainty of each loss. We compute the weighted sum of the three parts of the loss function and obtain the final loss

$$\mathcal{L} = \sum_{i=1}^{n} \frac{1}{2\sigma_i^2}\mathcal{L}_i + \log\left(1+\sigma_i^2\right), \quad \sigma_i = \exp(p_i) \tag{11}$$

where $\mathcal{L}$ is the final loss, $\mathcal{L}_i$ is the ith loss, σ_i is the homoscedastic uncertainty of the ith task, and p_i is the learnable parameter used to estimate σ_i.

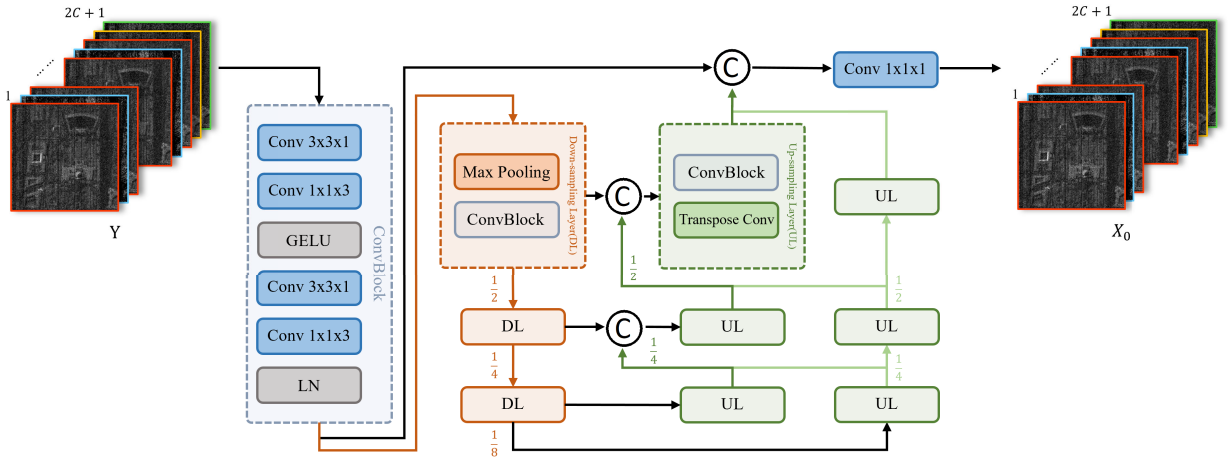

Fig. 4. Multiscale degradation guidance module (MDGM).

2) LN and GELU: In traditional CNNs, batch normalization (BN) and rectified linear units (ReLUs) are widely used for better performance. However, ReLU suffers from the vanishing gradient problem, where the gradient becomes zero for negative input values, thereby affecting the training efficiency of the model. Additionally, BN requires batch processing, which limits its use in small batch sizes and may cause overfitting in some cases. To address these issues, alternative functions such as LN and GELU have been proposed, which have shown to be advantageous over BN and ReLU in terms of computational efficiency, improved accuracy, and better generalization performance. Specifically, LN does not rely on batch statistics and can adaptively normalize feature maps with higher accuracy. And GELU has a smoother derivative than ReLU, leading to better gradient propagation and thus faster convergence. These two approaches have been proven to improve the performance of CNNs in the new powerful network, ConvNeXt [63]. Therefore, we use LN instead of BN and GELU instead of ReLU in the ConvBlock to bring better performance.

IV. Experimental Results and Analysis

In this section, we demonstrate the superiority of KPInet through simulated and real experiments. We also present the parameter settings through parametric analysis and demonstrate the necessity and effectiveness of our proposed module through ablation experiments.

A. Experimental Setting

1) Datasets: In simulation experiments, we use the Washington DC Mall (WDC) dataset. It is collected by the hyperspectral digital imagery collection experiment (HYDICE) sensor. The image has a size of 1208 × 307 pixels and consists of 210 spectral bands. However, only 191 bands are considered for analysis after eliminating noisy bands. The size of the training image patches used in our experiments is 128 × 128 × 32, and a total of 22 200 patches are used for training. For testing, we use the spectral grouping strategy, which divides data of size 256 × 256 × 191 into six patches of size 256 × 256 × 32.

In real experiments, we select four satellite datasets, which are affected by mixed noise, especially stripe noise.

1) Zhuhai-1 (ZH-1) dataset, which is affected by wide strips at the left image edge, is cropped to 256 × 256 × 32 for testing.
2) Gaofen-5 (GF-5) dataset, which is affected by periodic wide strips and Gaussian noise, is cropped to 256 × 256 × 150 for testing.
3) Earth Observing-1 (EO-1) Hyperion dataset, which is affected by complex noise, is cropped to 400 × 200 × 166 for testing.
4) HYDICE Urban dataset, which is affected by extremely heavy complex noise, is cropped to 307 × 307 × 210 for testing.

2) Quantitative Evaluation Metrics: To evaluate the performance of the proposed method, we utilize three commonly used metrics: mean peak signal-to-noise ratio (mPSNR), mean structural similarity index (mSSIM), and spectral angle mapper (SAM). mPSNR is a widely used metric that measures the quality of the reconstructed image by computing the mean square error (mse) between the original and reconstructed images. A higher mPSNR value indicates better image quality. mSSIM is another popular metric that measures the structural similarity between the original and reconstructed images in terms of luminance, contrast, and structure. A higher mSSIM value indicates better structural similarity. SAM measures the spectral similarity between the two images by computing the angle between their pixel vectors. Lower SAM values indicate better spectral similarity. The definitions of these indices are as follows:

$$\text{mPSNR} = \frac{1}{C}\sum_{c=1}^{C} 10 \cdot \log_{10} \times \left(\frac{R^2 \cdot W \cdot H}{\sum_{i=1}^{W}\sum_{j=1}^{H}(X_c(i,j) - \hat{X}_c(i,j))^2} \right) \tag{12}$$

$$\text{mSSIM} = \frac{1}{C}\sum_{c=1}^{C} \frac{(2\mu_{X_c}\mu_{\hat{X}_c} + S_1)(2\sigma_{X_c\hat{X}_c} + S_2)}{(\mu_{X_c}^2 + \mu_{\hat{X}_c}^2 + S_1)(\sigma_{X_c}^2 + \sigma_{\hat{X}_c}^2 + S_2)} \tag{13}$$

$$\text{SAM} = \arccos\left(\frac{a \cdot b}{\|a\|\|b\|} \right) \tag{14}$$

TABLE I
QUANTITATIVE RESULTS OF DIFFERENT METHODS ON WASHINGTON DC MALL, THE BEST INDEX IS PLOTTED IN RED, AND THE SECOND BEST INDEX IS PLOTTED IN BLUE

