Deep Unfolding Tensor Rank Minimization with Generalized Detail Injection for Pansharpening

Truong Thanh Nhat Mai, *Student Member, IEEE,* Edmund Y. Lam, *Fellow, IEEE,* and Chul Lee, *Member, IEEE*

***Abstract*—Pansharpening aims to generate a high-resolution multispectral (HRMS) image by merging a low-resolution multispectral (LRMS) image with a high-resolution panchromatic (PAN) image. While traditional model-based pansharpening algorithms have strong theoretical foundations, their performance and generalizability are limited by handcrafted formulations. In contrast, recent deep learning approaches outperform model-based algorithms but do not effectively consider the physical properties of multispectral (MS) images, such as their spatial and spectral dependencies. These physical properties facilitate the exploitation of the actual imaging process, leading to enhanced spatial and spectral fidelities. In this work, we propose a deep unfolded tensor rank minimization framework with generalized detail injection for pansharpening to overcome the weaknesses of both model- and learning-based approaches while leveraging their advantages. Specifically, we first formulate the pansharpening task as a tensor rank minimization problem to exploit the low-rankness of MS images, providing a robust theoretical foundation on the physical properties of MS data. We also develop a generalized detail injection component, which effectively exploits the information in the PAN images, and incorporate it into the optimization to improve generalizability and representation capability. Then, we define a data-driven regularizer to compensate for modeling inaccuracies in the low-rank model and solve the optimization problem using an iterative technique. Finally, the iterative algorithm is unfolded into a multistage deep network, in which the optimization variables are solved by closed-form solutions and a data-driven regularizer in each stage. Experimental results on various MS image datasets demonstrate that the proposed algorithm achieves better pansharpening performance and interpretability than state-of-the-art algorithms.**

***Index Terms*—Tensor rank minimization, deep unfolding, model-based deep learning, pansharpening.**

I. INTRODUCTION

REMOTE sensing applications, such as change detection, target recognition, and environmental monitoring, demand high-resolution multispectral (HRMS) images with rich details to achieve superior performance [1]. However, remote sensing satellites can capture only low-resolution multispectral (LRMS) and high-resolution panchromatic (PAN) images via two different sensors because of the physical limitations of sensors, such as diffraction limits and sensor artifacts [2]. A PAN image has high spatial resolution with great detail but lacks spectral information, whereas the LRMS image provides rich spectral information but sacrifices spatial resolution. To overcome the limitations of satellite sensors in capturing HRMS images, pansharpening techniques that estimate HRMS images from pairs of LRMS and PAN images have been developed. Specifically, a pansharpening algorithm combines the respective spectral and spatial information from the LRMS image and its corresponding PAN image to generate an HRMS image with rich details. Research in pansharpening has attracted significant attention because of its practical importance. Thus, various approaches have been developed, which can be categorized into four groups based on how the information from the PAN images is exploited [1]: component substitution (CS)-, multiresolution analysis (MRA)-, variational optimization (VO)-, and deep learning (DL)-based algorithms.

The CS-based algorithms [3]–[6] separate the LRMS image into the spatial and spectral components and substitute the spatial component with that of the PAN image. On the other hand, the MRA-based algorithms [7]–[12] extract spatial details from the PAN image and inject them into the LRMS image. CS-based algorithms exhibit superior spatial fidelity at the cost of spectral distortion, whereas MRA-based algorithms excel at spectral fidelity but suffer from spatial distortion [13]. The VO-based algorithms [14]–[24] formulate the pansharpening task as regularized optimization problems by modeling the physical properties of multispectral (MS) images. However, their performance relies on handcrafted regularization and prior terms, which may not accurately represent the characteristics of MS images.

Recently, DL-based algorithms employing convolutional neural networks (CNNs) have been actively developed for pansharpening and have shown superior performance [13], [25]–[34]. However, as DL-based algorithms focus on manipulating the deep features, they may yield spatial or spectral distortions during the decoding of these features back to HRMS images [35]. Moreover, their network architectures are frequently empirically designed without domain-specific knowledge, limiting interpretability [36]. In addition, model-based DL techniques have recently been employed in pansharpening [35]–[42] to enhance interpretability by exploiting domain-specific knowledge, *i.e.*, the physical properties of MS images, with learnable components. Nevertheless, they inherit the limitations of their generic degradation models, which do not effectively represent the properties of MS and PAN images and are prone to modeling inaccuracies. Moreover, although different sensors capture PAN images with diverse characteristics, existing model-based DL approaches [35]–[42] process the PAN images using a fixed model without considering such differences, thereby degrading quality.

In this work, to address the aforementioned limitations of

T. T. N. Mai and C. Lee are with the Department of Multimedia Engineering, Dongguk University, Seoul 04620, South Korea.

E. Y. Lam is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong.

conventional algorithms, we propose a deep unfolding tensor rank minimization model with generalized detail injection, called UTeRM (**U**nfolding **Te**nsor **R**ank **M**inimization). Tensor rank minimization ensures high spatial and spectral fidelity by faithfully restoring multidimensional dependency, whereas the generalized detail injection increases the generalizability by effectively exploiting the characteristics of PAN images and their correlations with those of MS images. Therefore, the proposed UTeRM can overcome the weaknesses of both learning- and model-based approaches while leveraging their advantages. To the best of our knowledge, this is the first model-based DL approach with a generalized detail injection component, while existing model-based DL algorithms use a fixed model or formulation, thereby limiting their adaptivity. In addition, the proposed UTeRM is developed based on a more robust low-rank tensor model than existing model-based DL approaches [35]–[42] to mitigate their limitations.

Specifically, we first formulate the pansharpening task as a tensor rank minimization problem to enforce a low rank on the resulting HRMS images. Then, we develop a generalized detail injection term, which extends the generalizability and representation capability of the proposed algorithm, and incorporate it into the optimization problem. Next, we design a CNN-based learnable regularizer instead of explicit formulations to compensate for the potential modeling inaccuracy of the low-rank model. This data-driven regularizer learns the visual features by training the LRMS and PAN pairs, thereby producing accurate regularization for the model. Then, we iteratively solve the tensor rank minimization problem and establish a multistage deep unfolded network from the iterative solution. In each stage, the optimization variables are updated by closed-form solutions, and the regularizer is updated using a CNN-based solution. The proposed algorithm achieves state-of-the-art performance and better interpretability by incorporating a generalized detail injection component and learnable regularizer into the tensor rank minimization framework.

In summary, we make the following contributions.

- We formulate the pansharpening task as a tensor rank minimization problem with a generalized detail injection component, ensuring high spectral and spatial fidelity and enhancing generalizability.
- We develop an iterative algorithm to solve the optimization problem with various implementations of the generalized detail injection.
- We establish an interpretable multistage deep network for pansharpening by unfolding the iterative algorithm; each stage of the deep network corresponds to an iteration.
- We experimentally demonstrate that the proposed algorithm significantly outperforms state-of-the-art algorithms on several datasets, generating HRMS images with high spectral and spatial fidelities; we also analyze its behaviors and characteristics.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III describes the proposed UTeRM for pansharpening. Then, Section IV discusses the experimental results and ablation studies. Finally, Section V provides the concluding remarks.

II. Related Work

A. Model-Based Pansharpening

Model-based algorithms are categorized into CS-, MRA-, and VO-based algorithms. The CS-based algorithms separate the LRMS image into the spatial and spectral components by adopting spectral transformations, including principal component analysis [3], Gram–Schmidt analysis [4], partial replacement adaptive CS [5], and band-dependent spatial detail [6]. Then, they substitute the spatial component with that from the PAN image for HRMS image generation. The MRA-based algorithms employ multiresolution transformations, such as the Laplacian pyramid [7], low-pass filter [8], wavelet transform [9], [10], morphological operators [11], and modulation transfer function-based Gaussian filters [12], to extract spatial details from the PAN image, and generate the HRMS image by injecting the spatial details into the upsampled LRMS image. These CS- and MRA-based algorithms have opposing strengths and weaknesses. For example, CS-based algorithms exhibit good spatial fidelity but high spectral distortions, whereas MRA-based algorithms yield superior spectral performance but suffer from spatial distortions.

The VO-based algorithms formulate and solve regularized inverse problems that model the characteristics and relationships of LRMS, PAN, and HRMS images. Several regularization models, such as the total variation [14], hyper-Laplacian prior [15], adaptive maximum a posteriori [16], sparse priors [17], and DL-based priors [18], have been developed to capture the characteristics of satellite images. In addition, several VO-based algorithms exploit the low-rankness of MS images, assuming linear dependency between spectral channels [19]–[23]. More recently, a low-rank tensor completion (LRTC) model was employed to reflect the low-rankness of MS images more effectively by considering multidimensional dependency [24]. VO-based algorithms can achieve high spectral and spatial fidelity owing to their sophisticated optimization models; however, they are generally computationally complex, and their hyperparameters require manual adjustments. Moreover, the performance of these algorithms depends on handcrafted regularizers and priors, limiting their generalizability.

B. Deep Learning-Based Pansharpening

Early DL-based algorithms employed CNNs as end-to-end mappings to extract and merge the spectral and spatial features directly from the input LRMS and PAN images and then generate HRMS images [26]–[28]. More recent DL-based algorithms aim to achieve higher performance by extracting more advanced and complex features from the LRMS and PAN images by employing sophisticated architectures and learning strategies, such as bidirectional pyramid networks [25], detail injection-based networks [13], generative adversarial networks [29], feature modulation-based networks [30], attention-based networks [43], [44], unsupervised learning [31], dual-domain learning [32], and transformers [33], [34]. However, DL-based algorithms do not effectively reflect the physical properties of the PAN and MS images, resulting in HRMS images with spectral and spatial distortions. Moreover, interpreting and explaining the behavior of DL-based algorithms is difficult because of their empirically designed architectures.

Recently, algorithm unfolding or unrolling [45] has been employed for pansharpening to exploit the physical properties of MS images more effectively [35]–[42], constructing model-based deep networks. In this approach, the pansharpening task is first formulated as an optimization problem with learnable components, such as filter kernels [35]–[38] or degradation operators [39]–[42]. Then, the optimization problem is solved iteratively, and the iterations are unfolded into a series of network layers, where the learnable components are adjusted during training via backpropagation. Model-based deep networks inherit the robust theoretical foundation of model-based algorithms while retaining data-driven learning capabilities, achieving significant success. However, they inherit the limitations of generic models in representing the physical properties of the PAN and MS images and employ a fixed formulation to exploit spatial information from the PAN images that may not effectively represent the diverse characteristics of these images, resulting in low-fidelity HRMS images.

Despite superior results, existing DL-based approaches face certain limitations, *e.g.*, overlooking the physical properties of PAN and MS images and ineffectively exploiting spatial information of diverse PAN images. In this work, we address these limitations by developing a tensor rank minimization approach with effective generalized detail injection.

III. Proposed Algorithm—UTeRM

A. Notations and Definitions

In this paper, tensors are denoted by boldfaced calligraphic Latin or Greek letters, *e.g.*, $\boldsymbol{\mathcal{A}}$ or $\boldsymbol{\Lambda}$; matrices by boldfaced uppercase Latin letters, *e.g.*, $\mathbf{A}$; and scalars by italicized Latin or Greek letters, *e.g.*, a, N, and α, except for the letters f and $\mathcal{F}$, which are exclusively used to denote functions and tensor-based operators, respectively. Given a matrix $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$, the conjugate transpose and pseudo-inverse of $\mathbf{A}$ are denoted as $\mathbf{A}^H$ and $\mathbf{A}^\dagger$, respectively. Given a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its i-th frontal slice, *i.e.*, channel or band, is denoted by $\boldsymbol{\mathcal{A}}^{(i)}$.

Given two tensors $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor inner product $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle$ is defined as

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle \triangleq \sum_i \langle \boldsymbol{\mathcal{A}}^{(i)}, \boldsymbol{\mathcal{B}}^{(i)} \rangle, \tag{1}$$

where $\langle \boldsymbol{\mathcal{A}}^{(i)}, \boldsymbol{\mathcal{B}}^{(i)} \rangle \triangleq \mathrm{tr}\big(\boldsymbol{\mathcal{A}}^{(i)^T} \boldsymbol{\mathcal{B}}^{(i)}\big)$ represents the matrix inner product. The tensor-tensor product (t-product) [46] $\boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}}$ is defined as

$$\boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}} = \texttt{fold}(\texttt{bcirc}(\boldsymbol{\mathcal{A}})\texttt{unfold}(\boldsymbol{\mathcal{B}})), \tag{2}$$

where $\texttt{bcirc}(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ indicates the block circulant matrix of $\boldsymbol{\mathcal{A}}$, *i.e.*,

$$\texttt{bcirc}(\boldsymbol{\mathcal{A}}) = \begin{bmatrix} \boldsymbol{\mathcal{A}}^{(1)} & \boldsymbol{\mathcal{A}}^{(n_3)} & \cdots & \boldsymbol{\mathcal{A}}^{(2)} \\ \boldsymbol{\mathcal{A}}^{(2)} & \boldsymbol{\mathcal{A}}^{(1)} & \cdots & \boldsymbol{\mathcal{A}}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{A}}^{(n_3)} & \boldsymbol{\mathcal{A}}^{(n_3-1)} & \cdots & \boldsymbol{\mathcal{A}}^{(1)} \end{bmatrix}, \tag{3}$$

and $\texttt{unfold}(\boldsymbol{\mathcal{B}}) = \big[\boldsymbol{\mathcal{B}}^{(1)}; \boldsymbol{\mathcal{B}}^{(2)}; \ldots; \boldsymbol{\mathcal{B}}^{(n_3)}\big] \in \mathbb{R}^{n_1 n_3 \times n_2}$, and $\texttt{fold}$ is the inverse of $\texttt{unfold}$, *i.e.*, $\texttt{fold}(\texttt{unfold}(\boldsymbol{\mathcal{B}})) = \boldsymbol{\mathcal{B}}$.

B. Problem Formulation

We define an LRMS image as $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{h \times w \times C}$, where h, w, and C denote its height, width, and the number of spectral channels, respectively, and a PAN image as $\mathbf{P} \in \mathbb{R}^{H \times W}$, where $H = rh$ and $W = rw$ denote its height and width, respectively, and $r > 1$ is the resolution ratio. In [24], it was experimentally shown that HRMS images exhibit the low tubal-rank property. Therefore, given $\boldsymbol{\mathcal{M}}$ and $\mathbf{P}$, the pansharpening task can be formulated as an LRTC problem. However, as previously discussed, LRTC models require accurately determined sets of reliable pixels, which is challenging because of the different sizes of the LRMS and PAN images. To address this limitation, we formulate the pansharpening task as a more general tensor rank minimization problem that does not require sets of reliable pixels. Furthermore, we incorporate the spatial details of $\mathbf{P}$ into the optimization model by employing a detail injection component. Then, the optimization problem for pansharpening via tensor rank minimization can be expressed as

$$\underset{\boldsymbol{\mathcal{X}}}{\text{minimize}} \ \mathrm{rank}(\boldsymbol{\mathcal{X}}) + \lambda f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}), \tag{4}$$

where $\mathrm{rank}(\boldsymbol{\mathcal{X}})$ enforces low-rankness on the desired HRMS image $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{H \times W \times C}$, $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$ represents the detail injection component that incorporates spatial information from $\mathbf{P}$ into $\boldsymbol{\mathcal{X}}$, and λ balances the relative importance of the two terms. Next, we present a detailed formulation of $\mathrm{rank}(\boldsymbol{\mathcal{X}})$ and $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$.

