Proximal PanNet: A Model-Based Deep Network for Pansharpening

Xiangyong Cao[1,2], **Yang Chen**[3], **Wenfei Cao**[3*]
[1] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China.
[2] Ministry of Education Key Lab For Intelligent Networks and Network Security, Xi'an 710049, China.
[3] School of Mathematics and Statistics, Shaanxi Normal University, Xi'an 710119, China.
caoxiangyong@mail.xjtu.edu.cn, yangchen9103@163.com, caowenf2015@snnu.edu.cn

Abstract

Recently, deep learning techniques have been extensively studied for pansharpening, which aims to generate a high resolution multispectral (HRMS) image by fusing a low resolution multispectral (LRMS) image with a high resolution panchromatic (PAN) image. However, existing deep learning-based pansharpening methods directly learn the mapping from LRMS and PAN to HRMS. These network architectures always lack sufficient interpretability, which limits further performance improvements. To alleviate this issue, we propose a novel deep network for pansharpening by combining the model-based methodology with the deep learning method. Firstly, we build an observation model for pansharpening using the convolutional sparse coding (CSC) technique and design a proximal gradient algorithm to solve this model. Secondly, we unfold the iterative algorithm into a deep network, dubbed as *Proximal PanNet*, by learning the proximal operators using convolutional neural networks. Finally, all the learnable modules can be automatically learned in an end-to-end manner. Experimental results on some benchmark datasets show that our network performs better than other advanced methods both quantitatively and qualitatively.

Introduction

Pansharpening is an important task for remote sensing image processing. It aims at obtaining a HRMS image by fusing a PAN image with a LRMS image as shown in Figure 1. With the satellite technology developing rapidly, the PAN and LRMS images can be easily acquired simultaneously by sensors equipped on the satellites, which requires handling the pansharpening task efficiently. Recently, a large number of approaches have been proposed, which can be roughly divided into four categories, i.e., component substitution (CS) approaches, multiresolution analysis (MRA) methods, variational optimization (VO) techniques, and deep learning (DL) approaches (Vivone et al. 2014).

The CS-based approaches rely on the concept of the projection of the LRMS image into a new domain, where the spatial information can be easily separated into a component, usually called the intensity component. Then, the PAN image can be substituted with the intensity component. Representative methods include principal component analysis (PCA) (Shah, Younan, and King 2008), intensity–hue-saturation (Carper, Lillesand, and Kiefer 1990), and Gram–Schmidt spectral sharpening appoach (Laben and Brower 2000). The CS-based methods are easy to implement and can approximate the resolution of the HRMS image in the spatial domain but result in spectral distortion (Vivone et al. 2014). To alleviate this issue, some improved methods have been presented, such as the spatial-detail approach with the band-dependent property (BDSD) (Garzelli, Nencini, and Capobianco 2007), the robust version of BDSD approach (Vivone 2019), and the adaptive CS approach with partial replacement (Choi, Yu, and Kim 2010).

The MRA-based methods depend on injecting the spatial details captured by decomposing the PAN image in a multiresolution way to the LRMS image. This type of method can preserve spectral information but bring in spatial distortion. Typical methods include the intensity modulation with smooth filter (Liu 2000), the additive wavelet luminance proportional (AWLP) (Otazu et al. 2005), the Laplacian pyramid-based methods (Restaino et al. 2019; Vivone, Marano, and Chanussot 2020; Vivone, Restaino, and Chanussot 2018).

The VO-based techniques are adopted by formulating pansharpening as a variational optimization problem, which contains elegant mathematical derivation but often suffers from a high computational burden. Typical approaches include the Bayesian methods (Wang et al. 2018), variational approaches (Fang et al. 2013a,b; Deng et al. 2018; Deng, Feng, and Tai 2019; Fu et al. 2019; Tian et al. 2021), and compressed sensing approaches (Li and Yang 2010).

Recently, the DL-based approaches have been extensively studied, and this type of method directly learns the mapping from LRMS and PAN to HRMS by some stacked network architectures. For example, Huang et al. (Huang et al. 2015) firstly proposed a stacked modified sparse denoising autoencoder (S-MSDA) to train the relationship between high-resolution (HR) and low-resolution (LR) image patches. Masi et al. (Masi et al. 2016) designed a simple and effective three-layer convolutional neural network (CNN) for the pansharpening problem. Further, a deeper network based on the ResNet is built in (Yang et al. 2017; Fu et al. 2020) by focusing on the spectral and spatial preservations. A multiple-scale and multiple-depth CNN is proposed in (Yuan et al.

[*] Corresponding author.

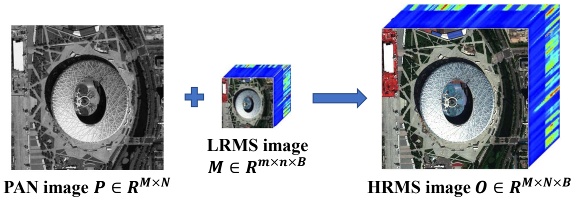
PAN image $P \in R^{M \times N}$
LRMS image
$M \in R^{m \times n \times B}$
HRMS image $O \in R^{M \times N \times B}$

Figure 1: The schematic diagram of pansharpening.