	Indexes	NOISE	LRMR	LRTV	LRTDTV	FastHyDe	FastHymMix	QRNN3D	T3SC	NSSNN	KPInet
case1	mPSNR↑	22.6069	27.7504	35.0111	32.4990	29.2197	33.0232	34.3886	30.3913	33.8274	34.8634
	mSSIM↑	0.5242	0.7526	0.9357	0.9068	0.7936	0.9078	0.9460	0.8746	0.9500	0.9494
	SAM↓	30.1776	20.3003	9.7425	12.0549	17.0825	11.1350	5.8028	8.8963	6.3607	5.5359
case2	mPSNR↑	22.4456	26.6742	34.4494	31.5361	28.2621	33.1383	33.5029	29.7841	33.0358	34.6449
	mSSIM↑	0.4923	0.7254	0.9295	0.8824	0.7521	0.8809	0.9415	0.8864	0.9273	0.9435
	SAM↓	36.2692	27.5954	14.0287	17.8957	24.1962	16.8217	6.5401	10.1245	7.6190	6.2421
case3	mPSNR↑	20.9891	27.2013	33.8279	32.5307	34.2259	37.6515	34.2705	32.4308	36.4963	37.8214
	mSSIM↑	0.4037	0.6703	0.9014	0.8688	0.9275	0.9511	0.9400	0.9091	0.9572	0.9595
	SAM↓	33.3444	17.5828	11.1958	9.7665	4.6313	5.3932	5.8955	6.4595	4.9928	4.3340
case4	mPSNR↑	19.4228	24.7926	30.7803	29.7016	27.5075	32.4079	32.5055	29.7171	32.5531	34.3226
	mSSIM↑	0.3687	0.6296	0.8637	0.8316	0.7334	0.8391	0.9260	0.8746	0.9190	0.9410
	SAM↓	42.2492	29.4958	17.5541	19.1724	23.5804	19.3175	7.2827	10.1539	7.7381	6.9578
case5	mPSNR↑	17.1846	22.7049	26.3104	28.1205	26.0106	30.7798	31.5794	28.7916	31.8599	33.4757
	mSSIM↑	0.2724	0.5303	0.6704	0.7786	0.6724	0.7986	0.9069	0.8566	0.9233	0.9312
	SAM↓	48.9299	33.6302	38.3190	24.2774	26.4951	22.2988	7.7024	10.8930	7.8978	6.7156

where R represents the maximum possible pixel value. $X_c(i, j)$ and $\hat{X}_c(i, j)$ denote the pixel values of the clean and processed images in channel c, respectively. μ_{X_c} and $\mu_{\hat{X}_c}$ represent the mean values of the clean and processed images in channel c, respectively. $\sigma_{X_c\hat{X}_c}$ stands for the covariance between the clean and processed images in channel c. σ_{X_c} and $\sigma_{\hat{X}_c}$ represent the variances of the clean and processed images in channel c. S_1 and S_2 are constants introduced to stabilize the division. a represents the spectral vector of the clean image, and b represents the spectral vector of the processed image. Additionally, $||a||$ and $||b||$ represent the magnitudes of the spectral vectors for the clean and processed images, respectively.

3) Comparison Methods: We compare the proposed KPInet with seven denoising algorithms, including LRMR [17], LRTV [19], LRTDTV [18], FastHyDe [39], FastHyMix [40], QRNN3D [33], T3SC [48], and NSSNN [34]. The first five are model-driven methods, and the last three are data-driven methods.

4) Simulation Experiments: For the training dataset, we introduce various types of noise to each band. Specifically, we add Gaussian noise with a standard deviation σ ranging from 0 to 75. Additionally, 30% of the bands are contaminated with impulse noise, where the intensities vary between 0.01 and 0.1. 30% of the bands contain deadlines, with densities ranging from 5% to 15%. Similarly, 30% of the bands have thin stripes, with densities varying from 15% to 65%. Finally, 30% of the bands are affected by one to ten wide stripes, where the widths range from 10 to 25. This diverse range of noise types and intensities ensures that the training dataset encompasses various challenging scenarios commonly encountered in practical applications.

For real HSI, the randomness of the noise is manifested in the following aspects: random intensity, random density, and it appears on random bands. In order to make our experimental results more realistic and credible, we randomly add noise to the WDC test dataset by the following settings.

1) *Case 1 (Gaussian Noise + Stripe Noise):* Gaussian noise with σ ranging from 0 to 55 is added to each band, and 30% of the bands have thin stripes with densities ranging from 15% to 50%.
2) *Case 2 (Gaussian Noise + Stripe Noise + Impulse Noise):* On the basis of *Case 1*, we add impulse noise with intensities ranging from 0.01% to 0.1% to 30% of the bands.
3) *Case 3 (Gaussian Noise + Stripe Noise + Impulse Noise + Deadlines):* On the basis of *Case 2*, we add deadlines with densities ranging from 5% to 15% to 30% of the bands.
4) *Case 4 (Gaussian Noise + Stripe Noise + Impulse Noise + Deadlines + Wild Stripes):* On the basis of *Case 3*, since wide stripes are also present in real remote sensing images, we add one to ten wide stripes with widths ranging from 5% to 10% to 30% of the bands.
5) *Case 5 (Gaussian Noise + Stripe Noise + Impulse Noise + Deadlines + Wild Stripes):* We increase the noise intensity of Gaussian noise and strips based on *Case 4*. Gaussian noise with σ ranging from 0 to 75 is added to each band. We add one to ten wide stripes with widths ranging from 10% to 25% to 30% of the bands and thin stripes with densities ranging from 15% to 65% to 30% of the bands.

5) Training Details: In the MDGM module, we employ 3-D convolution with channel combinations of [2, 8, 16]. For the KODM module, we utilize six stages, each consisting of six ResBlocks. This specific combination has consistently demonstrated superior performance for our network, as discussed in Section II. As for optimization, we adopt the Adam optimizer with a learning rate of $1e^{-4}$, betas set to (0.9, 0.999), and a weight decay of $1e^{-4}$. To control the learning rate during

training, we incorporate the cosine annealing learning rate strategy. With a total of ten epochs and a minimum learning rate of $1e^{-6}$, this scheduler facilitates a gradual reduction in the learning rate. Such an approach helps prevent the model from becoming trapped in local minima and enhances its generalization capabilities. Our training process is conducted on a Linux system, utilizing two NVIDIA RTX 2080Ti GPUs, each with 11 GB of memory. The specific software and hardware specifications include Python version 3.8.5, PyTorch version 1.11.0 + cu113, and CUDA version 11.3. Furthermore, all model-driven methods are implemented on a Windows 10 system using MATLAB R2021, which runs on an AMD Ryzen 9 5950X 16-core CPU.

B. Simulation Experiments

1) Quantitative Results: We conducted simulation experiments on WDC for removing mixed noise, and the quantitative results are presented in Table I. The noise intensity varies across the five cases due to the different distributions of random noise. We have sorted the noise levels from low to high for Cases 1–5. Since Gaussian noise with a high intensity of $\sigma = 50$ is added to each band, the mPSNR values are generally low. The mSSIM values exhibit significant variation, mainly due to the varying degrees of structural noise damage to the test image. Overall, our proposed method outperforms the compared methods in nearly all cases, achieving the best results.

For the cases with the strongest noise, Cases 4 and 5, KPInet demonstrates the best performance. Specifically, KPInet achieves significantly higher mSSIM values than other methods, validating our original design intent of the network to better handle structural noise. Additionally, we observed that data-driven methods outperform model-driven methods, highlighting the powerful nonlinear fitting capabilities of neural networks in addressing complex problems.