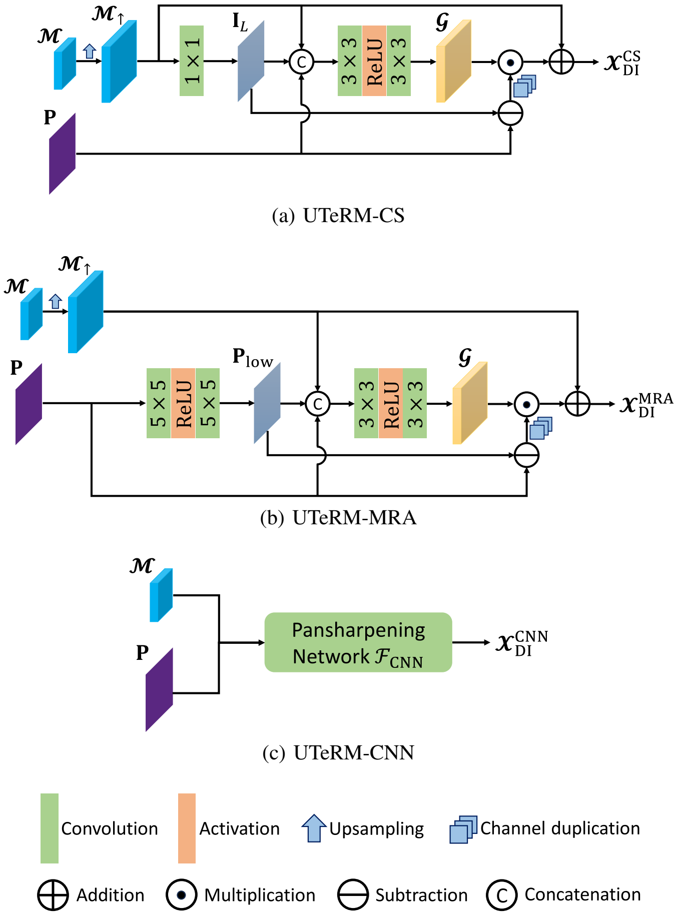
The main challenge in tensor rank-related problems is defining a tensor rank, which is difficult due to the complex multidimensional relationships between the elements of tensors [47]. Consequently, various definitions of tensor rank have been proposed from different perspectives. In this work, we employ a tubal rank via factorization [48] to preserve multidimensional information effectively while achieving computational efficiency. Specifically, the tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{H \times W \times C}$ with a tubal rank r can be factorized into two smaller tensors $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}$, where $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{H \times r \times C}$ and $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{r \times W \times C}$. By enforcing a low tubal rank via tensor factorization, the tensor rank minimization problem in (4) can be rewritten as

$$\underset{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}}{\text{minimize}} \ \frac{1}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}\|_F^2 + \lambda f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}). \tag{5}$$

In addition, although HRMS images are often low rank [24], the tensor rank minimization model in (5) may not accurately represent all the real-world scenarios. For example, the predefined tubal rank r does not match the true tubal rank, resulting in low-fidelity reconstruction results. To mitigate such modeling inaccuracies, we develop an additional implicit regularizer $f_{\text{reg}}(\cdot)$ for $\boldsymbol{\mathcal{X}}$. Then, the complete optimization problem for pansharpening can be rewritten as

$$\underset{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}}{\text{minimize}} \ \frac{1}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}\|_F^2 + f_{\text{reg}}(\boldsymbol{\mathcal{X}}) + \lambda f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}). \tag{6}$$

In conventional optimization-based algorithms [14]–[24], including model-based DL algorithms [35]–[42], specific formulations for $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$ in (6) were proposed to exploit



(a) UTeRM-CS

(b) UTeRM-MRA

(c) UTeRM-CNN

Fig. 1. Architectures of (a) CS-, (b) MRA-, and (c) CNN-based detail injection modules. The operations are performed element-wise, except for concatenation, which is performed channel-wise.

the spatial information from the PAN images. However, different sensors collect information from different wavelength ranges, resulting in PAN images with diverse characteristics; therefore, relying on specific formulations may not effectively represent the spatial details of those images. To overcome this limitation, we propose an implicit formulation for $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$ as

$$f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}) = \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}_{\text{DI}}\|_F^2, \tag{7}$$

where $\boldsymbol{\mathcal{X}}_{\text{DI}}$ denotes the detail-injected HRMS image for $\boldsymbol{\mathcal{X}}$ computed from $\boldsymbol{\mathcal{M}}$ and $\mathbf{P}$. Since the implicit formulation in (7) can support a generalized design, it enables the flexible selection of any formulation or implementation for the detail injection component depending on the characteristics of the datasets. In this work, we present three implementations of the generalized detail injection component $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$—CS-, MRA-, and CNN-based formulations.

1) CS-based Detail Injection (UTeRM-CS): A CS-based algorithm constructs a desired HRMS image by combining the PAN image $\mathbf{P}$ and upsampled LRMS image $\boldsymbol{\mathcal{M}}_{\uparrow}$ substituted with the intensity component [49], *i.e.*,

$$\boldsymbol{\mathcal{X}}^{(i)} = \boldsymbol{\mathcal{M}}_{\uparrow}^{(i)} + g_i \times (\mathbf{P} - \mathbf{I}_L), \tag{8}$$

where g_i denotes the global injection coefficient, and $\mathbf{I}_L$ is the intensity component computed as the weighted sum of the channels of $\boldsymbol{\mathcal{M}}_{\uparrow}$. We extend the capability of the CS-based formulation by computing $\mathbf{I}_L$ via a 1×1 convolution, equivalent to performing a weighted sum with learnable weights, which is expressed as

$$\mathbf{I}_L = \text{Conv}_{1\times1}(\boldsymbol{\mathcal{M}}_{\uparrow}). \tag{9}$$

In addition, we extend the channel-wise coefficient g_i to the pixel-wise weight map $\boldsymbol{\mathcal{G}}$ for a more effective detail injection. To this end, we compute $\boldsymbol{\mathcal{G}}$ using the convolutional layers as

$$\boldsymbol{\mathcal{G}} = \text{Conv}_{3\times3}(\text{ReLU}(\text{Conv}_{3\times3}([\boldsymbol{\mathcal{M}}_{\uparrow}, \mathbf{P}, \mathbf{I}_L]))), \tag{10}$$

where $[\boldsymbol{\mathcal{M}}_{\uparrow}, \mathbf{P}, \mathbf{I}_L]$ denotes the concatenation along the channel dimension. By substituting $\mathbf{I}_L$ in (9) and $\boldsymbol{\mathcal{G}}$ in (10) into (8), the CS-based detail injection can be reformulated as

$$\boldsymbol{\mathcal{X}}_{\text{DI}}^{\text{CS}} = \boldsymbol{\mathcal{M}}_{\uparrow} + \boldsymbol{\mathcal{G}} \odot (\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{I}}_L), \tag{11}$$

where $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{I}}_L$ are the tensor version of $\mathbf{P}$ and $\mathbf{I}_L$, respectively, obtained by replicating the matrices C times along the channel dimension, and $\odot$ denotes element-wise multiplication. The overall architecture is illustrated in Fig. 1(a). Then, by substituting (11) into (7), the detail injection $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$ in (7) can be rewritten as

$$f_{\text{detail}}^{\text{CS}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}) = \frac{1}{2}\|\boldsymbol{\mathcal{X}} - (\boldsymbol{\mathcal{M}}_{\uparrow} + \boldsymbol{\mathcal{G}} \odot (\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{I}}_L))\|_F^2. \tag{12}$$

2) MRA-based Detail Injection (UTeRM-MRA): An MRA-based algorithm constructs an HRMS image as a combination of an upsampled LRMS image and high-frequency details from a PAN image [49] as

$$\boldsymbol{\mathcal{X}}^{(i)} = \boldsymbol{\mathcal{M}}_{\uparrow}^{(i)} + g_i \times (\mathbf{P} - \mathbf{P}_{\text{low}}), \tag{13}$$

where $\mathbf{P}_{\text{low}}$ is the low-pass filtered version of $\mathbf{P}$. Similar to CS-based detail injection, the weights of the low-pass filter are learned via convolutional layers. Specifically, a low-pass filter is implemented using 5×5 convolutional kernels as

$$\mathbf{P}_{\text{low}} = \text{Conv}_{5\times5}(\text{ReLU}(\text{Conv}_{5\times5}(\mathbf{P}))). \tag{14}$$

The pixel-wise weight map $\boldsymbol{\mathcal{G}}$ is computed similarly, as defined in (10), *i.e.*,

$$\boldsymbol{\mathcal{G}} = \text{Conv}_{3\times3}(\text{ReLU}(\text{Conv}_{3\times3}([\boldsymbol{\mathcal{M}}_{\uparrow}, \mathbf{P}, \mathbf{P}_{\text{low}}]))). \tag{15}$$

By substituting $\mathbf{P}_{\text{low}}$ in (14) and $\boldsymbol{\mathcal{G}}$ in (15) into (13), the MRA-based detail injection can be reformulated as

$$\boldsymbol{\mathcal{X}}_{\text{DI}}^{\text{MRA}} = \boldsymbol{\mathcal{M}}_{\uparrow} + \boldsymbol{\mathcal{G}} \odot (\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{P}}_{\text{low}}), \tag{16}$$

where $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{P}}_{\text{low}}$ are the tensor versions of $\mathbf{P}$ and $\mathbf{P}_{\text{low}}$, respectively. Its architecture is shown in Fig. 1(b). Then, the detail injection $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$ in (7) is reformulated for the MRA-based approach as

$$f_{\text{detail}}^{\text{MRA}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}) = \frac{1}{2}\|\boldsymbol{\mathcal{X}} - (\boldsymbol{\mathcal{M}}_{\uparrow} + \boldsymbol{\mathcal{G}} \odot (\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{P}}_{\text{low}}))\|_F^2. \tag{17}$$

3) CNN-based Detail Injection (UTeRM-CNN): Finally, we formulate a CNN-based detail injection component using existing pansharpening deep networks. Specifically, let $\mathcal{F}_{\text{CNN}}(\boldsymbol{\mathcal{M}}, \mathbf{P})$ be the output of a conventional pansharpening network. Then, the CNN-based detail injection can be formulated as

$$\boldsymbol{\mathcal{X}}_{\text{DI}}^{\text{CNN}} = \mathcal{F}_{\text{CNN}}(\boldsymbol{\mathcal{M}}, \mathbf{P}). \tag{18}$$

Thus, the detail injection $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$ in (7) becomes

$$f_{\text{detail}}^{\text{CNN}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}) = \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \mathcal{F}_{\text{CNN}}(\boldsymbol{\mathcal{M}}, \mathbf{P})\|_F^2. \tag{19}$$

Its architecture is illustrated in Fig. 1(c). The formulation of the UTeRM-CNN supports both pretraining and end-to-end training for $\mathcal{F}_{\text{CNN}}$, *i.e.*, $\mathcal{F}_{\text{CNN}}$ can be pretrained and then integrated into (26) or trained from scratch with the proposed algorithm. In this work, we employ FusionNet [13] for $\mathcal{F}_{\text{CNN}}$ due to its simplicity.

C. Solution to the Optimization

We solve the optimization problem in (6) iteratively by employing the augmented Lagrange multiplier method [50]. To this end, we first reformulate the optimization in (6) for variable splitting using the auxiliary variable $\boldsymbol{\mathcal{V}}$ as

$$\begin{aligned} \underset{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{V}}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}\|_F^2 + f_{\text{reg}}(\boldsymbol{\mathcal{V}}) + \lambda f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}), \\ \text{subject to} \quad & \boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{X}}. \end{aligned} \tag{20}$$

Then, the augmented Lagrangian function for (20) can be defined as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{V}}, \mathbf{\Lambda}) = {} & \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}\|_F^2 + f_{\text{reg}}(\boldsymbol{\mathcal{V}}) \\ & + \frac{\alpha}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}}\|_F^2 + \langle \mathbf{\Lambda}, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}} \rangle + \lambda f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}), \end{aligned} \tag{21}$$

where $\alpha > 0$ is the penalty parameter, and $\mathbf{\Lambda} \in \mathbb{R}^{H \times W \times C}$ is the Lagrange multiplier tensor.

The optimal solutions to the problem in (20) are obtained by minimizing the augmented Lagrangian function $\mathcal{L}$ in (21), *i.e.*,

$$(\boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{L}}^*, \boldsymbol{\mathcal{R}}^*, \boldsymbol{\mathcal{V}}^*) = \arg\min_{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{V}}} \mathcal{L}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{V}}, \mathbf{\Lambda}). \tag{22}$$

To this end, we employ the alternating direction method of multipliers [51] because directly solving the joint optimization problem in (22) is intractable. Specifically, we split the optimization in (22) into subproblems over the individual variables $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{L}}$, $\boldsymbol{\mathcal{R}}$, and $\boldsymbol{\mathcal{V}}$, and the multiplier $\mathbf{\Lambda}$, and then solve the subproblems sequentially and iteratively. The following section details how each subproblem is solved at the k-th iteration.

$\boldsymbol{\mathcal{X}}$-subproblem: Given $\boldsymbol{\mathcal{L}}_k$, $\boldsymbol{\mathcal{R}}_k$, $\boldsymbol{\mathcal{V}}_k$, and $\mathbf{\Lambda}_k$ from the previous $(k-1)$-th iteration, we first update $\boldsymbol{\mathcal{X}}$ as

$$\begin{aligned} \boldsymbol{\mathcal{X}}_{k+1} &= \arg\min_{\boldsymbol{\mathcal{X}}} \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}}_k * \boldsymbol{\mathcal{R}}_k\|_F^2 + \frac{\alpha_k}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}}_k\|_F^2 \\ &\qquad + \langle \mathbf{\Lambda}_k, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}}_k \rangle + \lambda_k f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P}) \\ &= \arg\min_{\boldsymbol{\mathcal{X}}} \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}}_k * \boldsymbol{\mathcal{R}}_k\|_F^2 + \frac{\alpha_k}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}}_k\|_F^2 \\ &\qquad + \langle \mathbf{\Lambda}_k, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}}_k \rangle + \frac{\lambda_k}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}_{\text{DI}}\|_F^2 \\ &= \frac{\boldsymbol{\mathcal{L}}_k * \boldsymbol{\mathcal{R}}_k + \lambda_k \boldsymbol{\mathcal{X}}_{\text{DI}} + \alpha_k \boldsymbol{\mathcal{V}}_k - \mathbf{\Lambda}_k}{(1 + \lambda_k + \alpha_k)\mathbf{1}}, \end{aligned} \tag{23}$$

where $\mathbf{1}$ is an all-ones tensor, and the division is performed element-wise.

We presented three implementations of the detail injection component $\boldsymbol{\mathcal{X}}_{\text{DI}}$ in Section III-B. Therefore, we obtain three closed-form solutions for each detail injection component. Specifically, by substituting (11), (16), and (18) into (23) for $\boldsymbol{\mathcal{X}}_{\text{DI}}$, we can readily obtain the respective closed-form solutions, which are given below.