2018). Later, He et al. (He et al. 2019) proposed a network by combining the detail injection strategy with the CNN. Lately, Xie et al. (Xie et al. 2020b) designed a generative adversarial network (GAN) based network. Zhou et al. (Zhou, Liu, and Wang 2021) proposed an unsupervised generative multiadversarial network. Similarly, Deng et al. (Deng et al. 2020) devised a detailed injection-based network by deriving from the CS and MRA frameworks. More recently, Yang et al. (Yang et al. 2021) proposed a dual-stream CNN with residual information enhancement. Zhang et al. (Zhang and Ma 2021) developed a residual learning network based on gradient transformation prior. However, the network architectures of these DL-based approaches are empirically designed by stacking some network modules, and thus the entire architectures can be seen as black-box mechanisms, and the role of different modules in these networks is difficult to analyze and understand. To alleviate this issue, some attempts by combining the model-based methodology with the deep learning method have been made. For example, Xie et al. (Xie et al. 2020a) designed an interpretable deep network for multispectral and hyperspectral image fusion. Dong et al. (Dong et al. 2021) proposed a model-guided deep network for hyperspectral image super-resolution. Yin et al. (Yin 2021) and Cao et al. (Cao et al. 2021) proposed two model-driven deep networks for pansharpening.

In this paper, we keep exploring in this direction and propose a novel deep network, dubbed as *Proximal PanNet*, for the pansharpening problem by combining the model-based methodology with the deep learning method. Firstly, we propose a new observation model for pansharpening using the convolutional sparse coding (CSC) technique. Specifically, we model the PAN and MS image separately using the CSC with two types of features, i.e., common features and unique features. The two features are based on two observations. One is that the PAN and MS images capture the same scene, they thus share some common features, and the other one is that the two images are generated by different sensors equipped on the satellite, they thus contain their unique features. Therefore, to generate a better fused HRMS image, we need to capture the common and unique features from the two images, and then reconstruct the HRMS image using both features. Secondly, we reformulate the observation model into an optimization problem and add implicit regularizations modeling priors on both common and unique features. Thirdly, we design a proximal gradient algorithm to solve this optimization model, and unfold the iterative algorithm into a deep network, where each module corresponds to a specific operation of the iterative algorithm. The proximal operators are approximated by the convolutional neural network. Therefore, each module of the unfolded network can be clearly interpreted. Finally, all the learnable modules can be automatically learned end-to-end, and thus the hyperparameter selection can be avoided. Experimental results show the superiority of our network both quantitatively and qualitatively comparing with other state-of-the-art methods.

Proposed Model

In this section, a new pansharpening model is firstly proposed. Then, a proximal gradient algorithm is derived to optimize this model. Finally, the algorithm is further unfolded into a deep neural network.

Proposed Model for Pansharpening

The pansharpening model is proposed based on the following observations. Firstly, since both the LRMS image $\mathbf{M} \in \mathbb{R}^{m\times n\times B}$ and the PAN image $\mathbf{P} \in \mathbb{R}^{M\times N}$ are captured in the same scene, they should share some common features. Secondly, since the two images are acquired by different sensors in the satellite, each image thus has its unique feature. Additionally, we also assume that the information underlying both the LRMS image and the PAN image are beneficial to estimate the HRMS image $\mathbf{O} \in \mathbb{R}^{M\times N\times B}$. Therefore, we firstly extract these effective features from both images, and then fuse these features to generate the HRMS image $\mathbf{O}$. To extract the features from both images, we adopt the CSC technique (Deng and Dragotti 2020; Li et al. 2021) to model the PAN and LRMS images separately.

Specifically, our new model is defined as follows:

$$\mathbf{P} = \sum_{k=1}^{K} \mathbf{D}_k^c \otimes \mathbf{C}_k + \sum_{k=1}^{K} \mathbf{D}_k^u \otimes \mathbf{U}_k, \tag{1}$$

$$\tilde{\mathbf{M}} = \sum_{k=1}^{K} \mathbf{H}_k^c \otimes \mathbf{C}_k + \sum_{k=1}^{K} \mathbf{H}_k^v \otimes \mathbf{V}_k, \tag{2}$$

where $\tilde{\mathbf{M}} \in \mathbb{R}^{M\times N\times B}$ is the upsampled version of the LRMS image $\mathbf{M}$ using the polynomial kernel technique (Aiazzi et al. 2002), $\otimes$ is the convolutional operation, $\mathbf{D}_k^c \in \mathbb{R}^{s\times s}$ and $\mathbf{H}_k^c \in \mathbb{R}^{s\times s\times B}$ denote the common filter for both images; $\mathbf{D}_k^u \in \mathbb{R}^{s\times s}$ and $\mathbf{H}_k^v \in \mathbb{R}^{s\times s\times B}$ represent the separate filters of both images; $\mathbf{C}_k \in \mathbb{R}^{M\times N}$ denotes the shared common features, and $\mathbf{U}_k \in \mathbb{R}^{M\times N}$ and $\mathbf{V}_k \in \mathbb{R}^{M\times N}$ are unique features of HRMS image and LRHS image, respectively. Then, the HRMS image $\mathbf{O} \in \mathbb{R}^{M\times N\times B}$ can be generated by fusing these features as follows:

$$\mathbf{O} = \sum_{k=1}^{K} \mathbf{G}_k^c \otimes \mathbf{C}_k + \sum_{k=1}^{K} \mathbf{G}_k^u \otimes \mathbf{U}_k + \sum_{k=1}^{K} \mathbf{G}_k^v \otimes \mathbf{V}_k, \tag{3}$$

where $\mathbf{G}_k^c \in \mathbb{R}^{s\times s\times B}$ represents the common filter, $\mathbf{G}_k^u \in \mathbb{R}^{s\times s\times B}$ and $\mathbf{G}_k^v \in \mathbb{R}^{s\times s\times B}$ denote the unique filters concerning the features of both images. To make the notations clear, we denote $\mathbf{C} = \{\mathbf{C}_k\}_{k=1}^K \in \mathbb{R}^{M\times N\times K}$, $\mathbf{U} = \{\mathbf{U}_k\}_{k=1}^K \in \mathbb{R}^{M\times N\times K}$, and $\mathbf{V} = \{\mathbf{V}_k\}_{k=1}^K \in \mathbb{R}^{M\times N\times K}$.

In the following, we assume that all the filters are known. Therefore, to obtain the fused HRMS image $\mathbf{O}$, we need to calculate $\mathbf{C}$, $\mathbf{U}$, and $\mathbf{V}$ as follows:

$$\begin{aligned}\min_{\mathbf{C},\mathbf{U},\mathbf{V}} \frac{1}{2}||\mathbf{P} - \sum_{k=1}^{K} \mathbf{D}_k^c \otimes \mathbf{C}_k - \sum_{k=1}^{K} \mathbf{D}_k^u \otimes \mathbf{U}_k||_F^2 \\ + \frac{1}{2}||\tilde{\mathbf{M}} - \sum_{k=1}^{K} \mathbf{H}_k^c \otimes \mathbf{C}_k - \sum_{k=1}^{K} \mathbf{H}_k^v \otimes \mathbf{V}_k||_F^2 \\ + \lambda_1 f_1(\mathbf{U}) + \lambda_2 f_2(\mathbf{V}) + \lambda_3 f_3(\mathbf{C}),\end{aligned} \tag{4}$$

where λ_1, λ_2, and λ_3 are the trade-off parameters. $f_1(\cdot), f_2(\cdot)$, and $f_3(\cdot)$ are the regularization terms modeling the priors on $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{C}$, respectively.

Model Optimization

To solve the problem, we adopt the alternate optimization algorithm to alternately update each variable with other variables fixed, which leads to the three sub-problems:

$$\begin{aligned}\mathbf{U}^{(t)} = \arg\min_{\mathbf{U}} ||\mathbf{P} - \sum_{k=1}^{K} \mathbf{D}_k^c \otimes \mathbf{C}_k^{(t-1)} - \sum_{k=1}^{K} \mathbf{D}_k^u \otimes \mathbf{U}_k||_F^2 \\ + \lambda_1 f_1(\mathbf{U}),\end{aligned} \tag{5}$$

$$\begin{aligned}\mathbf{V}^{(t)} = \arg\min_{\mathbf{V}} ||\tilde{\mathbf{M}} - \sum_{k=1}^{K} \mathbf{H}_k^c \otimes \mathbf{C}_k^{(t-1)} - \sum_{k=1}^{K} \mathbf{H}_k^v \otimes \mathbf{V}_k||_F^2 \\ + \lambda_2 f_2(\mathbf{V}),\end{aligned} \tag{6}$$

$$\begin{aligned}\mathbf{C}^{(t)} = \arg\min_{\mathbf{C}} ||\mathbf{P} - \sum_{k=1}^{K} \mathbf{D}_k^c \otimes \mathbf{C}_k - \sum_{k=1}^{K} \mathbf{D}_k^u \otimes \mathbf{U}_k^{(t)}||_F^2 \\ + ||\tilde{\mathbf{M}} - \sum_{k=1}^{K} \mathbf{H}_k^c \otimes \mathbf{C}_k - \sum_{k=1}^{K} \mathbf{H}_k^v \otimes \mathbf{V}_k^{(t)}||_F^2 + \lambda_3 f_3(\mathbf{C}).\end{aligned} \tag{7}$$

Next, we introduce the optimization for each sub-problem.