In Case 3, model-driven methods outperform data-driven methods. However, we notice that the performance of model-driven methods vary across different cases due to the need for manual parameter tuning. The parameter set used achieves optimal results in Case 3. In contrast, data-driven methods consistently achieve stable and superior performance across different cases without the requirement for manual parameter tuning.

In the case of the weakest noise, Case 1, KPInet exhibits a slightly lower mSSIM value by 0.0006 dB compared to NSSNN, but the mPSNR results are higher than QRNN3D and NSSNN. Since the structural noise in this case is weak, the differences between KPInet and other methods are not significant. However, as the structural noise intensifies in the other cases, the advantages of KPInet become more pronounced. The KODM architecture in KPInet is specifically designed to effectively handle structural noise, allowing it to deliver superior results compared to other methods. As the structural noise becomes stronger, the capability of KPInet to preserve image structure and details are particularly beneficial, leading to improved denoising performance.

Additionally, the results obtained from T3SC are somewhat underwhelming. This could potentially be attributed to the design of algorithm, which seems to primarily address specific noise scenarios. Given that our task involves the simultaneous presence of strong structural noise and Gaussian noise, there might be some challenges in accurately estimating the underlying structure.

2) Qualitative Results: We present qualitative results using the strongest noise case, Case 5, as shown in Fig. 5. The noisy image contains Gaussian noise, impulse noise, dense thin stripes, deadlines, and several wide stripes. One of the wide stripes is located at the edge of the image and is up to 25 pixels wide.

First, we analyze the overall performance of the methods. Among the model-driven methods, LRMR can only remove some of the Gaussian noise. FastHyDe can remove more Gaussian noise, but the stripe removal is not clean. LRTV and LRTDTV leave heavy stripe artifacts. FastHyMix can remove most of the Gaussian noise and stripes, but the results for wide stripes and deadlines are poor, resulting in significant spectral distortion. Among the data-driven methods, QRNN3D, T3SC, and NSSNN can effectively remove Gaussian noise, deadlines, and thin stripes. However, QRNN3D makes the image blurry and loses more details, T3SC struggles to effectively handle wide stripes, and NSSNN still leaves a small amount of artifacts. Additionally, the results of both QRNN3D and NSSNN still show residual wide stripes in the yellow band. Finally, the proposed KPInet exhibits significant improvements in denoising. We provide two zoomed-in view, which clearly demonstrate that our method effectively removes the stripes and successfully recovers the structural details of buildings, roads, and vehicles in the image. There are fewer residual stripes in our results, which further demonstrates the excellent performance of KPInet in removing structural noise. The superior performance of KPInet in denoising demonstrates its effectiveness and highlights its potential for applications in denoising tasks.

Next, we show band 44 of the WDC test data in Fig. 6, which is corrupted with dense deadlines, some stripes, and Gaussian noise. LRMR, LRTV, and LRTDTV generate denoised results that still contain prominent stripes and residual Gaussian noise. FastHyDe and FastHyMix effectively remove the Gaussian noise but still leave behind noticeable and prominent stripe artifacts. The reconstructions of data-driven methods QRNN3D, T3SC, NSSNN, and KPInet are relatively better. Furthermore, we provide a zoomed-in view. It can be observed that both QRNN3D and NSSNN are affected by stripes and exhibit discontinuities in the horizontal direction. However, KPInet better reconstructs the corrupted information in the deadline area and ensures greater continuity of image information horizontally. T3SC is also proficient in addressing deadlines. This can be attributed to its sparse coding approach, which leverages information from other bands to recover missing pixels effectively.

C. Real Experiments

1) ZH-1 Dataset: The results of the real experiment on the ZH-1 dataset in bands (27, 14, 11) are shown in Fig. 7. The ZH-1 dataset contains Gaussian noise and strips, often with wide strips exceeding a width of ten pixels that cannot

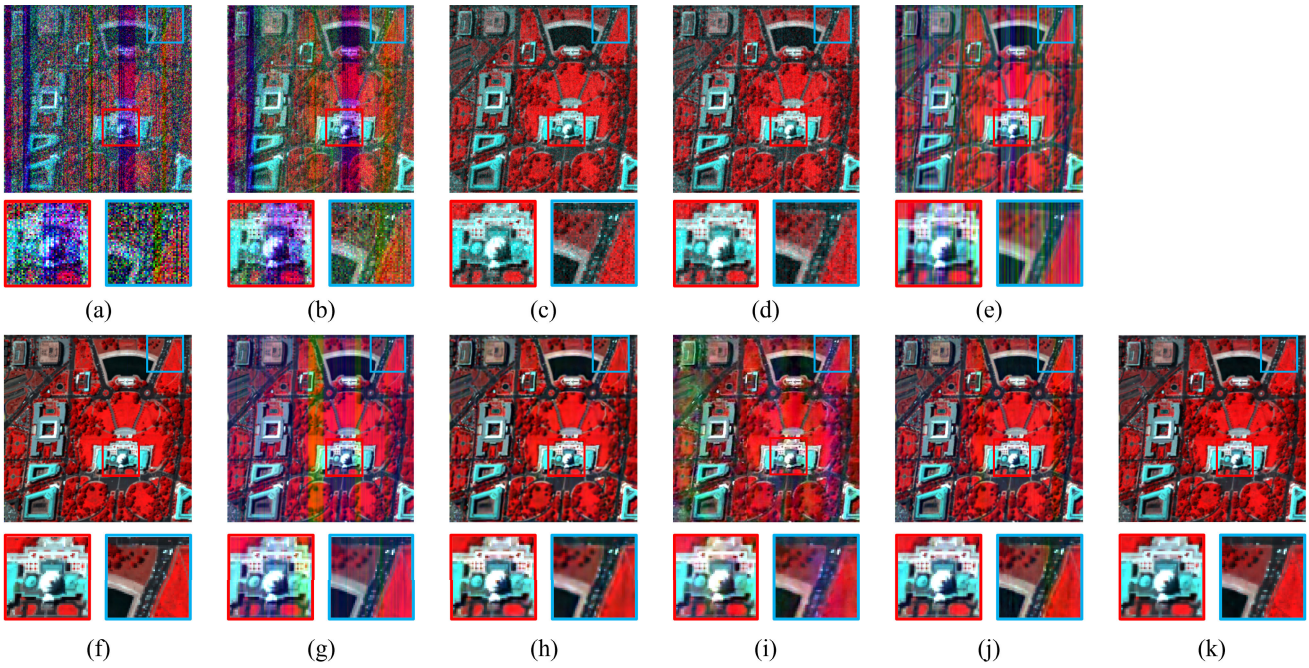

Fig. 5. Denoising results for WDC dataset in the synthetic data experiment, false color image with bands (60, 28, 19). (a) Noisy image. (b) LRMR. (c) LRTV. (d) LRTDTV. (e) FastHyDe. (f) Ground truth. (g) FastHyMix. (h) QRNN3D. (i) T3SC. (j) NSSNN. (k) KPInet.