1) UTeRM-CS:

$$\boldsymbol{\mathcal{X}}_{k+1} = \frac{\boldsymbol{\mathcal{L}}_k * \boldsymbol{\mathcal{R}}_k + \lambda_k(\boldsymbol{\mathcal{M}}_{\uparrow} + \boldsymbol{\mathcal{G}} \odot (\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{I}}_L)) + \alpha_k \boldsymbol{\mathcal{V}}_k - \mathbf{\Lambda}_k}{(1 + \lambda_k + \alpha_k)\mathbf{1}}. \tag{24}$$

2) UTeRM-MRA:

$$\boldsymbol{\mathcal{X}}_{k+1} = \frac{\boldsymbol{\mathcal{L}}_k * \boldsymbol{\mathcal{R}}_k + \lambda_k(\boldsymbol{\mathcal{M}}_{\uparrow} + \boldsymbol{\mathcal{G}} \odot (\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{P}}_{\text{low}})) + \alpha_k \boldsymbol{\mathcal{V}}_k - \mathbf{\Lambda}_k}{(1 + \lambda_k + \alpha_k)\mathbf{1}}. \tag{25}$$

3) UTeRM-CNN:

$$\boldsymbol{\mathcal{X}}_{k+1} = \frac{\boldsymbol{\mathcal{L}}_k * \boldsymbol{\mathcal{R}}_k + \lambda_k \mathcal{F}_{\text{CNN}}(\boldsymbol{\mathcal{M}}, \mathbf{P}) + \alpha_k \boldsymbol{\mathcal{V}}_k - \mathbf{\Lambda}_k}{(1 + \lambda_k + \alpha_k)\mathbf{1}}. \tag{26}$$

$\boldsymbol{\mathcal{L}}$- and $\boldsymbol{\mathcal{R}}$-subproblems: Then, we update $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{R}}$ simultaneously by solving

$$(\boldsymbol{\mathcal{L}}_{k+1}, \boldsymbol{\mathcal{R}}_{k+1}) = \arg\min_{\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{R}}} \frac{1}{2}\|(\boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}})\|_F^2. \tag{27}$$

This subproblem can be solved in the Fourier domain to circumvent the complexity of the t-product [48]. Specifically, let $\widehat{\boldsymbol{\mathcal{X}}}_{k+1}$ denote the result of the Fourier transform $\mathcal{F}_{\text{FFT}}(\cdot)$ of $\boldsymbol{\mathcal{X}}_{k+1}$ along the third dimension, *i.e.*, $\widehat{\boldsymbol{\mathcal{X}}}_{k+1}(i,j,:) = \mathcal{F}_{\text{FFT}}\{\boldsymbol{\mathcal{X}}_{k+1}(i,j,:)\}$. Then, the closed-form solutions $\widehat{\boldsymbol{\mathcal{L}}}_{k+1}$ and $\widehat{\boldsymbol{\mathcal{R}}}_{k+1}$, *i.e.*, the Fourier coefficients for $\boldsymbol{\mathcal{L}}_{k+1}$ and $\boldsymbol{\mathcal{R}}_{k+1}$, respectively, can be computed frontal slice-wise as

$$\begin{aligned} \widehat{\boldsymbol{\mathcal{L}}}_{k+1}^{(i)} &= \arg\min_{\widehat{\boldsymbol{\mathcal{L}}}^{(i)}} \frac{1}{2C}\left\|\left(\widehat{\boldsymbol{\mathcal{L}}}^{(i)} * \widehat{\boldsymbol{\mathcal{R}}}_k^{(i)} - \widehat{\boldsymbol{\mathcal{X}}}_{k+1}^{(i)}\right)\right\|_F^2 \\ &= \widehat{\boldsymbol{\mathcal{X}}}_{k+1}^{(i)}\left(\widehat{\boldsymbol{\mathcal{R}}}_k^{(i)}\right)^H\left(\widehat{\boldsymbol{\mathcal{R}}}_k^{(i)}\left(\widehat{\boldsymbol{\mathcal{R}}}_k^{(i)}\right)^H\right)^{\dagger}, \end{aligned} \tag{28}$$

$$\begin{aligned} \widehat{\boldsymbol{\mathcal{R}}}_{k+1}^{(i)} &= \arg\min_{\widehat{\boldsymbol{\mathcal{R}}}^{(i)}} \frac{1}{2C}\left\|\left(\widehat{\boldsymbol{\mathcal{L}}}_k^{(i)} * \widehat{\boldsymbol{\mathcal{R}}}^{(i)} - \widehat{\boldsymbol{\mathcal{X}}}_{k+1}^{(i)}\right)\right\|_F^2 \\ &= \left(\left(\widehat{\boldsymbol{\mathcal{L}}}_k^{(i)}\right)^H \widehat{\boldsymbol{\mathcal{L}}}_k^{(i)}\right)^{\dagger}\left(\widehat{\boldsymbol{\mathcal{L}}}_k^{(i)}\right)^H \widehat{\boldsymbol{\mathcal{X}}}_{k+1}^{(i)}. \end{aligned} \tag{29}$$

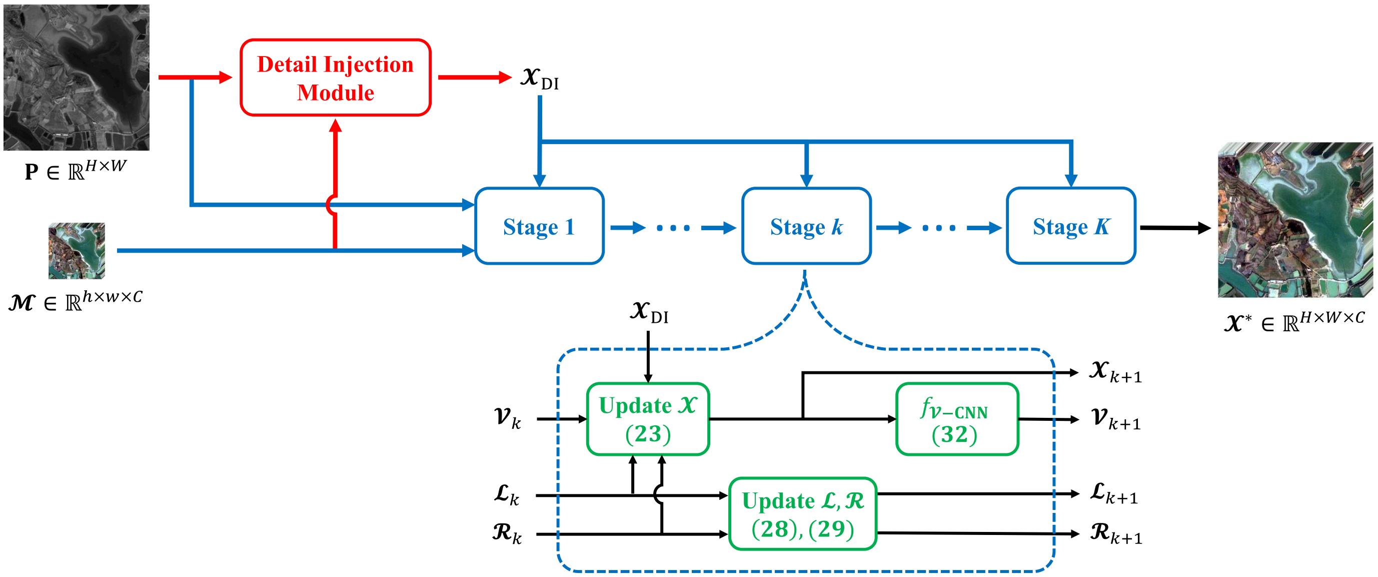

$\mathbf{P} \in \mathbb{R}^{H\times W}$
Detail Injection Module
$\boldsymbol{\mathcal{X}}_{\mathrm{DI}}$
$\boldsymbol{\mathcal{M}} \in \mathbb{R}^{h\times w\times C}$
Stage 1
Stage k
Stage K
$\boldsymbol{\mathcal{X}}^{*} \in \mathbb{R}^{H\times W\times C}$
$\boldsymbol{\mathcal{X}}_{\mathrm{DI}}$
$\boldsymbol{\mathcal{V}}_k$
Update $\boldsymbol{\mathcal{X}}$ (23)
$f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$ (32)
$\boldsymbol{\mathcal{L}}_k$
$\boldsymbol{\mathcal{R}}_k$
Update $\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}$ (28), (29)
$\boldsymbol{\mathcal{X}}_{k+1}$
$\boldsymbol{\mathcal{V}}_{k+1}$
$\boldsymbol{\mathcal{L}}_{k+1}$
$\boldsymbol{\mathcal{R}}_{k+1}$

Fig. 2. Architecture of the proposed network. The network comprises a generalized detail injection component and K stages of optimization. First, the detail injection component $\boldsymbol{\mathcal{X}}_{\mathrm{DI}}$ is obtained by exploiting the LRMS and PAN images. Then, $\boldsymbol{\mathcal{X}}_{\mathrm{DI}}$ and the LRMS and PAN images are input to a K-stage unfolded network, where the operations in each stage correspond to the closed-form solutions in an iteration of the iterative algorithm.

Finally, the closed-form solutions $\boldsymbol{\mathcal{L}}_{k+1}$ and $\boldsymbol{\mathcal{R}}_{k+1}$ can be obtained by performing an inverse Fourier transform along the third dimension as

$$\begin{aligned}\boldsymbol{\mathcal{L}}_{k+1}(i,j,:) &= \mathcal{F}_{\mathrm{FFT}}^{-1}\{\widehat{\boldsymbol{\mathcal{L}}}_{k+1}(i,j,:)\},\\ \boldsymbol{\mathcal{R}}_{k+1}(i,j,:) &= \mathcal{F}_{\mathrm{FFT}}^{-1}\{\widehat{\boldsymbol{\mathcal{R}}}_{k+1}(i,j,:)\}.\end{aligned} \tag{30}$$

$\boldsymbol{\mathcal{V}}$-subproblem: Next, we update the auxiliary variable $\boldsymbol{\mathcal{V}}$ as

$$\begin{aligned}\boldsymbol{\mathcal{V}}_{k+1} &= \arg\min_{\boldsymbol{\mathcal{V}}}\ f_{\mathrm{reg}}(\boldsymbol{\mathcal{V}}) + \frac{\alpha_k}{2}\|\boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{V}}\|_F^2\\ &\qquad + \langle \boldsymbol{\Lambda}_k, \boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{V}}\rangle\\ &= \arg\min_{\boldsymbol{\mathcal{V}}}\ f_{\mathrm{reg}}(\boldsymbol{\mathcal{V}}) + \frac{\alpha_k}{2}\|\boldsymbol{\mathcal{V}} - (\boldsymbol{\mathcal{X}}_{k+1} + \alpha_k^{-1}\boldsymbol{\Lambda}_k)\|_F^2\\ &= \mathrm{prox}_{f_{\mathrm{reg}}}(\boldsymbol{\mathcal{X}}_{k+1} + \alpha_k^{-1}\boldsymbol{\Lambda}_k),\end{aligned} \tag{31}$$

where $\mathrm{prox}_{f_{\mathrm{reg}}}(\cdot)$ denotes the proximal operator corresponding to the regularizer $f_{\mathrm{reg}}(\cdot)$. The regularizer imposes constraints on variable $\boldsymbol{\mathcal{X}}$, which is tailored to the specific characteristics of the desired HRMS image, to ensure high-fidelity results. However, in optimization-based approaches, regularizers are typically designed based on observations and practical experience and thus may not represent the wide variety of characteristics of real-world MS images [35]. To address this limitation of handcrafted regularizers, we establish a data-driven regularizer that can effectively represent a wide range of MS image characteristics. Specifically, the closed-form solution $\mathrm{prox}_{f_{\mathrm{reg}}}(\cdot)$ is computed using a CNN that learns the visual features from training data to infer the optimal solution $\boldsymbol{\mathcal{V}}_{k+1}$. Let $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$ be the CNN representing the proximal operator $\mathrm{prox}_{f_{\mathrm{reg}}}(\cdot)$ for the variable $\boldsymbol{\mathcal{V}}$ in (31). Then, $\boldsymbol{\mathcal{V}}_{k+1}$ is computed as

$$\boldsymbol{\mathcal{V}}_{k+1} = f_{\boldsymbol{\mathcal{V}}\text{-CNN}}(\boldsymbol{\mathcal{X}}_{k+1} + \alpha_k^{-1}\boldsymbol{\Gamma}_k; \boldsymbol{\Theta}_{\boldsymbol{\mathcal{V}},k}), \tag{32}$$

where $\boldsymbol{\Theta}_{\boldsymbol{\mathcal{V}},k}$ denotes the parameters of $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$ at the k-th iteration.

```
Algorithm 1 UTeRM: Optimization for Solving (6)
Input: $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{h\times w\times C}$, $\mathbf{P} \in \mathbb{R}^{H\times W}$, $r = C/2$, and $K$.
Initialize $k = 1$.
Initialize $\boldsymbol{\mathcal{X}}_0$, $\boldsymbol{\mathcal{V}}_0$, and $\boldsymbol{\Lambda}_0$ as zero tensors.
Initialize $\boldsymbol{\mathcal{L}}_0$ and $\boldsymbol{\mathcal{R}}_0$ as tensors of $10^{-2}$.
Obtain $\boldsymbol{\mathcal{X}}_{\mathrm{DI}}$ using (11), (16), or (18).
while $k \leq K$ do
    Update $\boldsymbol{\mathcal{X}}$ using (24), (25), or (26) depending on $\boldsymbol{\mathcal{X}}_{\mathrm{DI}}$.
    Update $\boldsymbol{\mathcal{L}}$ using (28).
    Update $\boldsymbol{\mathcal{R}}$ using (29).
    Update $\boldsymbol{\mathcal{V}}$ using (32).
    Update $\boldsymbol{\Lambda}$ using (33).
    $k = k + 1$.
end while
Output: $\boldsymbol{\mathcal{X}}^{*} = \boldsymbol{\mathcal{X}}_K$
```

Multipliers: Finally, the tensor of Lagrange multiplier $\boldsymbol{\Lambda}$ is updated as

$$\boldsymbol{\Lambda}_{k+1} = \boldsymbol{\Lambda}_k + \alpha_k(\boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{V}}_{k+1}). \tag{33}$$

The complete UTeRM algorithm with generalized detail injection is summarized in Algorithm 1.

D. Deep Unfolded Network

Finally, we establish a deep learning architecture for pansharpening by unfolding the iterations of the iterative tensor rank minimization algorithm developed in the previous section. Fig. 2 illustrates a deep unfolded network that takes the LRMS image $\boldsymbol{\mathcal{M}}$ and PAN image $\mathbf{P}$ as inputs and produces the HRMS image $\boldsymbol{\mathcal{X}}^{*}$.

The proposed network comprises two phases: detail injection and tensor rank minimization. In the first phase, $\boldsymbol{\mathcal{M}}$ and $\mathbf{P}$ are input into the detail injection module described in Section III-B to produce $\boldsymbol{\mathcal{X}}_{\text{DI}}$. Then, in the second phase, $\boldsymbol{\mathcal{M}}$, $\mathbf{P}$, and $\boldsymbol{\mathcal{X}}_{\text{DI}}$ are input into the K-stage unfolded network, where each stage corresponds to an iteration of the proposed iterative tensor rank minimization algorithm, as described in Section III-C. Specifically, the variables $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{L}}$, and $\boldsymbol{\mathcal{R}}$ are updated by the closed-form solutions in (23), (28), and (29), respectively, and the auxiliary variable $\boldsymbol{\mathcal{V}}$ is updated by the CNN-based proximal operator $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$ in (32). In addition to the parameters $\boldsymbol{\Theta}_{\boldsymbol{\mathcal{V}},k}$ for $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$, the hyperparameters α_k in (23) and λ_k in (24), (25), and (26) are adjusted using backpropagation during the training process. Finally, the output of the last stage is the desired HRMS image, *i.e.*, $\boldsymbol{\mathcal{X}}^* = \boldsymbol{\mathcal{X}}_K$.