$\mathbf{U}-$subproblem Instead of directly optimizing the objective (5), we first adopt the quadratic function to approximate the objective function, which is expressed as:

$$\begin{aligned}\mathbf{U}^{(t)} = \arg\min_{\mathbf{U}} g\left(\mathbf{U}^{(t-1)}\right) + \frac{1}{2\eta_1}||\mathbf{U} - \mathbf{U}^{(t-1)}||_F^2 \\ + \left\langle \nabla g(\mathbf{U}^{(t-1)}), \mathbf{U} - \mathbf{U}^{(t-1)} \right\rangle + \lambda_1 f_1(\mathbf{U}),\end{aligned} \tag{8}$$

where

$$g\left(\mathbf{U}^{(t-1)}\right) = ||\mathbf{P} - \sum_{k=1}^{K} \mathbf{D}_k^c \otimes \mathbf{C}_k^{(t-1)} - \sum_{k=1}^{K} \mathbf{D}_k^u \otimes \mathbf{U}_k^{(t-1)}||_F^2, \tag{9}$$

$$\nabla g(\mathbf{U}^{(t-1)}) = \mathbf{D}^u \otimes^{\dagger} \left(\sum_{k=1}^{K} \mathbf{D}_k^c \otimes \mathbf{C}_k^{(t-1)} + \sum_{k=1}^{K} \mathbf{D}_k^u \otimes \mathbf{U}_k^{(t-1)} - \mathbf{P} \right), \tag{10}$$

$\nabla(\cdot)$ is gradient operator, η_1 is the step size, $\mathbf{D}^u \in \mathbb{R}^{s \times s \times K \times 1}$ is a $4-$D tensor stacked by all $\mathbf{D}_k^u$, and $\otimes^{\dagger}$ denotes the transposed convolution. By a simple mathematical derivation, Eq. (8) can be equivalent to

$$\begin{aligned}\mathbf{U}^{(t)} = \arg\min_{\mathbf{U}} \frac{1}{2}||\mathbf{U} - (\mathbf{U}^{(t-1)} - \eta_1 \nabla g(\mathbf{U}^{(t-1)}))||_F^2 \\ + \eta_1 \lambda_1 f_1(\mathbf{U}).\end{aligned} \tag{11}$$

Eq.(11) can then be optimized by the general proximal operator (Beck and Teboulle 2009), which is defined as

$$\mathbf{U}^{(t)} = prox_{\eta_1 \lambda_1} \left(\mathbf{U}^{(t-1)} - \eta_1 \nabla g(\mathbf{U}^{(t-1)}) \right), \tag{12}$$

where $prox_{\eta_1 \lambda_1}(\cdot)$ is the proximal operator related to $f_1(\cdot)$. To alleviate the limitation of human designed regularizers, we adopt the deep CNNs to automatically learn the prior from the data, which will be discussed in detail in the next subsection.

$\mathbf{V}-$subproblem For the updating of $\mathbf{V}$, we adopt a similar strategy. Firstly, the quadratic approximation of (6) with respect to $\mathbf{V}$ is defined as:

$$\begin{aligned}\mathbf{V}^{(t)} = \arg\min_{\mathbf{V}} h\left(\mathbf{V}^{(t-1)}\right) + \frac{1}{2\eta_2}||\mathbf{V} - \mathbf{V}^{(t-1)}||_F^2 \\ + \left\langle \nabla h(\mathbf{V}^{(t-1)}), \mathbf{V} - \mathbf{V}^{(t-1)} \right\rangle + \lambda_2 f_2(\mathbf{V}),\end{aligned} \tag{13}$$

where

$$h\left(\mathbf{V}^{(t-1)}\right) = ||\tilde{\mathbf{M}} - \sum_{k=1}^{K} \mathbf{H}_k^c \otimes \mathbf{C}_k^{(t-1)} - \sum_{k=1}^{K} \mathbf{H}_k^v \otimes \mathbf{V}_k^{(t-1)}||_F^2, \tag{14}$$

$$\nabla h(\mathbf{V}^{(t-1)}) = \mathbf{H}^v \otimes^{\dagger} \left(\sum_{k=1}^{K} \mathbf{H}_k^c \otimes \mathbf{C}_k^{(t-1)} + \sum_{k=1}^{K} \mathbf{H}_k^v \otimes \mathbf{V}_k^{(t-1)} - \tilde{\mathbf{M}} \right), \tag{15}$$

η_2 is the step size, and $\mathbf{H}^v \in \mathbb{R}^{s \times s \times K \times B}$ is a $4-$D tensor stacked by all $\mathbf{H}_k^v$. Further, Eq. (13) can be equivlently formulated as

$$\begin{aligned}\mathbf{V}^{(t)} = \arg\min_{\mathbf{V}} \frac{1}{2}||\mathbf{V} - (\mathbf{V}^{(t-1)} - \eta_2 \nabla h(\mathbf{V}^{(t-1)}))||_F^2 \\ + \eta_2 \lambda_2 f_2(\mathbf{V}),\end{aligned} \tag{16}$$

Then, Eq.(16) can also be solved by

$$\mathbf{V}^{(t)} = prox_{\eta_2 \lambda_2} \left(\mathbf{V}^{(t-1)} - \eta_2 \nabla h(\mathbf{V}^{(t-1)}) \right), \tag{17}$$

where $prox_{\eta_2 \lambda_2}(\cdot)$ is the proximal operator related to $f_2(\cdot)$.