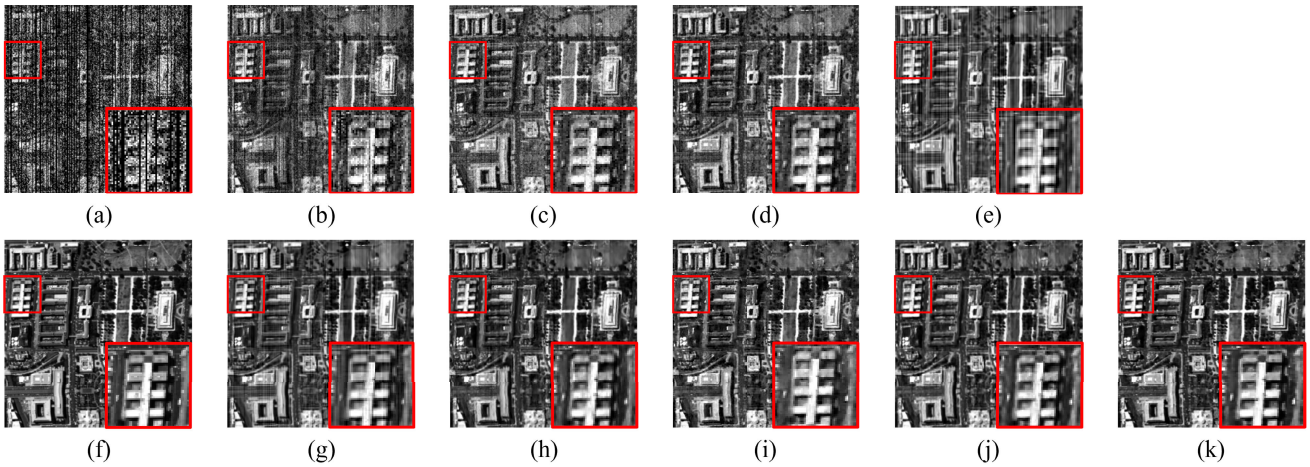

Fig. 6. Denoising results for WDC dataset in the synthetic data experiment, grayscale display with band 44. (a) Noisy image. (b) LRMR. (c) LRTV. (d) LRTDTV. (e) FastHyDe. (f) Ground truth. (g) FastHyMix. (h) QRNN3D. (i) T3SC. (j) NSSNN. (k) KPInet.

be effectively eliminated by most existing methods. Although LRTDTV and FastHyMix demonstrate good results, noticeable stripes still exist in the left side of the image, disrupting the horizontal continuity of the image. The results of T3SC are also impressive, but some areas still display color distortions attributable to the presence of wide stripes. LRMR, FastHyDe, QRNN3D, and NSSNN exhibit noticeable color distortion in the upper right corner of the urban area, resulting from the presence of wide stripes. This indicates their limited capability in handling wide stripes. In contrast, the proposed KPInet not only effectively removes stripes but also preserves the spectral correlation of the image, delivering better overall performance. Furthermore, we provide the horizontal digital number (DN) curve of the image in Fig. 8. In Fig. 8(a), large waves can be observed in the curve due to stripe interference. In comparison to other methods, the curves obtained by KPInet appear smoother, indicating the superior effectiveness of the proposed method in stripe removal.

2) GF-5 Dataset: The results of the real experiment on GF-5 dataset in bands (165, 135, 95) are presented in Fig. 9. We can observe that real images have periodic wide strips and Gaussian noise and are severely affected by stripe noise. LRMR, LRTV, FastHyDe, and FastHyMix are unable to remove so many stripes, which result in noticeable color distortion. Due to the excessive stripes, the deep learning models QRNN3D, T3SC, and NSSNN are unable to restore the image properly, resulting in multiple artifacts. The proposed KPInet in this article and LRTDTV remove periodic wide stripes in the image and have better performance than other methods Furthermore, we give the vertical DN curve of the

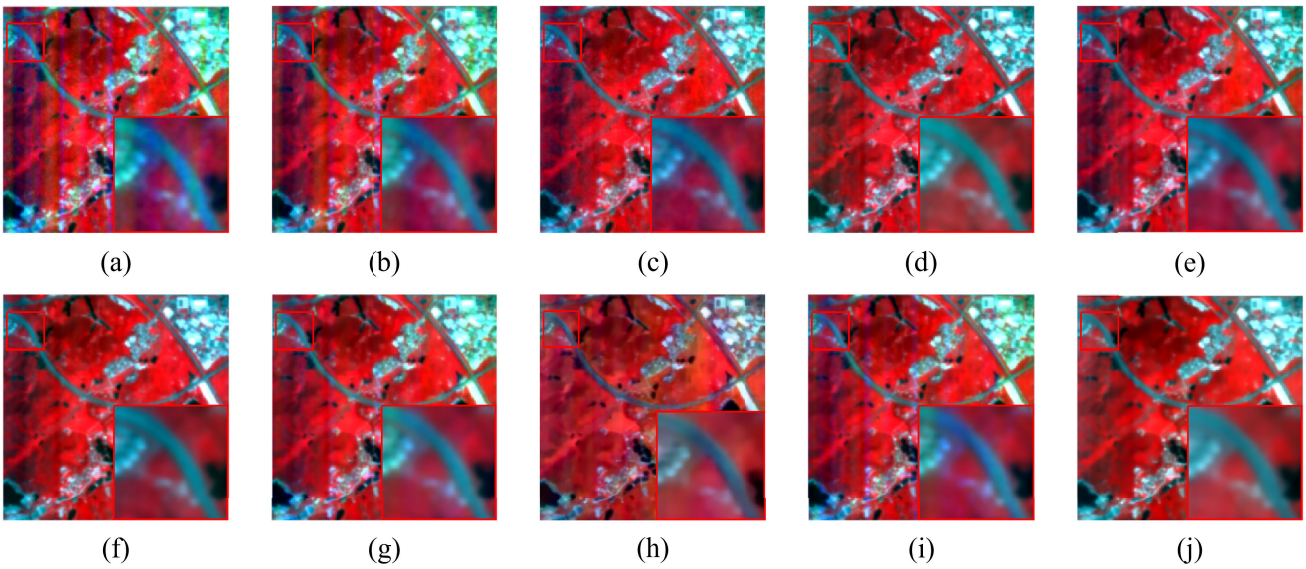

Fig. 7. Denoising results for ZH-1 dataset in the real data experiment, false color image with bands (27, 14, 11). (a) Noisy image. (b) LRMR. (c) LRTV. (d) LRTDTV. (e) FastHyDe. (f) FastHyMix. (g) QRNN3D. (h) T3SC. (i) NSSNN. (j) KPInet.