The CNN-based proximal operator $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$ is intended to compensate for modeling inaccuracies by providing inferences based on learned visual features; thus, $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$ can be implemented using any deep network architecture that can effectively extract features from the training data. For simplicity, we employ the residual dense block (RDB) [52] to implement $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$. Specifically, we use an RDB with eight convolutional layers to construct $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}$, which takes the tensor $(\boldsymbol{\mathcal{X}}_{k+1} + \alpha_k^{-1}\boldsymbol{\Gamma}_k)$ as input and produces the optimal solution $\boldsymbol{\mathcal{V}}_{k+1}$. We will discuss the effects of the number of convolutional layers in the RDB in Section IV-D4.

E. Training

To train the proposed algorithm, we define a loss function that considers the fidelity of the generated HRMS images and the effect of the detail injection module as

$$L_{\text{total}} = L_{\text{fidelity}} + \omega L_{\text{detail}}, \tag{34}$$

where L_{fidelity} and L_{detail} denote the losses on the pansharpened and detail injected results, respectively, and $\omega = 0.25$ is the parameter for balancing the relative importance between the two terms. The fidelity loss L_{fidelity} is computed as the ℓ_1-norm of the difference between the ground-truth $\boldsymbol{\mathcal{X}}_{\text{gt}}$ and the pansharpened HRMS image $\boldsymbol{\mathcal{X}}^*$ as

$$L_{\text{fidelity}} = \|\boldsymbol{\mathcal{X}}_{\text{gt}} - \boldsymbol{\mathcal{X}}^*\|_1. \tag{35}$$

The detail injection loss L_{detail} is defined similarly to $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$ for the CS-, MRA-, and CNN-based detail injections, respectively, as

$$L^{\text{CS}}_{\text{detail}} = \|\boldsymbol{\mathcal{X}}_{\text{gt}} - (\boldsymbol{\mathcal{M}}_{\uparrow} + \boldsymbol{\mathcal{G}} \odot (\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{I}}_L))\|_1, \tag{36}$$

$$L^{\text{MRA}}_{\text{detail}} = \|\boldsymbol{\mathcal{X}}_{\text{gt}} - (\boldsymbol{\mathcal{M}}_{\uparrow} + \boldsymbol{\mathcal{G}} \odot (\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{P}}_{\text{low}}))\|_1, \tag{37}$$

$$L^{\text{CNN}}_{\text{detail}} = \|\boldsymbol{\mathcal{X}}_{\text{gt}} - \mathcal{F}_{\text{CNN}}(\boldsymbol{\mathcal{M}}, \mathbf{P})\|_1. \tag{38}$$

We train the proposed network using the Adam optimizer [53] and L_{total} for 90 epochs with a learning rate of $\eta = 10^{-5}$. Then, we fine-tune the network using only L_{fidelity} for 10 epochs with $\eta = 10^{-6}$. Note that since the Fourier transform is involved, it is difficult to implement multi-batch training; thus, the proposed algorithm uses a batch size of one for training.

IV. Experimental Results

A. Datasets and Settings

We evaluate the performance of the proposed algorithm against conventional algorithms on three datasets containing 200 MS–PAN image pairs captured by the IKONOS satellite, 500 pairs captured by the WorldView-2 satellite, and 270 pairs captured by the WorldView-4 satellite [1]. The specifications of the datasets are provided in Table I. We randomly divide the IKONOS and WorldView-2 datasets into training (70%) and testing (30%) sets. To evaluate the generalizability, we select all MS images from the ‘green vegetation’ and ‘water scenario’ categories in the WorldView-4 dataset as training set and those from the ‘urban’ category as testing set.[1]

TABLE I
Settings for the Datasets in the Experiments

Satellite	IKONOS	WorldView-2	WorldView-4
Number of MS bands	4	8	4
Size of an MS image	256 × 256	256 × 256	256 × 256
GSD* in an MS	4.0 m	2.0 m	1.24 m
Size of a PAN image	1024 × 1024	1024 × 1024	1024 × 1024
GSD* in a PAN	1.0 m	0.5 m	0.31 m
Radiometric resolution	11-bit	11-bit	11-bit
Training/test pairs	140/60	350/150	180/90

*Ground sampling distance

Since ground-truths are unavailable, MS and PAN images are degraded and downsampled using Wald’s protocol [54], implemented by Vivone *et al.* [55],[2] to generate reduced-resolution (RR) LRMS and PAN images. Specifically, the MS images are low-pass filtered using kernels designed to consider the characteristics of sensors, followed by bicubic interpolation to generate LRMS images, while PAN images are generated by bicubic interpolation. The original MS images are used as the reference HRMS images. In the RR experiment, we set the resolution ratio to $r = 4$, *i.e.*, the generated LRMS and PAN images have resolutions of 64×64 and 256×256, respectively. The corresponding full-resolution (FR) images of the test set are used to conduct the FR experiment, *i.e.*, the original MS and PAN images are used as input. In this work, the tubal rank r in (5) of the desired HRMS image is empirically set to half the number of spectral channels.

We compare the performance of all three variants of the proposed algorithm with those of the state-of-the-art algorithms. We select notable model-based algorithms: MF [11], BDSD-PC [6], MTF-GLP-HPM-DS [12] (referred to as HPM-DS), and the latest LRTC approach, LRTCFPan [24]. We also compare the proposed algorithm with DL-based algorithms—MSDCNN [27], DiCNN [28], FusionNet [13], MDA-Net [44], MSDDN [32], TRRNet [33], and PCTINN [34]—and model-based deep networks—GPPNN [39] and MD^{3}Net [35]. The source codes for LRTCFPan, GPPNN, MD^{3}Net, MDA-Net, MSDDN, TRRNet, and PCTINN, were provided by the respective authors, whereas those for the other algorithms were obtained from the benchmark provided by Deng *et al.* [49]. The DL-based algorithms, including the proposed UTeRM, were retrained using each dataset and the settings recommended by the respective authors. The source code and

[1] The original authors of [1] provided the categories of MS images.

[2] `https://openremotesensing.net/knowledgebase/a-benchmarking-protocol-for-pansharpening-dataset-preprocessing-and-quality-assessment`

TABLE II
QUANTITATIVE EVALUATION OF THE REDUCED-RESOLUTION TEST ON THE IKONOS DATASET. THE ↑ AND ↓ SYMBOLS DENOTE "HIGHER IS BETTER" AND "LOWER IS BETTER," RESPECTIVELY. FOR EACH METRIC, THE **BOLDFACED** AND UNDERLINED NUMBERS DENOTE THE BEST AND SECOND-BEST RESULTS, RESPECTIVELY

	PSNR (↑)	SSIM (↑)	SAM (↓)	SCC (↑)	ERGAS (↓)	$Q4$ (↑)	D_λ (↓)	D_S (↓)	QNR (↑)
BDSD-PC [6]	39.90	0.9320	2.4253	0.9451	1.6990	0.8186	0.0735	0.0972	0.8387
MF [11]	39.20	0.9279	2.4339	0.9416	1.7784	0.8090	0.1210	0.1385	0.7603
HPM-DS [12]	39.95	0.9311	2.3425	0.9453	1.6748	0.8146	0.0945	0.1151	0.8047
LRTCFPan [24]	40.47	0.9342	2.1654	0.9517	1.6002	0.8205	0.0579	**0.0568**	0.8897
MSDCNN [27]	41.45	0.9527	1.9445	0.9615	1.4083	0.8683	0.0650	0.0862	0.8559
DiCNN [28]	41.24	0.9510	1.9685	0.9603	1.4358	0.8611	0.0605	0.0835	0.8625
FusionNet [13]	41.77	0.9553	1.8543	0.9638	1.3696	0.8786	0.0479	0.0677	0.8887
GPPNN [39]	41.48	0.9574	1.8248	0.9650	1.3893	0.8656	0.0565	0.0787	0.8703
MD^{3}Net [35]	42.32	0.9597	1.7326	0.9677	1.3082	0.8772	0.0487	0.0719	0.8839
MDA-Net [44]	42.24	0.9605	1.7335	0.9685	1.2950	0.8864	0.0461	0.0661	0.8917
MSDDN [32]	42.52	0.9615	1.7026	0.9692	1.2819	0.8855	0.0485	0.0685	0.8872
TRRNet [33]	41.04	0.9511	1.9943	0.9601	1.4658	0.8630	0.0487	0.0611	**0.8940**
PCTINN [34]	42.40	0.9616	1.7276	0.9684	1.2834	0.8830	0.0509	0.0675	0.8856
UTeRM-CS	42.66	**0.9632**	1.6616	0.9705	1.2644	0.8859	0.0445	0.0678	0.8915
UTeRM-MRA	42.60	0.9629	1.6689	0.9704	1.2639	0.8864	**0.0443**	0.0658	0.8936
UTeRM-CNN	**42.78**	0.9630	**1.6514**	**0.9708**	**1.2564**	**0.8874**	0.0446	0.0673	0.8919

pretrained models will be released on our project website.[3]

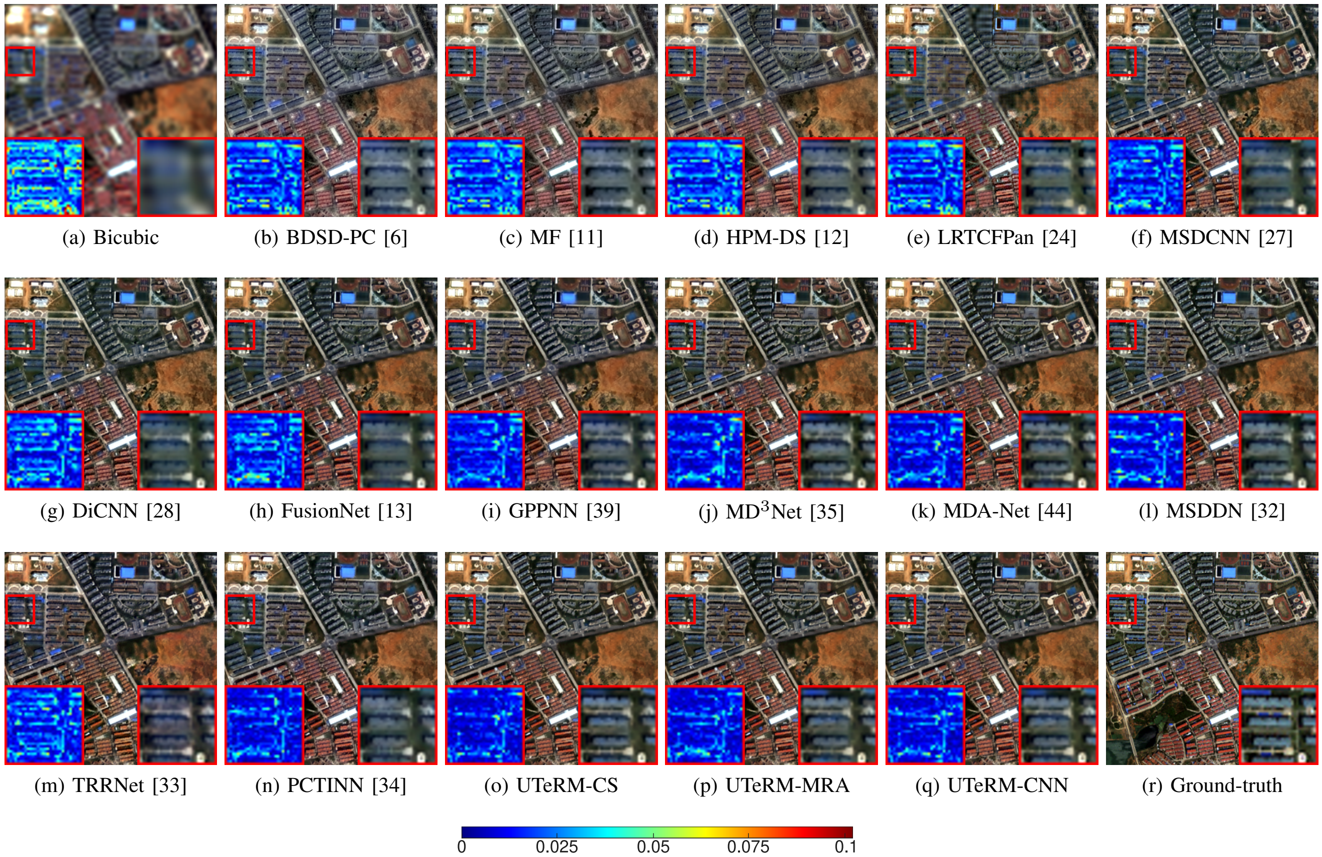
For quantitative comparisons, in the RR experiments, we use both full-reference metrics, including PSNR, SSIM [56], spectral angle mapper (SAM) [57], spatial correlation coefficient (SCC) [58], relative dimensionless global error in synthesis (ERGAS) [54], and universal image quality index for 2^n-band images ($Q2^n$) [59], and non-reference metrics, including the spectral distortion index D_λ, spatial distortion index D_S, and quality without reference (QNR), which is computed from D_λ and D_S [60]. In the FR experiments, we evaluate the performance of the algorithms using three non-reference metrics D_λ, D_S, and QNR. The PSNR measures the quality of the pixel value reconstruction, whereas the SSIM measures the structural similarity between two images. The SAM metric measures the similarity between the spectra of the HRMS images, and the SCC metric quantifies the correlation of the high-frequency components between two HRMS images. The ERGAS and $Q2^n$ metrics measure the spectral distortions of each band and joint inter- and intra-band, respectively, of the pansharpened image. The non-reference metrics D_λ and D_S measure the spectral and spatial distortions, respectively, and QNR is the overall quality index computed from D_λ and D_S. Higher PSNR, SSIM, SCC, $Q2^n$, and QNR scores imply better results, whereas low SAM, ERGAS, D_λ, and D_S scores indicate better performance.

B. Reduced-Resolution Assessment

1) IKONOS (4-band Sensor): Table II quantitatively compares the pansharpening performance on the IKONOS dataset. All three variants of the proposed UTeRM outperform all the conventional algorithms. For example, UTeRM-CNN achieves 2.31 dB and 0.0288 higher PSNR and SSIM scores, respectively, than that of the best model-based algorithm, LRTCFPan [24], indicating the highest fidelity of the pansharpened HRMS images. In addition, the proposed algorithm outperforms all DL-based algorithms on most metrics because of the use of a robust low-rank model with a learnable regularizer to compensate for modeling inaccuracy. For example, UTeRM-CNN yields 0.26 dB and 0.0015 higher PSNR and SSIM scores than the best DL-based algorithm MSDDN [32]. Furthermore, these variants provide the best scores in the metrics for evaluating the HRMS images, *i.e.*, SAM, SCC, ERGAS, and $Q4$, indicating the effectiveness of the proposed algorithm in restoring information across spectral bands. In particular, UTeRM-CNN achieves the best SAM and $Q4$ scores, implying that the spectra of its pansharpened HRMS images are the most similar to those of the ground-truths. The best SCC and ERGAS scores of the proposed algorithm indicate high-fidelity pansharpened results. Note that as CS-based algorithms can compute more complex transformations with more hyperparameters [61], UTeRM-CS achieves overall better performance than UTeRM-MRA for 4-band MS datasets. Finally, UTeRM-CNN performs the best among the variants because of the ability of the CNN-based detail injection component to learn visual features compared to the others.