$\mathbf{C}-$subproblem Eq. (7) can be re-written as:

$$\begin{aligned}\mathbf{C}^{(t)} = \arg\min_{\mathbf{C}} ||\hat{\mathbf{P}} - \sum_{k=1}^{K} \mathbf{D}_k^c \otimes \mathbf{C}_k||_F^2 \\ + ||\hat{\mathbf{M}} - \sum_{k=1}^{K} \mathbf{H}_k^c \otimes \mathbf{C}_k||_F^2 + \lambda_3 f_3(\mathbf{C}),\end{aligned} \tag{18}$$

where $\hat{\mathbf{P}} = \mathbf{P} - \sum_{k=1}^{K} \mathbf{D}_k^u \otimes \mathbf{U}_k^{(t)}$ and $\hat{\mathbf{M}} = \tilde{\mathbf{M}} - \sum_{k=1}^{K} \mathbf{H}_k^v \otimes \mathbf{V}_k^{(t)}$. Further, Eq. (18) can be expressed as:

$$\mathbf{C}^{(t)} = \arg\min_{\mathbf{C}} ||\mathbf{N} - \sum_{k=1}^{K} \mathbf{L}_k^c \otimes \mathbf{C}_k||_F^2 + \lambda_3 f_3(\mathbf{C}), \tag{19}$$

where $\mathbf{N} \in \mathbb{R}^{M \times N \times (B+1)}$ is the concatenation of $\hat{\mathbf{P}}$ and $\hat{\mathbf{M}}$, and $\mathbf{L}_k^c \in \mathbb{R}^{s \times s \times (B+1)}$ is the concatenation of $\mathbf{D}_k^c$ and $\mathbf{H}_k^c$. Eq. (19) can then be solved by the same way as optimizing

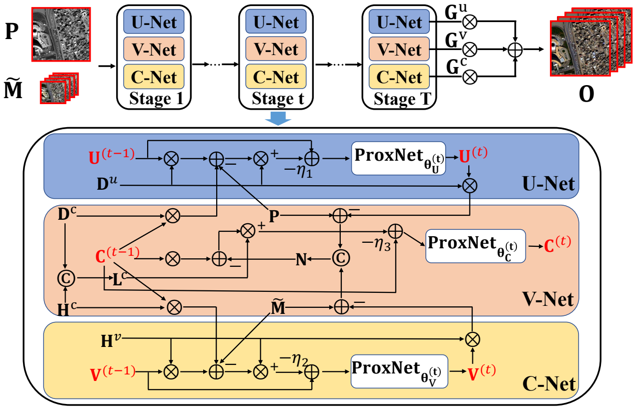

Figure 2: The overall architecture of our proposed unfolding network for pansharpening. This network is composed of T stages shown at the top, and each stage contains three sub-networks (i.e., U-Net, V-Net, and C-net). The detailed network architectures of the three sub-networks in the stage t are shown at the bottom. $\otimes$ and $\oplus$ represent the convolutional operation and the element-wise addition, respectively.

$\mathbf{U}$ and $\mathbf{V}$. More specifically, we first derive the quadratic approximation of Eq. (19) with respect to $\mathbf{C}$ as follows:

$$\mathbf{C}^{(t)} = \arg\min_{\mathbf{C}} l\left(\mathbf{C}^{(t-1)}\right) + \frac{1}{2\eta_3}||\mathbf{C}-\mathbf{C}^{(t-1)}||_F^2 + \left\langle \nabla l(\mathbf{C}^{(t-1)}), \mathbf{C}-\mathbf{C}^{(t-1)} \right\rangle + \lambda_3 f_3(\mathbf{C}), \tag{20}$$

where

$$l\left(\mathbf{C}^{(t-1)}\right) = ||\mathbf{N}-\sum_{k=1}^{K}\mathbf{L}_k^c \otimes \mathbf{C}_k^{(t-1)}||_F^2, \tag{21}$$

$$\nabla l(\mathbf{C}^{(t-1)}) = \mathbf{L}^c \otimes^{\dagger} \left(\sum_{k=1}^{K}\mathbf{L}_k^c \otimes \mathbf{C}_k^{(t-1)} - \mathbf{N}\right), \tag{22}$$

η_3 is the step size, and $\mathbf{L}^c \in \mathbb{R}^{s\times s\times K\times(B+1)}$ is a $4-$D tensor stacked by all $\mathbf{L}_k^c$. Further, Eq. (20) can be rewritten as

$$\mathbf{C}^{(t)} = \arg\min_{\mathbf{C}} \frac{1}{2}||\mathbf{C}-(\mathbf{C}^{(t-1)}-\eta_3\nabla l(\mathbf{C}^{(t-1)}))||_F^2 + \eta_3\lambda_3 f_3(\mathbf{C}). \tag{23}$$

Finally, Eq.(23) can be solved as

$$\mathbf{C}^{(t)} = prox_{\eta_3\lambda_3}\left(\mathbf{C}^{(t-1)} - \eta_3\nabla l(\mathbf{C}^{(t-1)})\right), \tag{24}$$

where $prox_{\eta_3\lambda_3}(\cdot)$ is the proximal operator related to $f_3(\cdot)$.

Proximal PanNet

Based on the iterative algorithm, we build a deep network shown in the top row of Fig. 2. This network contains T stages corresponding to T iterations of the iterative algorithm for solving Eq. (4). Each stage takes the outputs of the previous stage $\mathbf{U}^{(t-1)}$, $\mathbf{V}^{(t-1)}$ and $\mathbf{C}^{(t-1)}$ as the inputs and produces the updates $\mathbf{U}^{(t)}$, $\mathbf{V}^{(t)}$ and $\mathbf{C}^{(t)}$.