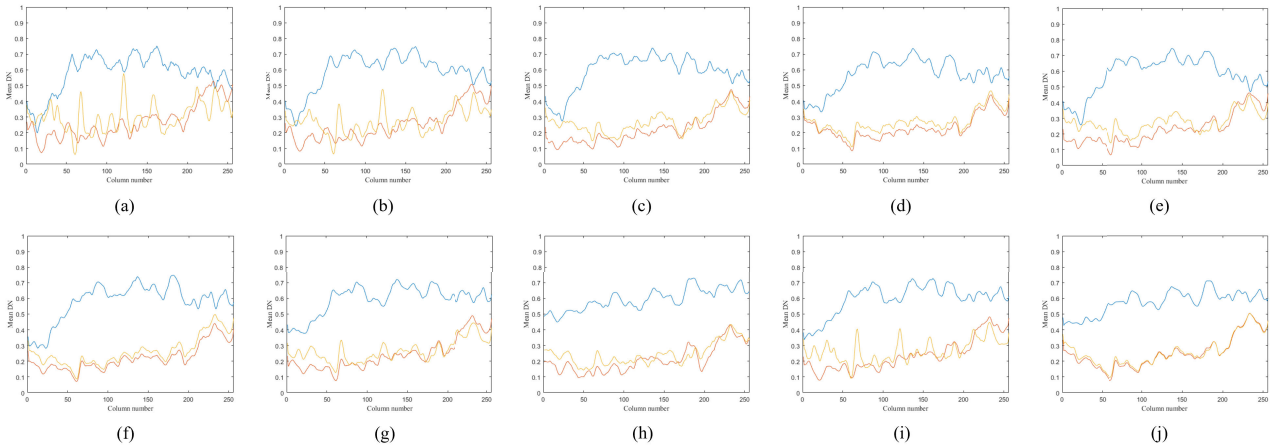

Fig. 8. Denoising results for ZH-1 dataset in the real data experiment, displayed in vertical mean DN curves of band (blue: 27, red: 14, and yellow: 11). (a) Noisy image. (b) LRMR. (c) LRTV. (d) LRTDTV. (e) FastHyDe. (f) FastHyMix. (g) QRNN3D. (h) T3SC. (i) NSSNN. (j) KPInet.

image in Fig. 10. In Fig. 10(a), there are large waves in the curve due to the interference of the stripes. Compared with other methods, the curves of the results obtained by KPInet are smoother, which proves that the method is able to remove the stripes effectively.

3) EO-1 Hyperion Dataset: The EO-1 dataset consists of images where the beginning and last few bands are damaged by Gaussian, deadlines, and stripe noise. The results of the real experiment on its bands 1 are displayed in Fig. 11. The left side of this band is affected by strong Gaussian noise, and none of the compared methods can effectively correct the local color deviation caused by Gaussian noise. However, KPInet excels in suppressing the whitish colors caused by noise on the left side of the image, resulting in a more consistent color representation across both the left and right sides of the image. Regarding strip noise, QRNN3D and NSSNN exhibit poor performance in handling wide stripes, leaving behind noticeable artifacts. In addition, they fail to remove the deadline in the middle. LRMR, FastHyDe, and FastHyMix all achieve some degree of effective strip noise removal, but none of them successfully eliminate the deadline in the middle. LRTV demonstrates the worst removal effect among the methods. Additionally, we provide local zoom-in images of LRTDT, T3SC, and KPInet, which demonstrate the best denoising effects. When compared to LRTDTV and T3SC, KPInet not only effectively removes stripes, deadlines, and Gaussian noise but also preserves and enhances fine details in the image's ground features.

4) Urban Dataset: The Urban dataset contains bands that are severely disturbed by complex noise, including dense strips in multiple directions, strong Gaussian noise, and impulse noise. The results of the real experiment on its bands 107 are displayed in Fig. 12. LRMR, LRTV, and FastHyDe are unable to restore the ground features effectively. NSSNN exhibits significant color distortion in the middle of the image and fails to reconstruct the image edges due to the heavy influence of strips. FastHyMix and QRNN3D result in uneven color in the restored image, with noticeable stripes still present.

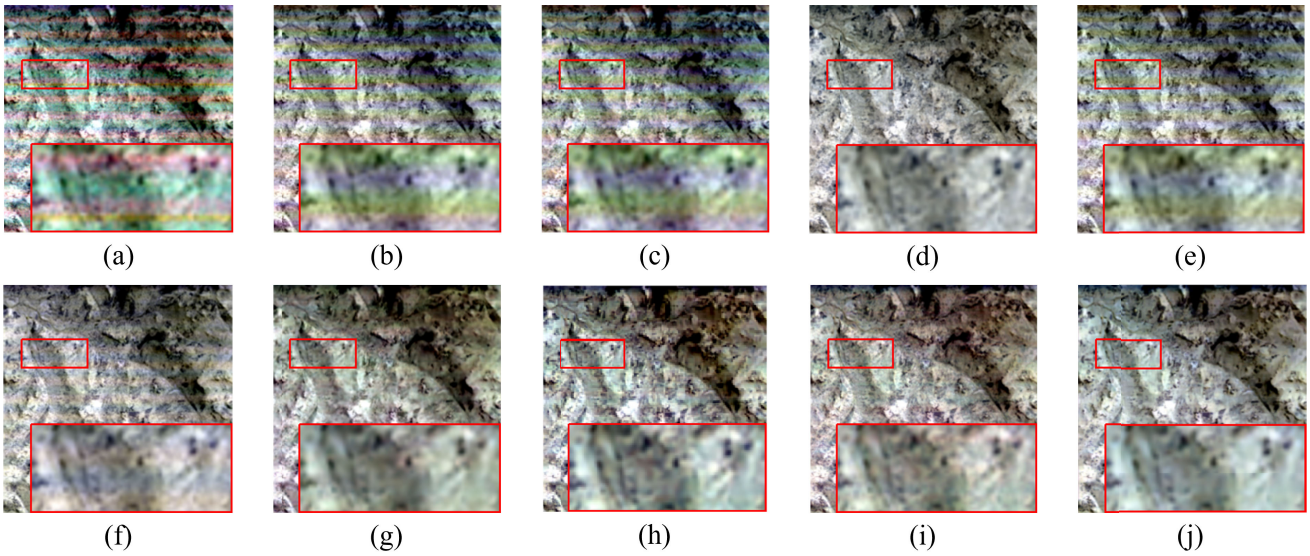

Fig. 9. Denoising results for GF-5 dataset in the real data experiment, false color image with bands (165, 135, 95). (a) Noisy image. (b) LRMR. (c) LRTV. (d) LRTDTV. (e) FastHyDe. (f) FastHyMix. (g) QRNN3D. (h) T3SC. (i) NSSNN. (j) KPInet.

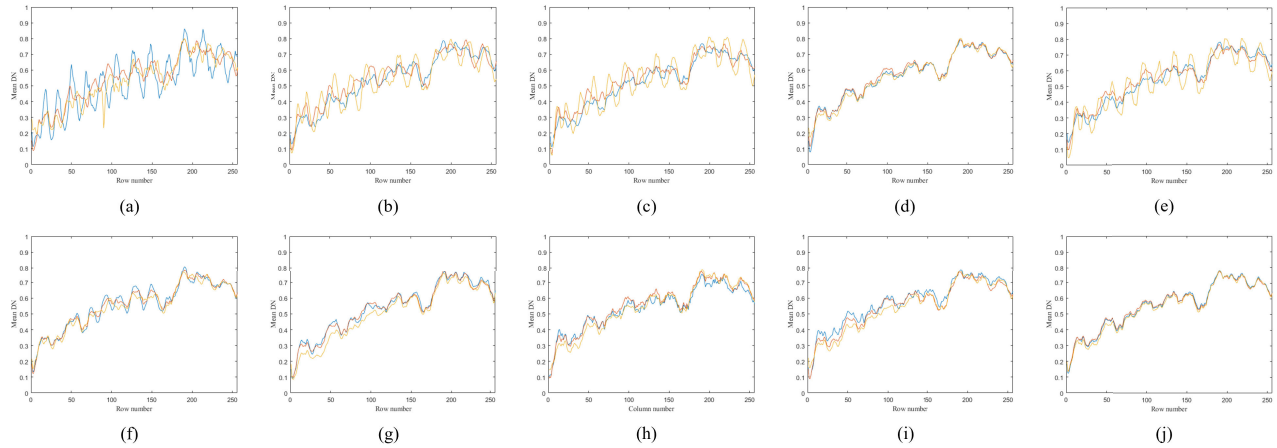

Fig. 10. Denoising results for GF-5 dataset in the real data experiment, displayed in horizontal mean DN curves of band (blue: 165, red: 135, and yellow: 95). (a) Noisy image. (b) LRMR. (c) LRTV. (d) LRTDTV. (e) FastHyDe. (f) FastHyMix. (g) QRNN3D. (h) T3SC. (i) NSSNN. (j) KPInet.