Fig. 3 compares the pansharpened HRMS images produced by each algorithm, including bicubic upsampling in Fig. 3(a), for the 19th image in the IKONOS test set. The CS-based algorithm BDSD-PC [6] and MRA-based algorithms MF [11] and HPM-DS [12] yield blurry results as shown in Figs. 3(b)–(d). LRTCFPan [24] in Fig. 3(e) provides better results by adopting a low-rank model; however, fine details are lost, generating blurring artifacts, as it is challenging to determine the set of reliable pixels. DL-based MSDCNN [27], DiCNN [28], and FusionNet [13] in Figs. 3(f)–(h) produce sharper results; however, since they employ CNNs to generate HRMS images through direct mappings from a pair of LRMS and PAN images, they severely lose high-frequency details, *e.g.*, the buildings appear blurry and are unclearly separated from the background. Although more recent DL-

[3] https://github.com/mtntruong/UTeRM



(a) Bicubic (b) BDSD-PC [6] (c) MF [11] (d) HPM-DS [12] (e) LRTCFPan [24] (f) MSDCNN [27]

(g) DiCNN [28] (h) FusionNet [13] (i) GPPNN [39] (j) MD^{3}Net [35] (k) MDA-Net [44] (l) MSDDN [32]

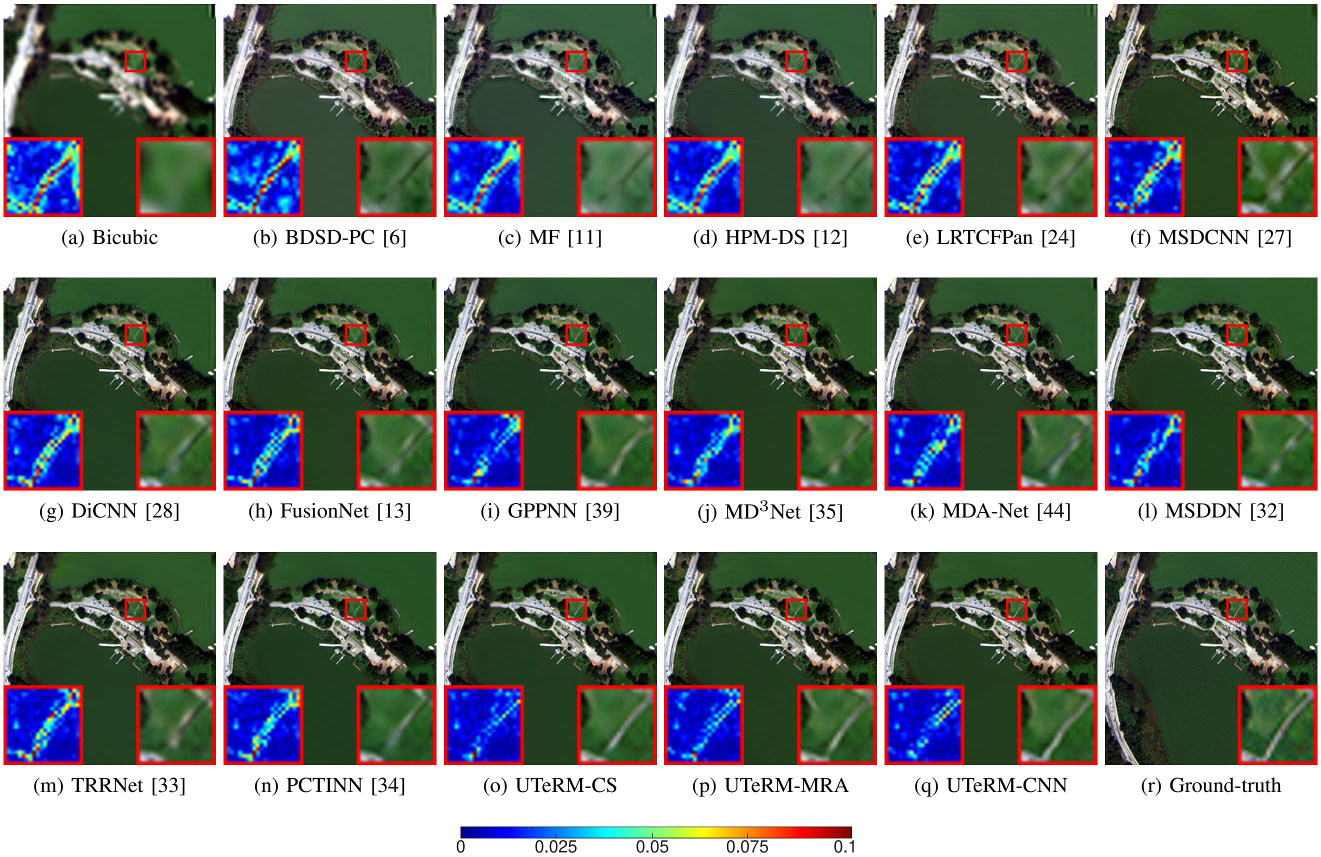
(m) TRRNet [33] (n) PCTINN [34] (o) UTeRM-CS (p) UTeRM-MRA (q) UTeRM-CNN (r) Ground-truth

Fig. 3. Comparison of the results for the 19th image of the IKONOS dataset in the RR experiment. The magnified region and corresponding error map for the red square are shown. The proposed UTeRMs in (o)–(q) generate faithful details, *e.g.*, the buildings in the magnified regions are sharp and clearly separated from the background.

TABLE III
QUANTITATIVE EVALUATION OF THE REDUCED-RESOLUTION TEST ON THE WORLDVIEW-2 DATASET

	PSNR (↑)	SSIM (↑)	SAM (↓)	SCC (↑)	ERGAS (↓)	$Q8$ (↑)	D_λ (↓)	D_S (↓)	QNR (↑)
BDSD-PC [6]	37.96	0.9241	4.5521	0.9422	3.8493	0.6877	0.0705	0.0838	0.8530
MF [11]	36.31	0.9053	4.5485	0.9274	4.3575	0.6068	0.0954	0.0838	0.8300
HPM-DS [12]	37.02	0.9084	4.6654	0.9282	6.7349	0.6151	0.0779	0.0878	0.8427
LRTCFPan [24]	38.23	0.9289	4.1523	0.9473	3.6676	0.6741	**0.0409**	0.0564	**0.9052**
MSDCNN [27]	39.49	0.9574	3.4463	0.9611	3.0664	0.7575	0.0629	0.0687	0.8731
DiCNN [28]	39.48	0.9577	3.4305	0.9618	3.0812	0.7526	0.0554	0.0646	0.8841
FusionNet [13]	39.99	0.9615	3.2481	0.9649	2.8841	0.7699	0.0580	0.0690	0.8775
GPPNN [39]	40.18	0.9644	3.1459	0.9661	2.8153	0.7795	0.0532	0.0641	0.8867
MD^{3}Net [35]	40.31	0.9643	3.1051	0.9670	2.7823	0.7743	0.0509	0.0664	0.8866
MDA-Net [44]	40.49	0.9654	3.0719	0.9691	2.6732	0.7816	0.0521	0.0621	0.8894
MSDDN [32]	40.19	0.9640	3.1073	0.9662	2.8196	0.7727	0.0459	0.0545	0.9025
TRRNet [33]	39.68	0.9600	3.3110	0.9627	2.9865	0.7666	0.0581	0.0573	0.8882
PCTINN [34]	40.07	0.9634	3.1515	0.9651	2.8772	0.7625	0.0499	0.0618	0.8917
UTeRM-CS	40.60	0.9665	3.0159	0.9689	2.6672	0.7840	0.0458	**0.0540**	0.9028
UTeRM-MRA	40.66	**0.9669**	**2.9911**	0.9693	2.6420	**0.7849**	0.0462	0.0589	0.8979
UTeRM-CNN	**40.71**	0.9668	2.9967	**0.9694**	**2.6364**	0.7846	0.0447	0.0601	0.8982

based algorithms MDA-Net [44], MSDDN [32], TRRNet [33], and PCTINN [34] in Figs. 3(k)–(n) and model-based deep networks GPPNN [39] and MD^{3}Net [35] in Figs. 3(i) and (j), respectively, generate better HRMS images, the fine details are still unfaithful to the ground-truth, as indicated in the error maps. Moreover, TRRNet in Fig. 3(m) exhibits color differences, implying spectral distortions. In contrast, as shown in Figs. 3(o)–(q), the HRMS images obtained by the proposed UTeRM-CS, UTeRM-MRA, and UTeRM-CNN are sharper, and their fine details are faithful to the ground-truth, indicating that the proposed low-rank model effectively captures the physical properties of MS images without the need for


(a) Bicubic (b) BDSD-PC [6] (c) MF [11] (d) HPM-DS [12] (e) LRTCFPan [24] (f) MSDCNN [27]

(g) DiCNN [28] (h) FusionNet [13] (i) GPPNN [39] (j) MD^{3}Net [35] (k) MDA-Net [44] (l) MSDDN [32]

(m) TRRNet [33] (n) PCTINN [34] (o) UTeRM-CS (p) UTeRM-MRA (q) UTeRM-CNN (r) Ground-truth

Fig. 4. Comparison of the results for the 150th image in the WorldView-2 dataset in the RR experiment. The proposed UTeRMs in (o)–(q) faithfully recover the fine details in the magnified regions, while others generate poor textures.

information on reliable pixels.

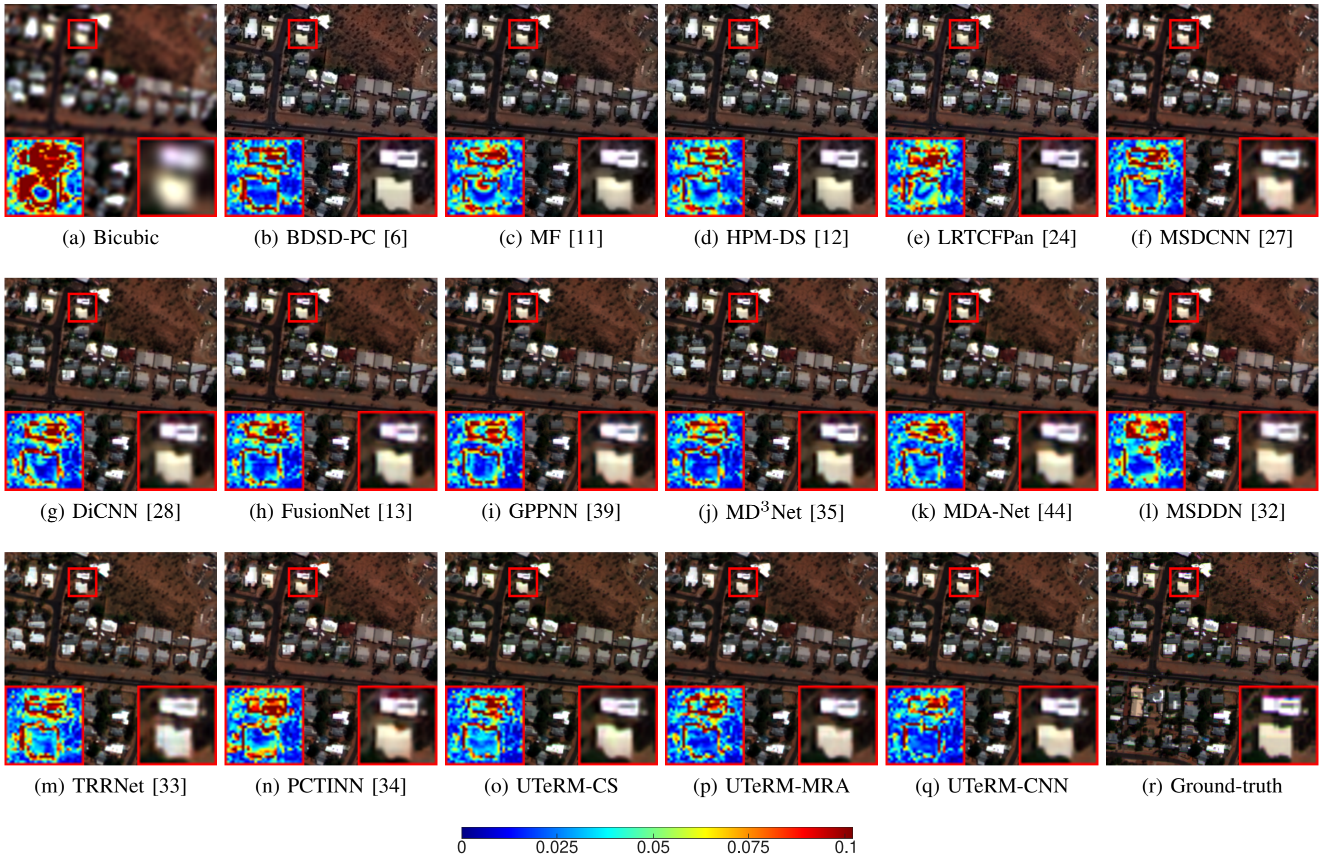
2) WorldView-2 (8-band Sensor): Table III quantitatively compares the pansharpening performance of the algorithms on the WorldView-2 dataset. The results exhibit tendencies similar to those in Table II. Specifically, all variants of the proposed UTeRM outperform both model- and DL-based algorithms by large margins for all full-reference metrics, indicating the highest fidelity for spectral and spatial details. For non-reference metrics, UTeRM-CS achieves the lowest D_S and second-highest QNR scores; however, the scores of D_λ and QNR for UTeRM-CS and the best LRTCFPan are comparable. Since CS-based algorithms require more hyperparameters than MRA-based algorithms, estimating accurate results for an 8-band dataset is challenging [61]. Therefore, contrary to a 4-band dataset in Table II, UTeRM-MRA outperforms UTeRM-CS. In addition, UTeRM-MRA and UTeRM-CNN provide comparable and overall the best results. Specifically, UTeRM-MRA achieves the best SSIM, SAM, and Q8 scores, indicating the highest spectral fidelity and lowest inter- and intra-band spectral distortions. In contrast, UTeRM-CNN yields the highest PSNR and SCC and lowest ERGAS by restoring the spatial information more faithfully. This is because the MRA-based detail injection formulation can accurately model the complex spectral properties in the 8-band dataset, whereas the black-box CNN-based detail injection cannotlearn these features effectively.

Fig. 4 compares the pansharpened HRMS images obtained by each algorithm for the 150th image in the WorldView-2 test set. In Figs. 4(b)–(d), the model-based algorithms BDSD-PC, MF, and HPM-DS fail to restore fine details, producing blurry HRMS images because they have limited capability to inject the information into the LRMS image, especially when the LRMS image has poor textures. Although the low-rank-based LRTCFPan in Fig. 4(e) produces superior performance in terms of non-reference metrics in Table III, it suffers from the same weaknesses as other model-based algorithms due to its MRA-based detail injection. MSDCNN, DiCNN, and FusionNet in Figs. 4(f)–(h), respectively, produce the HRMS images with inaccurately restored details because of the information loss caused by direct mappings. Model-based deep networks, GPPNN and MD^{3}Net, in Figs. 4(i) and (j), respectively, and transformer-based TRRNet and PCTINN in Figs. 4(m) and (n), respectively, generate better HRMS images. However, they still yield visible artifacts due to inaccurate restoration. In contrast, the variants of the proposed algorithm in Figs. 4(o)–(q) restore the HRMS images with details most faithful to the ground-truth due to the robust tensor rank minimization model and learnable regularization.