As shown in Eq. (12), (17) and (24), the three proximal operators $\mathrm{prox}_{\eta_1\lambda_1}(\cdot)$, $\mathrm{prox}_{\eta_2\lambda_2}(\cdot)$, and $\mathrm{prox}_{\eta_3\lambda_3}(\cdot)$ play a key role in the algorithm. To update each variable, classical methods usually design hand-crafted priors which make the proximal operators be closed form. However, due to the limited representation abilities, human-designed priors cannot always achieve the expected effect. Therefore, we utilize convolutional neural networks to explore prior for $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{C}$, namely learning their corresponding proximal operators $\mathrm{prox}_{\eta_1\lambda_1}(\cdot)$, $\mathrm{prox}_{\eta_2\lambda_2}(\cdot)$, and $\mathrm{prox}_{\eta_3\lambda_3}(\cdot)$ for updating $\mathbf{U}^{(t)}$, $\mathbf{V}^{(t)}$, and $\mathbf{C}^{(t)}$ in each stage t.

Network design Our network is designed as several stages, and each stage updates $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{C}$ in a cascaded manner. At stage t, $\mathbf{U}^{(t)}$ is updated using $\mathbf{P}$, $\mathbf{U}^{(t-1)}$, and $\mathbf{C}^{(t-1)}$ by the following sub-steps:

$$\mathbf{U-Net}: \begin{cases} \mathbf{P}_c^{(t-1)} = \sum_{k=1}^{K}\mathbf{D}_k^c \otimes \mathbf{C}_k^{(t-1)}, \\ \mathbf{P}_u^{(t-1)} = \sum_{k=1}^{K}\mathbf{D}_k^u \otimes \mathbf{U}_k^{(t-1)}, \\ \varepsilon_P^{(t-1)} = \mathbf{P}_c^{(t-1)} + \mathbf{P}_u^{(t-1)} - \mathbf{P}, \\ \nabla g^{(t-1)} = \mathbf{D}^u \otimes^{\dagger} \varepsilon_P^{(t-1)}, \\ \mathbf{U}^{(t-0.5)} = \mathbf{U}^{(t-1)} - \eta_1\nabla g^{(t-1)}, \\ \mathbf{U}^{(t)} = \mathrm{proxNet}_{\Theta_{\mathbf{U}}^{(t)}}\left(\mathbf{U}^{(t-0.5)}\right). \end{cases} \tag{25}$$

Each step is unfolded into one module of the U-Net shown in Figure 2. Then, $\mathbf{V}^{(t)}$ is updated by $\tilde{\mathbf{M}}$, $\mathbf{V}^{(t-1)}$, and $\mathbf{C}^{(t-1)}$:

$$\mathbf{V-Net}: \begin{cases} \mathbf{M}_c^{(t-1)} = \sum_{k=1}^{K}\mathbf{H}_k^c \otimes \mathbf{C}_k^{(t-1)}, \\ \mathbf{M}_v^{(t-1)} = \sum_{k=1}^{K}\mathbf{H}_k^v \otimes \mathbf{V}_k^{(t-1)}, \\ \varepsilon_M^{(t-1)} = \mathbf{M}_c^{(t-1)} + \mathbf{M}_v^{(t-1)} - \tilde{\mathbf{M}}, \\ \nabla h^{(t-1)} = \mathbf{H}^v \otimes^{\dagger} \varepsilon_M^{(t-1)}, \\ \mathbf{V}^{(t-0.5)} = \mathbf{V}^{(t-1)} - \eta_2\nabla h^{(t-1)}, \\ \mathbf{V}^{(t)} = \mathrm{proxNet}_{\Theta_{\mathbf{V}}^{(t)}}\left(\mathbf{V}^{(t-0.5)}\right). \end{cases} \tag{26}$$

Similarly, each step is unfolded into one module of the V-Net as shown in Figure 2. Finally, $\mathbf{C}^{(t)}$ is updated using $\mathbf{P}$, $\tilde{\mathbf{M}}$, $\mathbf{C}^{(t-1)}$, $\mathbf{U}^{(t)}$ and $\mathbf{V}^{(t)}$:

$$\mathbf{C-Net}: \begin{cases} \mathbf{F}_c^{(t-1)} = \sum_{k=1}^{K}\mathbf{L}_k^c \otimes \mathbf{C}_k^{(t-1)}, \\ \varepsilon_C^{(t-1)} = \mathbf{F}_c^{(t-1)} - \mathbf{N}, \\ \nabla l^{(t-1)} = \mathbf{L}^c \otimes^{\dagger} \varepsilon_C^{(t-1)}, \\ \nabla l^{(t-0.5)} = \mathbf{C}^{(t-1)} - \eta_3\nabla l^{(t-1)}, \\ \mathbf{C}^{(t)} = \mathrm{proxNet}_{\Theta_{\mathbf{C}}^{(t)}}\left(l^{(t-0.5)}\right). \end{cases} \tag{27}$$