TABLE II
QUANTITATIVE EVALUATION RESULTS OF DIFFERENT NETWORK CONSTRUCTION IN CASE 5 ON WDC TEST DATA

SFIM	3D-UNet	MDGM	KODM	mPSNR↑	mSSIM↑	SAM↓
	✓			32.2476	0.8960	6.7481
		✓		32.5308	0.9084	6.7475
✓		✓		32.9001	0.9143	6.7172
✓		✓	✓	33.4757	0.9312	6.7156

Furthermore, we provide local zoom-in images of LRTDTV, T3SC, and KPInet, which demonstrate the best denoising effects. When compared with LRTDTV and T3SC, KPInet not only effectively removes stripes and Gaussian noise but also highly preserves the detailed features of the ground.

D. Ablation Study

1) Different Network Construction: In Table II, we conduct ablative experiments on Case 5 of the WDC dataset to analyze the impact of different modules added to the baseline 3D-UNet network. The first module we introduce is the low-resolution guidance strategy, MDGM, which leads to improvements in the mPSNR and mSSIM metrics. This indicates that incorporating low-resolution guidance enhances the network's denoising performance. Next, we introduce the proposed SFIM. Although there is no significant improvement in mPSNR and mSSIM, there is a noticeable decrease in SAM. This suggests that augmenting the data with statistical information and extracting gradient information between spectra effectively utilizes the spectral information in the denoising process. Finally, we incorporate the knowledge-driven optimization strategy, KODM, resulting in a significant enhancement in both mSSIM and mPSNR metrics. In addition, SAM has also decreased by 0.0016 dB. In comparison to the baseline, we observed a notable increase of 0.0325 dB in mSSIM and 1.2281 dB in mPSNR. This clearly highlights the effectiveness of KODM in improving the denoising performance.

In summary, the ablative experiments show that the addition of the MDGM, SFIM, and KODM modules progressively improves the denoising performance of the baseline 3D-UNet

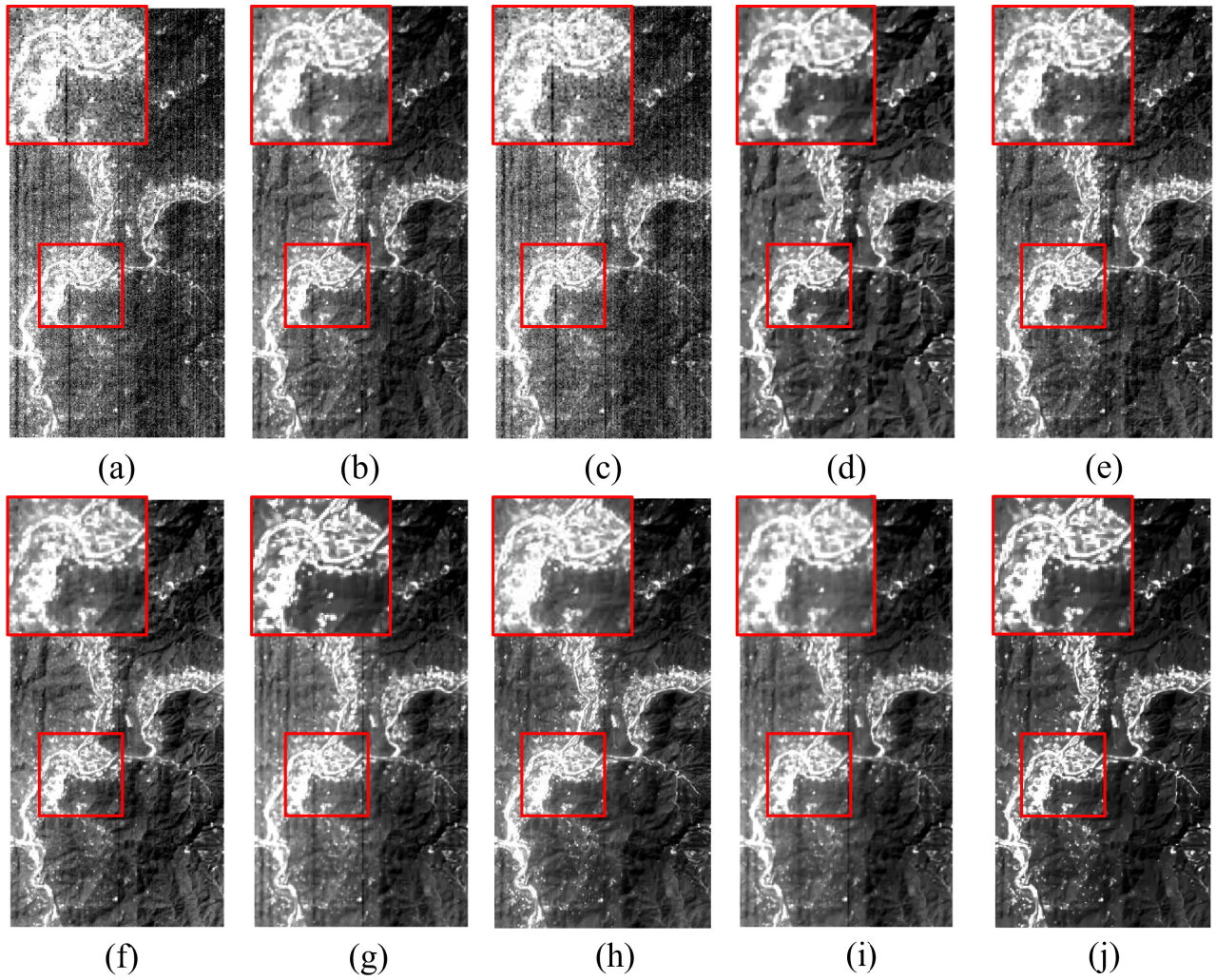

Fig. 11. Denoising results for EO-1 Hyperion dataset in the real data experiment, grayscale display with band 1. (a) Noisy image. (b) LRMR. (c) LRTV. (d) LRTDTV. (e) FastHyDe. (f) FastHyMix. (g) QRNN3D. (h) T3SC. (i) NSSNN. (j) KPInet.