3) WorldView-4 (4-band Sensor): As described in Section IV-A, we evaluate the pansharpening performance using test MS images from the WorldView-4 dataset, which belong to different categories from those used for training, to assess

TABLE IV
QUANTITATIVE EVALUATION OF THE REDUCED-RESOLUTION TEST ON THE WORLDVIEW-4 DATASET

	PSNR (↑)	SSIM (↑)	SAM (↓)	SCC (↑)	ERGAS (↓)	$Q4$ (↑)	D_λ (↓)	D_S (↓)	QNR (↑)
BDSD-PC [6]	28.24	0.8253	4.8065	0.9347	4.1015	0.6404	0.0515	0.0941	0.8594
MF [11]	28.63	0.8378	4.6850	0.9391	3.8451	0.6069	0.0954	0.0999	0.8145
HPM-DS [12]	29.03	0.8334	4.9353	0.9420	3.7658	0.6212	0.0871	0.1049	0.8172
LRTCFPan [24]	29.24	0.8398	4.1004	0.9461	3.6632	0.6356	0.0356	0.0599	0.9066
MSDCNN [27]	29.76	0.8663	3.6920	0.9526	3.4172	0.6723	0.0302	0.0554	0.9162
DiCNN [28]	29.62	0.8606	3.7364	0.9508	3.4844	0.6566	0.0327	0.0567	0.9126
FusionNet [13]	29.69	0.8641	3.6671	0.9515	3.4501	0.6742	0.0288	0.0539	0.9189
GPPNN [39]	29.77	0.8742	3.6995	0.9533	3.3991	0.6775	0.0317	0.0575	0.9128
MD^{3}Net [35]	29.91	0.8743	3.6033	0.9540	3.3515	0.6861	0.0342	0.0667	0.9015
MDA-Net [44]	29.89	0.8719	3.5868	0.9557	3.3024	0.6768	**0.0202**	0.0402	0.9405
MSDDN [32]	29.90	0.8793	3.6595	0.9560	3.3198	0.6541	0.0298	**0.0285**	**0.9428**
TRRNet [33]	28.91	0.8589	4.0911	0.9452	3.7558	0.6715	0.0320	0.0506	0.9192
PCTINN [34]	28.97	0.8545	4.1221	0.9429	3.7657	0.6212	0.0243	0.0471	0.9299
UTeRM-CS	29.91	0.8792	3.4599	0.9549	3.3430	0.6850	0.0327	0.0575	0.9119
UTeRM-MRA	29.54	0.8703	3.4854	0.9503	3.5239	0.6797	0.0283	0.0514	0.9219
UTeRM-CNN	**30.13**	**0.8831**	**3.3980**	**0.9565**	**3.2717**	**0.6949**	0.0282	0.0549	0.9186


(a) Bicubic (b) BDSD-PC [6] (c) MF [11] (d) HPM-DS [12] (e) LRTCFPan [24] (f) MSDCNN [27]

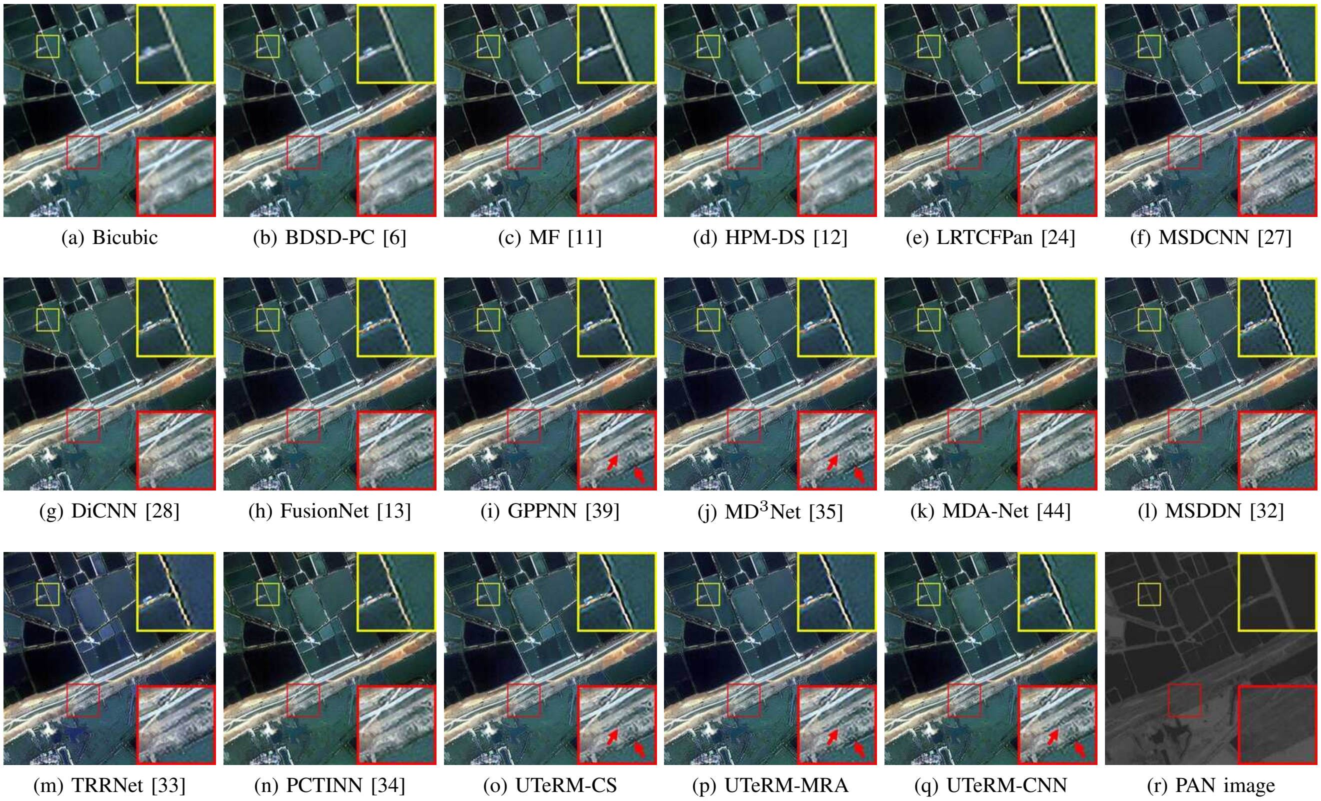
(g) DiCNN [28] (h) FusionNet [13] (i) GPPNN [39] (j) MD^{3}Net [35] (k) MDA-Net [44] (l) MSDDN [32]

(m) TRRNet [33] (n) PCTINN [34] (o) UTeRM-CS (p) UTeRM-MRA (q) UTeRM-CNN (r) Ground-truth

Fig. 5. Comparison of the results for the 19th image of the WorldView-4 dataset in the RR experiment. The proposed UTeRMs in (o)–(q) faithfully recover the fine details in the magnified regions, while others generate blurry textures.

the generalizability. Table IV presents the quantitative assessment results for the pansharpening algorithms. The results exhibit tendencies similar to the results in Table II. Specifically, the scores in all full-reference metrics of the proposed UTeRM-CNN consistently remain the highest, indicating its superior generalizability. In addition, UTeRM-CS achieves overall better performance than UTeRM-MRA for 4-band datasets, as discussed previously in the results in Table II.

Fig. 5 compares the pansharpened results of each algorithm for the 19th image in the WorldView-4 test set. The model-based algorithms in Figs. 5(b)–(e) produce blurry HRMS images. The DL-based algorithms in Figs. 4(f)–(n) generate better pansharpend results; however, they lose the strong edge information on the buildings in the red rectangles, resulting in



(a) Bicubic (b) BDSD-PC [6] (c) MF [11] (d) HPM-DS [12] (e) LRTCFPan [24] (f) MSDCNN [27]

(g) DiCNN [28] (h) FusionNet [13] (i) GPPNN [39] (j) MD^{3}Net [35] (k) MDA-Net [44] (l) MSDDN [32]

(m) TRRNet [33] (n) PCTINN [34] (o) UTeRM-CS (p) UTeRM-MRA (q) UTeRM-CNN (r) PAN image

Fig. 6. Comparison of the results for the 18th image of the IKONOS dataset in the FR experiment. The proposed UTeRMs in (o)–(q) generate HRMS images with finer details compared to others.

large error. In contrast, the variants of the proposed algorithm in Figs. 4(o)–(q) restore the HRMS images with finer textures and higher spectral fidelity. These results demonstrate that the proposed UTeRM has superior generalizability compared to other DL-based algorithms. Furthermore, note that MDA-Net and MSDDN in Figs. 4(k) and (l), respectively, provide the overall best non-reference score, but their visual quality is significantly inferior to that of the proposed UTeRMs.

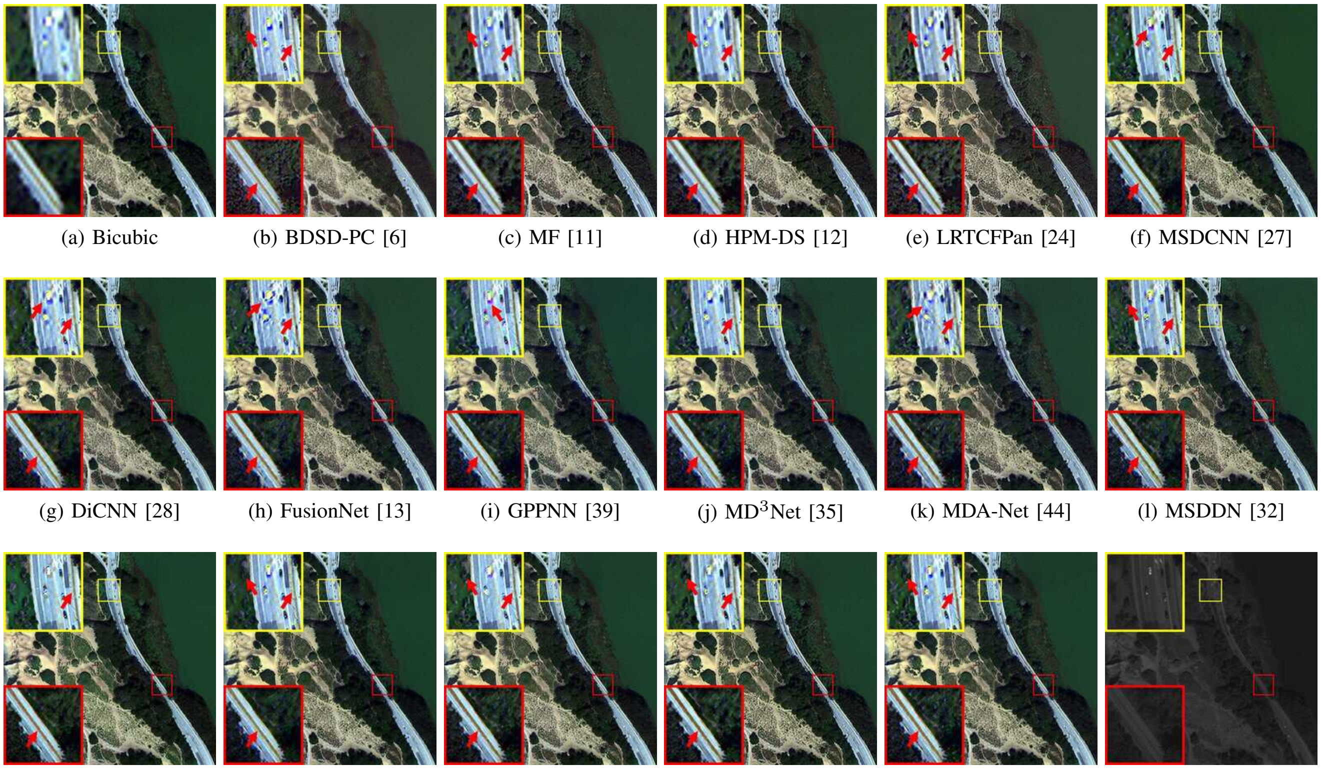
C. Full-Resolution Assessment

1) IKONOS (4-band Sensor): Table V quantitatively compares the pansharpening performance of the algorithms on the IKONOS dataset using non-reference metrics. The proposed UTeRM-MRA achieves the best D_S score, 0.0015 lower than the second-best TRRNet, indicating the lowest spatial distortion, whereas spectral distortions in D_λ are comparable to those of the state-of-the-arts. UTeRM-CNN achieves the second-best QNR score, comparable to the best algorithm, MDA-Net, indicating high-fidelity HRMS images. However, the proposed UTeRM is significantly more efficient than MDA-Net in terms of the number of trainable parameters and runtime, as will be discussed in Section IV-E. To summarize, the proposed algorithm performs comparably to that of the best MDA-Net on the IKONOS dataset in the FR experiment due to the tensor rank minimization formulation, effectively representing the physical properties of the MS images.

TABLE V
QUANTITATIVE EVALUATION OF THE FULL-RESOLUTION TEST ON THE IKONOS DATASET

	D_λ (↓)	D_S (↓)	QNR (↑)
BDSD-PC [6]	0.0875	0.1157	0.8103
MF [11]	0.1392	0.1528	0.7315
HPM-DS [12]	0.1029	0.1216	0.7905
LRTCFPan [24]	0.0897	0.0991	0.8218
MSDCNN [27]	0.0482	0.0476	0.9086
DiCNN [28]	0.0464	0.0514	0.9066
FusionNet [13]	0.0528	0.0424	0.9085
GPPNN [39]	0.0679	0.0533	0.8847
MD^{3}Net [35]	0.0620	0.0415	0.9001
MDA-Net [44]	**0.0463**	0.0381	**0.9191**
MSDDN [32]	0.0699	0.0477	0.8868
TRRNet [33]	0.0554	0.0380	0.9092
PCTINN [34]	0.0514	0.0454	0.9068
UTeRM-CS	0.0639	0.0440	0.8963
UTeRM-MRA	0.0613	**0.0365**	0.9053
UTeRM-CNN	0.0519	0.0394	0.9119

Fig. 6 compares the pansharpened HRMS images obtained by each algorithm for the 18th image of the IKONOS test set. The corresponding PAN image is included for a reference in Fig. 6(r). In Figs. 6(b)–(e), MF, BDSD-PC, HPM-DS, and LRTCFPan generate jagged edges in the yellow rectangles and blurry results in the red rectangles due to the failure of the detail extraction from the PAN image. In addition, MSDCNN, DiCNN, and FusionNet in Figs. 6(f)–(h) also generate visible artifacts because their CNN-based mappings learn insufficient visual features for HRMS image reconstruction. Further,



(a) Bicubic (b) BDSD-PC [6] (c) MF [11] (d) HPM-DS [12] (e) LRTCFPan [24] (f) MSDCNN [27]

(g) DiCNN [28] (h) FusionNet [13] (i) GPPNN [39] (j) MD^{3}Net [35] (k) MDA-Net [44] (l) MSDDN [32]

(m) TRRNet [33] (n) PCTINN [34] (o) UTeRM-CS (p) UTeRM-MRA (q) UTeRM-CNN (r) PAN image

Fig. 7. Comparison of the results for the 144th image of the WorldView-2 dataset in the FR experiment. The proposed UTeRMs in (o)–(q) provide results with clearer and sharper textures on the roads and vehicles compared to others.