Similarly, each step is unfolded into one module of the C-Net as shown in Figure 2. Additionally, $\mathrm{proxNet}_{\Theta_{\mathbf{U}}^{(t)}}(\cdot)$, $\mathrm{proxNet}_{\Theta_{\mathbf{V}}^{(t)}}(\cdot)$ and $\mathrm{proxNet}_{\Theta_{\mathbf{C}}^{(t)}}(\cdot)$ are all ResNet (He et al. 2016) with 3 ResBlocks, which are used to approximate the three proximal operators, and $\Theta_{\mathbf{U}}^{(t)}$, $\Theta_{\mathbf{V}}^{(t)}$, and $\Theta_{\mathbf{C}}^{(t)}$ are the corresponding network parameters at the t^{th} stage. Besides, $\Theta_{\mathbf{U}}^{(t)}$, $\Theta_{\mathbf{V}}^{(t)}$, and $\Theta_{\mathbf{C}}^{(t)}$ are the parameters of the three ResNets at the t^{th} stage. Note that all the network parameters can be end-to-end learned from training data, including $\{\Theta_{\mathbf{U}}^{(t)}, \Theta_{\mathbf{V}}^{(t)}, \Theta_{\mathbf{C}}^{(t)}\}_{t=1}^{T}$, $\mathbf{D}^c = \{\mathbf{D}_k^c\}_{k=1}^{K}$, $\mathbf{D}^u = \{\mathbf{D}_k^u\}_{k=1}^{K}$, $\mathbf{H}^c = \{\mathbf{H}_k^c\}_{k=1}^{K}$, $\mathbf{H}^v = \{\mathbf{H}_k^v\}_{k=1}^{K}$, $\mathbf{G}^c = \{\mathbf{G}_k^c\}_{k=1}^{K}$, $\mathbf{G}^u = \{\mathbf{G}_k^u\}_{k=1}^{K}$, $\mathbf{G}^v = \{\mathbf{G}_k^v\}_{k=1}^{K}$, η_1, η_2, and η_3. Once training finished, the testing can be shortened due to its feed-forward architecture. Note that our network shown in Figure

Methods	**Q8**	**SAM**	**ERGAS**
EXP	0.617±0.161	5.712±2.114	6.732±3.003
GS	0.810±0.130	6.773±2.831	5.035±2.385
BDSD	0.848±0.127	5.331±2.087	4.104±1.968
BDSD-PC	0.755±0.149	5.616±2.120	5.151±2.293
PRACS	0.806±0.140	5.477±2.094	4.566±1.978
AWLP	0.847±0.119	5.339±1.971	4.199±1.909
GLP	0.853±0.120	5.126±1.996	4.017±1.851
GLP-HPM	0.847±0.124	5.211±2.069	4.071±1.887
MO	0.826±0.132	5.245±2.004	4.427±1.955
SRD	0.832±0.129	5.172±2.021	4.311±1.947
PanNet	0.907±0.125	3.921±1.495	2.635±1.172
Fusion-Net	0.912±0.122	3.803±1.440	2.577±1.160
Our	**0.925±0.120**	**3.642±1.437**	**2.426±1.235**

Table 1: Results on the Worldview3 dataset (1258 images).

Methods	**Q8**	**SAM**	**ERGAS**
EXP	0.703	6.708	8.331
GS	0.815	7.206	6.826
BDSD	0.833	6.533	6.559
BDSD-PC	0.770	7.117	7.191
PRACS	0.811	6.583	6.848
AWLP	0.843	6.381	6.380
GLP	0.846	6.288	6.322
GLP-HPM	0.840	6.340	6.392
MO	0.838	6.215	6.514
SRD	0.840	6.248	6.335
PanNet	0.981	5.051	3.613
Fusion-Net	0.981	4.867	3.475
Our	**0.983**	**4.740**	**3.394**

Table 2: Results on the Tripoli dataset (Worldview-3).

Methods	D_λ	D_s	**QNR**
EXP	0.010±0.003	0.059±0.013	0.927±0.012
GS	0.027±0.007	0.068±0.019	0.905±0.022
BDSD	0.009±0.002	0.042±0.007	0.948±0.008
BDSD-PC	0.009±0.002	0.043±0.007	0.947±0.009
PRACS	0.010±0.002	0.036±0.007	0.952±0.008
AWLP	0.016±0.032	0.032±0.005	0.951±0.006
GLP	0.017±0.006	0.038±0.006	0.944±0.011
GLP-HPM	0.015±0.005	0.036±0.006	0.948±0.010
MO	0.021±0.006	0.035±0.005	0.943±0.011
SRD	0.009±0.002	0.061±0.016	0.934±0.018
PanNet	0.009±0.002	0.032±0.004	0.954±0.006
Fusion-Net	0.009±0.002	0.032±0.004	0.956±0.005
Our	**0.008±0.002**	**0.031±0.004**	**0.960±0.005**

Table 3: Results on the WorldView-3 dataset (30 samples).

2 is interpretable. For example, to update $\mathbf{U}^{(t)}$ by Eq. (25), computing $\mathbf{U}^{(t-0.5)}$ means the gradient descent, and the $\text{proxNet}_{\Theta_{\mathbf{U}}^{(t)}}(\cdot)$ performs nonlinear operation on $\mathbf{U}^{(t-0.5)}$.