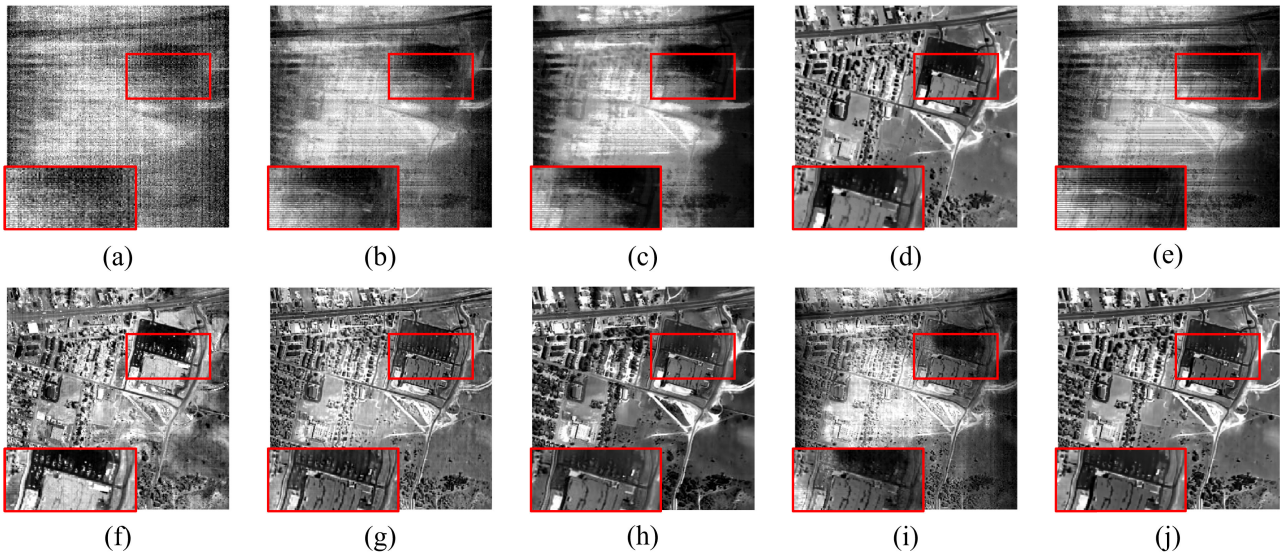

Fig. 12. Denoising results for HYDICE Urban dataset in the real data experiment, grayscale display with band 107. (a) Noisy image. (b) LRMR. (c) LRTV. (d) LRTDTV. (e) FastHyDe. (f) FastHyMix. (g) QRNN3D. (h) T3SC. (i) NSSNN. (j) KPInet.

network. The incorporation of these modules enhances the network's ability to handle mixed noise and exploit spectral information, leading to superior denoising results.

2) Different Loss Function: In Table III, we compare different loss functions on Case 5 of the WDC experiments. Compared with directly calculating the loss of the entire

TABLE III
COMPARISON OF DIFFERENT LOSS FUNCTIONS IN CASE 5 ON WDC TEST DATA

Loss	mPSNR↓	mSSIM↑	SAM↓
$\mathcal{L}_1$	32.9186	0.9265	7.7103
$\mathcal{L}_1$+$\mathcal{L}_2$	33.2198	0.9234	7.2920
$\mathcal{L}_1$+$\mathcal{L}_2$+$\mathcal{L}_3$	33.4757	0.9312	6.7156

TABLE IV
COMPARISON OF DIFFERENT NUMBER OF STAGES IN CASE 5 ON WDC TEST DATA

Number	mPSNR↓	mSSIM↑	SAM↓
5	32.2198	0.9234	7.2920
6	33.4757	0.9312	6.7156
7	31.9186	0.9165	7.7103

TABLE V
COMPARISON OF DIFFERENT RESBLOCK NUMBERS IN CASE 5 ON WDC TEST DATA

Number	mPSNR↓	mSSIM↑	SAM↓
5	33.0401	0.9254	6.8576
6	33.4757	0.9312	6.7156
7	33.3102	0.9309	6.7007

image, introducing the loss function with spatial gradient constraint leads to better image denoising results, with an improvement of 0.0047 dB in mSSIM and 0.5571 dB in mPSNR.

E. Parametric Analysis

Finally, a parameter analysis is conducted on Case 5 of the WDC experiments. In Table IV, the influence of the number of stages in the KODM on the network performance is discussed. It is determined that the network achieved the best performance in each metric when the number of stages is set to 6. In Table V, the impact of the number of ResBlocks used in each KODM layer on network performance is examined. Although using seven ResBlocks results in a slight decrease in SAM, it significantly increases the computational cost. Therefore, it is decided to select six ResBlocks to achieve the optimal results.

V. CONCLUSION

In this article, we propose an optimization-driven CNN that injected with prior information, named KPInet. The KPInet complements the advantages of the model-driven and data-driven method. In general, we propose three key modules. Through deep unrolling, a model considering noise structure is unfolded into an end-to-end CNN in the KODM. Second, the statistical information is used to extract more features in SFIM. Furthermore, the MDGM with a daul-stream decoder is guided by low-resolution information. Finally, we present the results of our experiments to demonstrate the effectiveness of KPInet on both simulated and real datasets. We specifically highlight its excellent ability to remove strip noise, especially the wide one. Additionally, we thoroughly discuss the effectiveness of each module in KPInet and analyze the sensitivity of the parameters used in the framework. While the strips in our experimental data are predominantly horizontal or vertical, real-world HSIs have strips at various angles. In future research, we will work on solving more complex and more realistic noise, including oblique stripes.

REFERENCES

[1] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[2] J. He, Q. Yuan, J. Li, Y. Xiao, and L. Zhang, "A self-supervised remote sensing image fusion framework with dual-stage self-learning and spectral super-resolution injection," *ISPRS J. Photogramm. Remote Sens.*, vol. 204, pp. 131–144, Oct. 2023.

[3] R. Dian, A. Guo, and S. Li, "Zero-shot hyperspectral sharpening," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12650–12666, Oct. 2023.

[4] J. Wu, X. Su, Q. Yuan, H. Shen, and L. Zhang, "Multivehicle object tracking in satellite video enhanced by slow features and motion features," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5616426.

[5] R. Näsi et al., "Using UAV-based photogrammetry and hyperspectral imaging for mapping bark beetle damage at tree-level," *Remote Sens.*, vol. 7, no. 11, pp. 15467–15493, Nov. 2015.

[6] J. He et al., "Spectral super-resolution meets deep learning: Achievements and challenges," *Inf. Fusion*, vol. 97, Sep. 2023, Art. no. 101812.

[7] R. Dian, T. Shan, W. He, and H. Liu, "Spectral super-resolution via model-guided cross-fusion network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 27, 2023, doi: 10.1109/TNNLS.2023.3238506.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[9] R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," 2018, *arXiv:1810.03982*.

[10] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jul. 2005, pp. 60–65.

[11] Y. Qian and M. Ye, "Hyperspectral imagery restoration using nonlocal spectral–spatial structured sparse representation with noise estimation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 499–515, Apr. 2013.

[12] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

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

[14] T. Lu, S. Li, L. Fang, Y. Ma, and J. A. Benediktsson, "Spectral–spatial adaptive sparse representation for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 373–385, Jan. 2016.