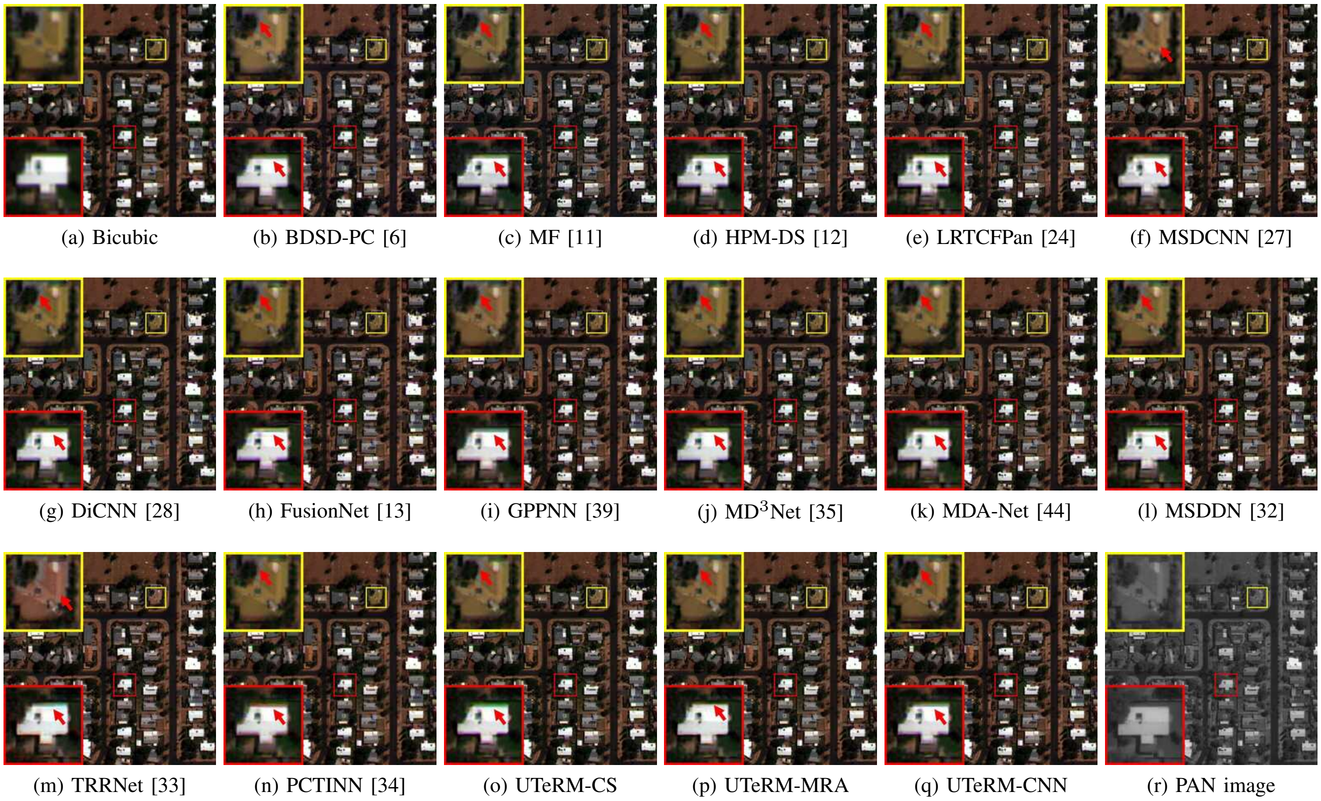
MDA-Net, MSDDN, and PCTINN in Figs. 6(k), (l), and (n), respectively, degrade the quality due to over-sharpening, generating ringing artifacts in the yellow rectangles, whereas TRRNet in Fig. 6(m) exhibits spectral distortions. GPPNN and MD^{3}Net in Figs. 6(i) and (j), respectively, generate better HRMS images, but their results still exhibit blurry artifacts in the regions indicated by the arrows. In contrast, the proposed UTeRMs in Figs. 6(o)–(q) produce HRMS images with rich details without visible artifacts, proving the effectiveness of the tensor rank minimization model.

2) WorldView-2 (8-band Sensor): Table VI quantitatively compares the pansharpening performance of the algorithms on the WorldView-2 dataset using non-reference metrics. The proposed UTeRM-CS provides the overall best FR pansharpening performance. Specifically, it exhibits 0.0448 lower D_S and 0.0132 higher QNR scores than the second-best algorithm, MSDDN, indicating its superior performance in terms of spatial and spectral distortions. This result also confirms the effectiveness of the proposed tensor rank minimization and generalized detail injection component, which can be flexibly modified to achieve better results based on various requirements.

TABLE VI
QUANTITATIVE EVALUATION OF THE FULL-RESOLUTION TEST ON THE WORLDVIEW-2 DATASET

	D_λ (↓)	D_S (↓)	QNR (↑)
BDSD-PC [6]	0.1446	0.1529	0.7465
MF [11]	0.1602	0.1751	0.7066
HPM-DS [12]	0.1424	0.1675	0.7302
LRTCFPan [24]	0.1219	0.1489	0.7559
MSDCNN [27]	0.1594	0.1643	0.7320
DiCNN [28]	**0.1157**	0.1470	0.7730
FusionNet [13]	0.1313	0.1528	0.7570
GPPNN [39]	0.1649	0.1714	0.7171
MD^{3}Net [35]	0.1183	0.1495	0.7673
MDA-Net [44]	0.1197	0.1466	0.7674
MSDDN [32]	0.1158	0.1364	0.7810
TRRNet [33]	0.1590	0.1453	0.7424
PCTINN [34]	0.1233	0.1578	0.7558
UTeRM-CS	0.1371	**0.0916**	**0.7942**
UTeRM-MRA	0.1261	0.1429	0.7673
UTeRM-CNN	0.1378	0.1553	0.7473

Fig. 7 compares the pansharpened results of each algorithm for the 144th image in the WorldView-2 test set. They exhibit tendencies similar to those in Fig. 6. Specifically, since the PAN image in Fig. 7(r) has low contrast with poor details, the model-based algorithms in Figs. 7(b)–(e) fail to extract details from the PAN image, yielding blurry textures, *e.g.*, the roads. Further, MSDCNN, DiCNN, FusionNet, and MDA-Net in Figs. 7(f)–(h) and (k), respectively, result in color artifacts in the zoomed rectangles, *e.g.*, the vehicles in the yellow rectangles. Moreover, GPPNN and MD^{3}Net generate spectral and spatial distortions. For example, GPPNN in Fig. 7(i) yields color artifacts in the cars, and MD^{3}Net in Fig. 7(j) produces blurry artifacts in the roads. In addition, MSDDN, TRRNet, and PCTINN in Figs. 7(l)–(n) generate better results; however,


(a) Bicubic (b) BDSD-PC [6] (c) MF [11] (d) HPM-DS [12] (e) LRTCFPan [24] (f) MSDCNN [27]

(g) DiCNN [28] (h) FusionNet [13] (i) GPPNN [39] (j) MD^{3}Net [35] (k) MDA-Net [44] (l) MSDDN [32]

(m) TRRNet [33] (n) PCTINN [34] (o) UTeRM-CS (p) UTeRM-MRA (q) UTeRM-CNN (r) PAN image

Fig. 8. Comparison of the results for the 16th image of the WorldView-4 dataset in the FR experiment. The proposed UTeRMs in (o)–(q) provide results with clearer and sharper textures on the buildings.

TABLE VII
QUANTITATIVE EVALUATION OF THE FULL-RESOLUTION TEST ON THE WORLDVIEW-4 DATASET

	D_λ (↓)	D_S (↓)	QNR (↑)
BDSD-PC [6]	0.0178	0.0810	0.9027
MF [11]	0.0628	0.0849	0.8576
HPM-DS [12]	0.0452	0.0731	0.8849
LRTCFPan [24]	0.0178	**0.0313**	**0.9514**
MSDCNN [27]	0.0292	0.0452	0.9269
DiCNN [28]	**0.0116**	0.0528	0.9364
FusionNet [13]	0.0252	0.0472	0.9289
GPPNN [39]	0.0356	0.0649	0.9019
MD^{3}Net [35]	0.0263	0.0580	0.9173
MDA-Net [44]	0.0412	0.0474	0.9135
MSDDN [32]	0.0256	0.0590	0.9170
TRRNet [33]	0.0474	0.0720	0.8842
PCTINN [34]	0.0267	0.0623	0.9127
UTeRM-CS	0.0274	0.0518	0.9222
UTeRM-MRA	0.0379	0.0516	0.9124
UTeRM-CNN	0.0274	0.0531	0.9209

spatial distortions persist, noticeable in the thin and blurry textures of the roads. In contrast, the proposed UTeRMs in Figs. 7(o)–(q) provide results containing fine texture details without visible artifacts.

3) WorldView-4 (4-band Sensor): As similarly done in Section IV-B3, we evaluate the generalizability of the proposed UTeRM using test MS images from categories that are different from those used for training on the WorldView-4 dataset. Table VII quantitatively compares the pansharpening performance using non-reference metrics. The results exhibit tendencies similar to those in Table III. Note that, because ground truth is unavailable, the scores are computed by considering LRMS and PAN images as references. Thus, they may not measure the quality of pansharpened images [62] accurately; for example, blurry textures or low inter-band correlation may increase the scores. In particular, although LRTCFPan achieves the best overall performance and DiCNN the second-best, their results contain blurry artifacts, as shown in Fig. 8(e) and (g), respectively, thus degrading the image quality. Nevertheless, the proposed UTeRM-CS outperforms most DL-based algorithms—DiCNN, GPPNN, MD^{3}Net, MSDDN, TRRNet, and PCTINN on D_S and GPPNN, MD^{3}Net, MDA-Net, MSDDN, and TRRNet on QNR—while achieving results that are comparable to those of the best algorithms.

Fig. 8 compares the pansharpened HRMS images of the 16th image in the WorldView-4 test set. In Figs. 8(b)–(e), the model-based algorithms yield blurry textures, *e.g.*, the edges of trees and buildings in the yellow and red rectangles, respectively, as they fail to effectively inject details from the PAN images. MSDCNN in Fig. 8(f) severely degrades textures, while DiCNN, FusionNet, GPPNN, and MD^{3}Net in Figs. 8(g)–(j) lose fine details due to blurring artifacts. The result of TRRNet exhibits spectral distortion, *i.e.*, color change, in the yellow rectangle in Fig. 8(m). In contrast, the proposed UTeRMs in Figs. 8(o)–(q) faithfully retain the fine details in the PAN image without visible spectral distortion compared to

TABLE VIII
EFFECTS OF LOW-RANK (LR), LEARNABLE REGULARIZER (R), AND DETAIL INJECTION (DI) ON PANSHARPENING PERFORMANCE

LR	R	DI	PSNR (↑)	SSIM (↑)	SAM (↓)	SCC (↑)	ERGAS (↓)	$Q4$ (↑)
✓	✓		23.30	0.6689	10.1465	0.0029	14.9680	0.6872
✓		✓	34.20	0.9166	3.4323	0.9520	2.9455	0.6970
	✓	✓	42.18	0.9587	1.7604	0.9669	1.3286	0.8792
✓	✓	✓	**42.78**	**0.9630**	**1.6514**	**0.9708**	**1.2564**	**0.8874**

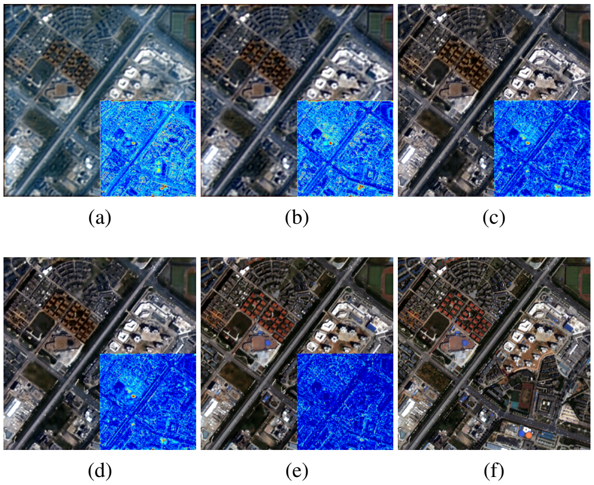


Fig. 9. Pansharpened results at intermediate stages. The pansharpened results at stages (a) $k = 2$, (b) $k = 4$, (c) $k = 6$, (d) $k = 8$, (e) $k = 10$ (final), and (f) ground-truth. The lower right corner of each image displays the corresponding error map.

the other state-of-the-art algorithms, which reveals its superior generalizability.

D. Ablation Studies

We conduct several ablation studies to analyze the effects of components in the proposed algorithm on performance. All the experiments are performed on the RR IKONOS dataset using the CNN-based detail injection (UTeRM-CNN).

1) Components in the Optimization Model: The proposed UTeRM is developed based on the tensor rank minimization model in (6), which comprises the low-rank term $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}\|_F^2$, tlearnable regularizer $f_{\text{reg}}(\boldsymbol{\mathcal{X}})$, and detail injection term $f_{\text{detail}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \mathbf{P})$. We analyze the contribution of each term to the pansharpening performance by training the network with different combinations. Table VIII compares the results. Without the detail injection term, the proposed UTeRM yields the worst performance due to lack of information from the PAN images. Without the regularizer, the deep unfolded algorithm becomes a traditional iterative algorithm; however, its performance is inferior due to an insufficient number of iterations and modeling inaccuracies. If only the regularizer and detail injection term are used, the proposed UTeRM is a regularized FusionNet, which outperforms FusionNet because of the regularization. These results confirm that all three components contribute to the superiority of the proposed algorithm, especially detail injection and learnable regularizer.

TABLE IX
EFFECTS OF LOSS FUNCTIONS ON PANSHARPENING PERFORMANCE

	PSNR (↑)	SSIM (↑)	SAM (↓)	SCC (↑)	ERGAS (↓)	$Q4$ (↑)
ℓ_1	**42.78**	**0.9630**	**1.6514**	**0.9708**	**1.2564**	**0.8874**
ℓ_2	42.54	0.9620	1.6965	0.9691	1.2732	0.8869

TABLE X
EFFECTS OF ω IN LOSS FUNCTION ON PANSHARPENING PERFORMANCE

ω	PSNR (↑)	SSIM (↑)	SAM (↓)	SCC (↑)	ERGAS (↓)	$Q4$ (↑)
0	42.46	0.9619	1.6935	0.9691	1.2842	0.8808
0.25	**42.78**	**0.9630**	**1.6514**	**0.9708**	**1.2564**	0.8874
0.50	42.74	0.9628	1.6574	0.9706	1.2601	**0.8876**
0.75	42.68	0.9623	1.6709	0.9702	1.2689	0.8846
1	42.68	0.9625	1.6667	0.9702	1.2645	0.8857

2) Interpretability: Because the proposed UTeRM is developed based on mathematical formulation, *i.e.*, tensor rank minimization, the proposed network inherits the interpretability of the theoretical model and its optimal solution. We analyze the interpretability of the proposed UTeRM by visualizing intermediate results at selected stages. Specifically, Fig. 9 shows the pansharpened results of a test image in the IKONOS dataset at every two stages. As the stage advances, the details in the pansharpened HRMS gradually become finer, resulting in a progressive reduction of errors. These intermediate results demonstrate that the behavior of UTeRM follows that of the optimization model, wherein the results are refined throughout the iterations.

3) Loss Functions: Conventional DL-based pansharpening algorithms employ either the ℓ_1- or ℓ_2-norm loss functions for training. To analyze the effects of loss functions on the performance of the proposed algorithm, we train the network using different loss functions. Table IX quantitatively compares the results. The ℓ_1-norm provides superior pansharpened results for all metrics. This is because the ℓ_1-norm encourages the network to yield sharper images, resulting in better spectral and spatial fidelity.

Next, we analyze the impacts of the two losses on the fidelity of the HRMS image and detail injection in (34) by training the network with different values of ω. Table X compares the results. When $\omega = 0$, *i.e.*, when only fidelity is considered, the worst performance is obtained. As ω increases, the performance improves because more details are injected into the output HRMS image. However, increasing ω too much degrades the performance because it reduces the relative importance of the low-rank term, thereby compromising the low-rank property of the pansharpened HRMS image. Therefore, we choose $\omega = 0.25$ to achieve overall the best performance.