[15] J. Li, Q. Yuan, H. Shen, and L. Zhang, "Noise removal from hyperspectral image with joint spectral–spatial distributed sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5425–5439, Sep. 2016.

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

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

[18] Y. Chen, W. He, N. Yokoya, and T.-Z. Huang, "Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3556–3570, Aug. 2020.

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

[20] W. He, Q. Yao, C. Li, N. Yokoya, and Q. Zhao, "Non-local meets global: An integrated paradigm for hyperspectral denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6861–6870.

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

[22] S. Takeyama, S. Ono, and I. Kumazawa, "A constrained convex optimization approach to hyperspectral image restoration with hybrid spatio-spectral regularization," *Remote Sens.*, vol. 12, no. 21, p. 3541, Oct. 2020.

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

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

[25] R. Pande-Chhetri and A. Abd-Elrahman, "De-striping hyperspectral imagery using wavelet transform and adaptive frequency domain filtering," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 5, pp. 620–636, Sep. 2011.

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

[27] Y. Zhao, D. Zhai, J. Jiang, and X. Liu, "ADRN: Attention-based deep residual network for hyperspectral image denoising," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2668–2672.

[28] W. Liu and J. Lee, "A 3-D atrous convolution neural network for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5701–5715, Aug. 2019.

[29] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2808–2817.

[30] F. Wang, J. Li, Q. Yuan, and L. Zhang, "Local–global feature-aware transformer based residual network for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5546119.

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

[32] H. Chen, G. Yang, and H. Zhang, "Hider: A hyperspectral image denoising transformer with spatial–spectral constraints for hybrid noise removal," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 31, 2022, doi: 10.1109/TNNLS.2022.3215751.

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

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

[35] S. Gu, Z. Yan, X. Liu, and L. Zhang, "Weighted nuclear norm minimization with application to image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1506–1521, Jun. 2014.

[36] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, Jan. 2013.

[37] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sep. 1966.

[38] R. A. Harshman and M. E. Lundy, "PARAFAC: Parallel factor analysis," *Comput. Statist. Data Anal.*, vol. 18, no. 1, pp. 39–72, Aug. 1994.

[39] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5706–5721, Mar. 2021.

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

[41] J. Lin, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, and L. Zhuang, "A tensor subspace representation-based method for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7739–7757, Sep. 2021.

[42] H. Zhang, H. Chen, G. Yang, and L. Zhang, "LR-net: Low-rank spatial–spectral network for hyperspectral image denoising," *IEEE Trans. Image Process.*, vol. 30, pp. 8743–8758, 2021.

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

[44] H. Zeng et al., "Degradation-noise-aware deep unfolding transformer for hyperspectral image denoising," 2023, *arXiv:2305.04047*.

[45] L. Zhuang, M. K. Ng, L. Gao, J. Michalski, and Z. Wang, "Eigen-image2Eigenimage (E2E): A self-supervised deep learning network for hyperspectral image denoising," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 19, 2023, doi: 10.1109/TNNLS.2023.3293328.

[46] F. Xiong, J. Zhou, S. Tao, J. Lu, J. Zhou, and Y. Qian, "SMDS-net: Model guided spectral–spatial network for hyperspectral image denoising," *IEEE Trans. Image Process.*, vol. 31, pp. 5469–5483, 2022.

[47] 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.

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

[49] H. Aetesam, S. K. Maji, and H. Yahia, "Bayesian approach in a learning-based hyperspectral image denoising framework," *IEEE Access*, vol. 9, pp. 169335–169347, 2021.

[50] H.-X. Dou, X.-S. Lu, C. Wang, H.-Z. Shen, Y.-W. Zhuo, and L.-J. Deng, "PatchMask: A data augmentation strategy with Gaussian noise in hyperspectral images," *Remote Sens.*, vol. 14, no. 24, p. 6308, Dec. 2022.

[51] J. He, Q. Yuan, J. Li, and L. Zhang, "A knowledge optimization-driven network with normalizer-free group ResNet prior for remote sensing image pan-sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5410716.

[52] J. He, J. Li, Q. Yuan, H. Shen, and L. Zhang, "Spectral response function-guided deep optimization-driven network for spectral super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4213–4227, Sep. 2022.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[54] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[55] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[56] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.

[57] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[58] L. Lin, Y. Shen, J. Wu, and F. Nan, "CAFE: A cross-attention based adaptive weighting fusion network for MODIS and Landsat spatiotemporal fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 5001605.

[59] Y. Jing, L. Lin, X. Li, T. Li, and H. Shen, "An attention mechanism based convolutional network for satellite precipitation downscaling over China," *J. Hydrol.*, vol. 613, Oct. 2022, Art. no. 128388.

[60] Y. Xiao, Q. Yuan, K. Jiang, J. He, Y. Wang, and L. Zhang, "From degrade to upgrade: Learning a self-supervised degradation guided adaptive network for blind remote sensing image super-resolution," *Inf. Fusion*, vol. 96, pp. 297–311, Aug. 2023.

[61] Y. Xiao, X. Su, Q. Yuan, D. Liu, H. Shen, and L. Zhang, "Satellite video super-resolution via multiscale deformable convolution alignment and temporal grouping projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5610819.

[62] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

[63] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11976–11986.

Yajie Li received the B.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2021, where she is currently pursuing the M.S. degree with the School of Geodesy and Geomatics.

Her research interests include hyperspectral denoising, quality improvement, remote sensing image processing, and deep learning.

Jie Li (Member, IEEE) received the B.S. degree in sciences and techniques of remote sensing and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2011 and 2016, respectively.

He is currently an Associate Professor with the School of Geodesy and Geomatics, Wuhan University. His research interests include image quality improvement, image super-resolution reconstruction, data fusion, remote sensing image processing, sparse representation, and deep learning.

Jiang He (Graduate Student Member, IEEE) received the B.S. degree in remote sensing science and technology from the Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Geodesy and Geomatics, Wuhan University, Wuhan, China.

His research interests include hyperspectral super-resolution, image fusion, quality improvement, remote sensing image processing, and deep learning.

Xinxin Liu (Member, IEEE) received the B.S. degree in geographic information system and the Ph.D. degree in cartography and geographic information system from Wuhan University, Wuhan, China, in 2013 and 2018, respectively.

In July 2018, she joined the College of Electrical and Information Engineering, Hunan University, Changsha, China, where she is currently an Associate Professor. Her research interests include image quality improvement, remote sensing image processing, and remote sensing mapping and application.

Qiangqiang Yuan (Member, IEEE) received the B.S. degree in surveying and mapping engineering and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2006 and 2012, respectively.

In 2012, he joined the School of Geodesy and Geomatics, Wuhan University, where he is currently a Professor. He has published more than 90 research articles, including more than 70 peer-reviewed articles in international journals, such as the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. His research interests include image reconstruction, remote sensing image processing and application, and data fusion.

Dr. Yuan was a recipient of the Top-Ten Academic Star of Wuhan University in 2011 and the Youth Talent Support Program of China in 2019. In 2014, he received the Hong Kong Scholar Award from the Society of Hong Kong Scholars and the China National Postdoctoral Council. He is on the Editor Board of nine international journals and has frequently served as a referee for more than 50 international journals for remote sensing and image processing.