4) Architecture of the CNN-Based Proximal Operator: We perform an in-depth analysis to assess the effects of the architecture of the CNN-based proximal operator $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}(\,\cdot\,; \boldsymbol{\Theta}_{\boldsymbol{\mathcal{V}}})$ in (32) on the pansharpening performance. To this end, we train the proposed algorithm with different numbers of convolutional layers in the RDB in $f_{\boldsymbol{\mathcal{V}}\text{-CNN}}(\,\cdot\,; \boldsymbol{\Theta}_{\boldsymbol{\mathcal{V}}})$. Table XI compares the pansharpening performance under different settings. As the number of convolutional layers increases, the

TABLE XI
EFFECTS OF THE NUMBER OF CONVOLUTIONAL LAYERS IN THE RDB IN $f_{\mathcal{V}\text{-CNN}}$ ON PANSHARPENING PERFORMANCE

#	PSNR (↑)	SSIM (↑)	SAM (↓)	SCC (↑)	ERGAS (↓)	$Q4$ (↑)
6	42.67	0.9621	1.6711	0.9702	1.2687	0.8851
7	42.69	0.9622	1.6679	0.9702	1.2661	0.8848
8	42.78	0.9630	1.6514	0.9708	1.2564	**0.8874**
9	**42.83**	0.9633	**1.6443**	**0.9711**	1.2525	0.8866
10	42.82	**0.9635**	1.6444	**0.9711**	**1.2524**	0.8869

TABLE XII
EFFECTS OF THE NUMBER OF UNFOLDED ITERATIONS K ON PANSHARPENING PERFORMANCE

K	PSNR (↑)	SSIM (↑)	SAM (↓)	SCC (↑)	ERGAS (↓)	$Q4$ (↑)
5	42.50	0.9606	1.7051	0.9690	1.2894	0.8826
10	42.78	0.9630	1.6514	0.9708	1.2564	0.8874
15	**42.85**	**0.9636**	**1.6373**	**0.9712**	**1.2478**	0.8875
20	42.80	0.9632	1.6446	0.9709	1.2550	**0.8886**

TABLE XIII
COMPARISON OF RUNTIMES AND NUMBERS OF PARAMETERS

	Time (ms)	# Param. (K)
MF [11]	19.93	-
BDSD-PC [6]	17.78	-
HPM-DS [12]	29.75	-
LRTCFPan [24]	12 638.13	-
MSDCNN [27]	24.76	190
DiCNN [28]	15.49	42
FusionNet [13]	15.68	76
GPPNN [39]	16.01	120
MD^{3}Net [35]	16.04	152
MDA-Net [44]	91.90	11 979
MSDDN [32]	16.73	670
TRRNet [33]	46.68	2116
PCTINN [34]	54.41	63
UTeRM-CS	59.64	4310
UTeRM-MRA	60.82	4310
UTeRM-CNN	59.45	4386

performance improves because of the increased amount of learned features. However, an excessive increase in the number of convolutional layers causes the performance to degrade because it becomes more difficult for the training process to reach convergence while increasing the number of network parameters. Therefore, for the best performance-complexity trade-off, we use an RDB with eight convolutional layers to establish $f_{\mathcal{V}\text{-CNN}}(\,\cdot\,;\boldsymbol{\Theta}_{\mathcal{V}})$.

5) Unfolded Iteration: To analyze the effects of the number of unfolded iterations, or equivalent stages, K on the pansharpening performance, we train the network with different values of K. Table XII compares the results. The proposed algorithm produces the worst pansharpened results when $K = 5$ because the CNN-based proximal operator learns insufficient information. As K increases, the number of learned features of $f_{\mathcal{V}\text{-CNN}}$ also increases, and tensor rank minimization achieves better convergence, improving the pansharpening performance. However, an excessive increase in the number of unfolded iterations negatively affects performance due to the higher number of trainable parameters, similar to the increase in the number of convolutional layers in the RDB in $f_{\mathcal{V}\text{-CNN}}$. Therefore, we select $K = 10$ to achieve the best balance between performance and computational complexity.

E. Computational Complexity

We analyze the computational complexities in terms of the number of network parameters and average runtimes of the proposed and state-of-the-art algorithms for processing 60 images from the test set of the IKONOS dataset in the RR experiment. In this test, we use a computer with an AMD Ryzen CPU clocked at 3.8 GHz and an Nvidia RTX3090 GPU. Table XIII compares the results. The proposed UTeRM is significantly more efficient than LRTCFPan [24], an iterative low-rank tensor-based algorithm. This is because LRTCFPan requires the computationally expensive singular value decomposition to compute the convex surrogate for the tensor rank. In contrast, the proposed formulation avoids the need for it. Except for MDA-Net, which has the most complicated architecture, the DL-based algorithms are more efficient than the proposed UTeRM because UTeRM uses the Fourier transform in (28) and (29) in each block, which is slower than convolution operations. In addition, despite their larger model sizes, the runtimes of UTeRM are comparable to those of the transformer-based algorithms, TRRNet and PCTINN. However, it should be noted that the proposed UTeRM yields the best performance, as discussed in Sections IV-B and IV-C. In addition, $f_{\mathcal{V}\text{-CNN}}$ can be implemented using any network as discussed in Section III-D; therefore, more effective and less complex architectures can be employed to reduce complexity.

V. CONCLUSION

We proposed a deep unfolding algorithm for pansharpening that exploits the low-rankness of MS images. First, we formulated the pansharpening task as a tensor rank minimization problem with a generalized detail injection formulation to exploit the details from PAN images. Furthermore, we defined an implicit regularization function to compensate for the potential modeling inaccuracies of low-rankness. Then, we solve the tensor rank minimization problem using an iterative technique. Finally, we unfolded the iterative algorithm into a multistage deep network, where the optimization variables and a regularizer were solved using closed-form solutions and learned CNN, respectively. In contrast to existing low-rank-based algorithms, the proposed tensor rank minimization formulation can preserve the multidimensional structures of MS images more effectively without relying on observed entries. Moreover, the proposed detail injection component can be flexibly implemented because of its generalized design. Experimental results on various MS image datasets demonstrated that the proposed algorithm achieves state-of-the-art pansharpening performance.

REFERENCES

[1] X. Meng, Y. Xiong, F. Shao, H. Shen, W. Sun, G. Yang, Q. Yuan, R. Fu, and H. Zhang, "A large-scale benchmark data set for evaluat-

ing pansharpening performance: Overview and implementation," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 1, pp. 18–52, Mar. 2021.

[2] G. A. Shaw and H. K. Burke, "Spectral imaging for remote sensing," *Lincoln Lab. J.*, vol. 14, no. 1, pp. 3–28, 2003.

[3] P. S. Chavez, Jr. and A. Y. Kwarteng, "Extracting spectral contrast in landsat thematic mapper image data using selective principal component analysis," *Photogramm. Eng. Remote Sens.*, vol. 55, no. 3, pp. 339–348, Mar. 1989.

[4] B. Aiazzi, S. Baronti, and M. Selva, "Improving component substitution pansharpening through multivariate regression of MS+Pan data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3230–3239, Oct. 2007.

[5] J. Choi, K. Yu, and Y. Kim, "A new adaptive component-substitution-based satellite image fusion by using partial replacement," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 1, pp. 295–309, Jan. 2011.

[6] G. Vivone, "Robust band-dependent spatial-detail approaches for panchromatic sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6421–6433, Sep. 2019.

[7] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, Apr. 1983.

[8] J. G. Liu, "Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details," *Int. J. Remote Sens.*, vol. 21, no. 18, pp. 3461–3472, 2000.

[9] M. J. Shensa, "The discrete wavelet transform: Wedding the à trous and Mallat algorithms," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.

[10] X. Otazu, M. González-Audícana, O. Fors, and J. Núñez, "Introduction of sensor spectral response into image fusion methods. Application to wavelet-based methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 10, pp. 2376–2385, Oct. 2005.

[11] R. Restaino, G. Vivone, M. Dalla Mura, and J. Chanussot, "Fusion of multispectral and panchromatic images based on morphological operators," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2882–2895, Jun. 2016.

[12] P. Wang, H. Yao, C. Li, G. Zhang, and H. Leung, "Multiresolution analysis based on dual-scale regression for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5406319.

[13] L.-J. Deng, G. Vivone, C. Jin, and J. Chanussot, "Detail injection-based deep convolutional neural networks for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6995–7010, Aug. 2021.

[14] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "A new pansharpening algorithm based on total variation," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 318–322, Jan. 2014.

[15] Y. Jiang, X. Ding, D. Zeng, Y. Huang, and J. Paisley, "Pan-sharpening with a hyper-Laplacian penalty," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 540–548.

[16] L. Zhang, H. Shen, W. Gong, and H. Zhang, "Adjustable model-based fusion method for multispectral and panchromatic images," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 6, pp. 1693–1704, Dec. 2012.

[17] D. Zeng, Y. Hu, Y. Huang, Z. Xu, and X. Ding, "Pan-sharpening with structural consistency and $\ell_{1/2}$ gradient prior," *Remote Sens. Lett.*, vol. 7, no. 12, pp. 1170–1179, Sep. 2016.

[18] Z.-C. Wu, T.-Z. Huang, L.-J. Deng, J.-F. Hu, and G. Vivone, "VO+Net: An adaptive approach using variational optimization and deep learning for panchromatic sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5401016.

[19] K. Rong, L. Jiao, S. Wang, and F. Liu, "Pansharpening based on low-rank and sparse decomposition," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 12, pp. 4793–4805, Dec. 2014.

[20] P. Liu, L. Xiao, and T. Li, "A variational pan-sharpening method based on spatial fractional-order geometry and spectral–spatial low-rank priors," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1788–1802, Mar. 2018.

[21] Y. Yang, C. Wan, S. Huang, H. Lu, and W. Wan, "Pansharpening based on low-rank fuzzy fusion and detail supplement," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5466–5479, Sep. 2020.

[22] F. Zhang, H. Zhang, K. Zhang, Y. Xing, J. Sun, and Q. Wu, "Exploiting low-rank and sparse properties in strided convolution matrix for pansharpening," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2649–2661, Feb. 2021.

[23] P. Liu, "Pansharpening with spatial Hessian non-convex sparse and spectral gradient low rank priors," *IEEE Trans. Image Process.*, vol. 32, pp. 2120–2131, Apr. 2023.

[24] Z.-C. Wu, T.-Z. Huang, L.-J. Deng, J. Huang, J. Chanussot, and G. Vivone, "LRTCFPan: Low-rank tensor completion based framework for pansharpening," *IEEE Trans. Image Process.*, vol. 32, pp. 1640–1655, Feb. 2023.

[25] Y. Zhang, C. Liu, M. Sun, and Y. Ou, "Pan-sharpening using an efficient bidirectional pyramid network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5549–5563, Aug. 2019.

[26] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, and J. Paisley, "PanNet: A deep network architecture for pan-sharpening," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1753–1761.

[27] Q. Yuan, Y. Wei, X. Meng, H. Shen, and L. Zhang, "A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 978–989, Mar. 2018.

[28] L. He, Y. Rao, J. Li, J. Chanussot, A. Plaza, J. Zhu, and B. Li, "Pan-sharpening via detail injection based convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1188–1204, Apr. 2019.

[29] Q. Liu, H. Zhou, Q. Xu, X. Liu, and Y. Wang, "PSGAN: A generative adversarial network for remote sensing image pan-sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10 227–10 242, Dec. 2021.

[30] Y. Li, Y. Zheng, J. Li, R. Song, and J. Chanussot, "Hyperspectral pansharpening with adaptive feature modulation-based detail injection network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5538117.

[31] M. Ciotola, G. Poggi, and G. Scarpa, "Unsupervised deep learning-based pansharpening with jointly enhanced spectral and spatial fidelity," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5405417.

[32] X. He, K. Yan, J. Zhang, R. Li, C. Xie, M. Zhou, and D. Hong, "Multiscale dual-domain guidance network for pan-sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5403213.

[33] K. Zhang, Z. Li, F. Zhang, W. Wan, and J. Sun, "Pan-sharpening based on transformer with redundancy reduction," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 5513205.

[34] M. Zhou, J. Huang, Y. Fang, X. Fu, and A. Liu, "Pan-sharpening with customized transformer and invertible neural network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 3, Jun. 2022, pp. 3553–3561.

[35] Y. Yan, J. Liu, S. Xu, Y. Wang, and X. Cao, "MD^{3}Net: Integrating model-driven and data-driven approaches for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5411116.

[36] Z. Xiang, L. Xiao, J. Yang, W. Liao, and W. Philips, "Detail-injection-model-inspired deep fusion network for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5411315.

[37] X. Cao, X. Fu, D. Hong, Z. Xu, and D. Meng, "PanCSC-Net: A model-driven deep unfolding method for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5404713.

[38] Z. Li, J. Li, F. Zhang, and L. Fan, "CADUI: Cross-attention-based depth unfolding iteration network for pansharpening remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5402420.

[39] S. Xu, J. Zhang, Z. Zhao, K. Sun, J. Liu, and C. Zhang, "Deep gradient projection networks for pan-sharpening," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 1366–1375.

[40] G. Yang, M. Zhou, K. Yan, A. Liu, X. Fu, and F. Wang, "Memory-augmented deep conditional unfolding network for pansharpening," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 1778–1787.

[41] J. Yang, L. Xiao, Y.-Q. Zhao, and J. C.-W. Chan, "Variational regularization network with attentive deep prior for hyperspectral–multispectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5508817.

[42] 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.

[43] Y. Qu, R. K. Baghbaderani, H. Qi, and C. Kwan, "Unsupervised pansharpening based on self-attention mechanism," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3192–3208, Apr. 2021.

[44] P. Guan and E. Y. Lam, "Multistage dual-attention guided fusion network for hyperspectral pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5515214.

[45] 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.

[46] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

[47] C. Hillar and L. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, Nov. 2013, Art. no. 45.

[48] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2018.

[49] L.-j. Deng, G. Vivone, M. E. Paoletti, G. Scarpa, J. He, Y. Zhang, J. Chanussot, and A. Plaza, "Machine learning in pansharpening: A benchmark, from shallow to deep networks," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 3, pp. 279–315, Sep. 2022.

[50] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," University of Illinois, Urbana-Champaign, Tech. Rep. UILU-ENG-09-2215, Nov. 2009.

[51] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[52] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, May 2015.

[54] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogramm. Eng. Remote Sens.*, vol. 63, no. 6, pp. 691–699, Jun. 1997.

[55] G. Vivone, M. Dalla Mura, A. Garzelli, and F. Pacifici, "A benchmarking protocol for pansharpening: Dataset, preprocessing, and quality assessment," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6102–6118, Jun. 2021.

[56] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[57] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm," in *Proc. Summaries Annu. JPL Airborne Geosci. Workshop*, Jun. 1992, pp. 147–149.

[58] J. Zhou, D. L. Civco, and J. A. Silander, "A wavelet transform method to merge Landsat TM and SPOT panchromatic data," *Int. J. Remote Sens.*, vol. 19, no. 4, pp. 743–757, 1998.

[59] A. Garzelli and F. Nencini, "Hypercomplex quality assessment of multi/hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 662–665, Oct. 2009.

[60] L. Alparone, B. Aiazzi, S. Baronti, A. Garzelli, F. Nencini, and M. Selva, "Multispectral and panchromatic data fusion assessment without reference," *Photogramm. Eng. Remote Sens.*, vol. 74, no. 2, pp. 193–200, Feb. 2008.

[61] G. Vivone, L. Alparone, J. Chanussot, M. Dalla Mura, A. Garzelli, G. A. Licciardi, R. Restaino, and L. Wald, "A critical comparison among pansharpening algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2565–2586, May 2015.

[62] F. Palsson, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Quantitative quality evaluation of pansharpened imagery: Consistency versus synthesis," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1247–1259, Mar. 2016.