Network training The loss function of our training process is defined as

$$\mathcal{L} = \sum_{j=1}^{J} ||\mathbf{O}_p^j - \mathbf{O}_g^j||_2^2, \tag{28}$$

where $\mathbf{O}_p^j$ is the predicted HRMS image of the j^{th} training sample, $\mathbf{O}_g^j$ is the true HRMS image of the j^{th} training sample, and J is the number of training pairs. In our network, all the convolutional kernel sizes are set as 8×8. The channel number of each convolution is set as 16. The stage number of our network is set as 2. We implement our network using TensorFlow on a GTX 1080Ti GPU with 12 GB memory. Additionally, we adopt the Adam algorithm, and a decayed technique to set the learning rate, i.e., decayed by 0.9 every 50 epochs with a fixed initial learning rate 0.0001. Also, the epoch number is 100, and the mini-batch size is 64.

Experiments

We conduct several experiments using data from the Worldview3 satellite. Specifically, we extract $12.5K$ image pairs from the image of Worldview3, and each image pair contains one PAN image (64×64), one MS image (16×16×8) and one ground truth HRMS image (64×64×8). We split these image pairs into 90%/10% for training/testing. The ground truth HRMS image is obtained by using the Wald's protocol (Wald, Ranchin, and Mangolini 1997) due to its unavailability. We compare with twelve state-of-the-art methods: EXP (Aiazzi et al. 2002), GS (Laben and Brower 2000), BDSD (Garzelli, Nencini, and Capobianco 2007), BDSD-PC (Vivone 2019), PRACS (Choi, Yu, and Kim 2010), AWLP (Otazu et al. 2005), GLP (Restaino et al. 2019), GLP-HPM (Vivone et al. 2013), MO (Restaino et al. 2016), SRD (Vicinanza et al. 2014), PanNet (Yang et al. 2017), and Fusion-Net (Deng et al. 2020).

The performance assessment contains two cases. In the first case where the ground-truth HRMS is known, we adopt four popular reference-based indexes (Vivone et al. 2014): the Q4 (for 4 bands) and Q8 (for 8 bands), the spectral angle mapper (SAM), and the relative dimensionless global error in synthesis (ERGAS). In the second case where the ground-truth HRMS is not available, we use D_λ, D_s, and the quality without reference (QNR) criteria (Vivone et al. 2014).

Evaluation at Reduced Resolution

We use 90% of the image pairs to train our network and the remaining ones (1258 images) for testing. Table 1 shows the average quantitative results and standard deviation of each method. From Table 1, it can be easily observed that the DL-based methods (PanNet, Fusion-Net, and our network) perform better than the CS and MRA approaches in terms of all the indexes, which may attribute to the strong nonlinear approximation ability of the deep neural network. Additionally, compared with the PanNet and Fusion-Net, our proposed network obtains better performance, which further verifies that the CSC framework has a certain advantage over the CS and MRA frameworks since the PanNet and Fusion-Net are built based on the CS and MRA frameworks while our network is built based on the proposed CSC framework. Also, our proposed network is very robust since it has almost the smallest standard deviation.

Further, we conduct another experiment on one new WorldView-3 dataset (i.e., Tripoli dataset), which has never been seen in the training phase. This dataset contains one

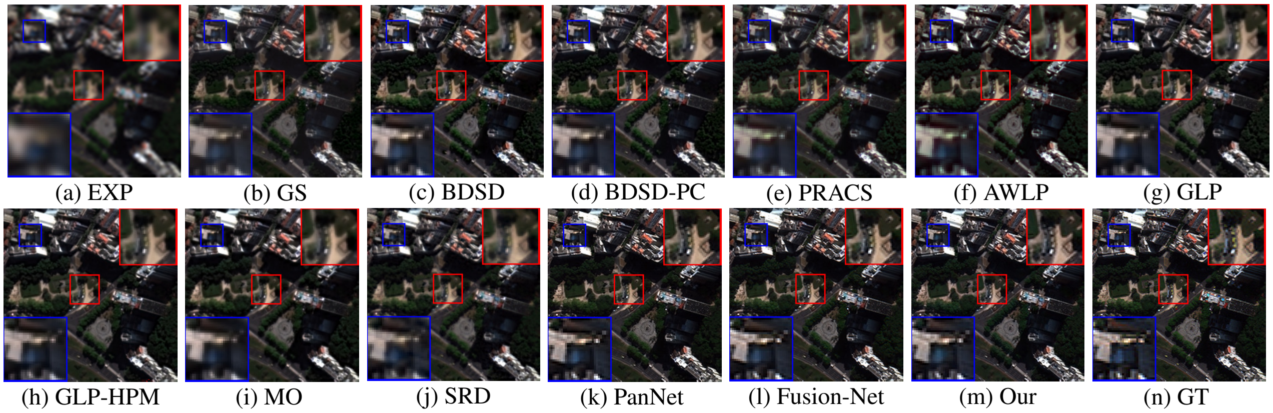

Figure 3: Visual comparisons of the fused HRMS image for all the methods on the Tripoli dataset.

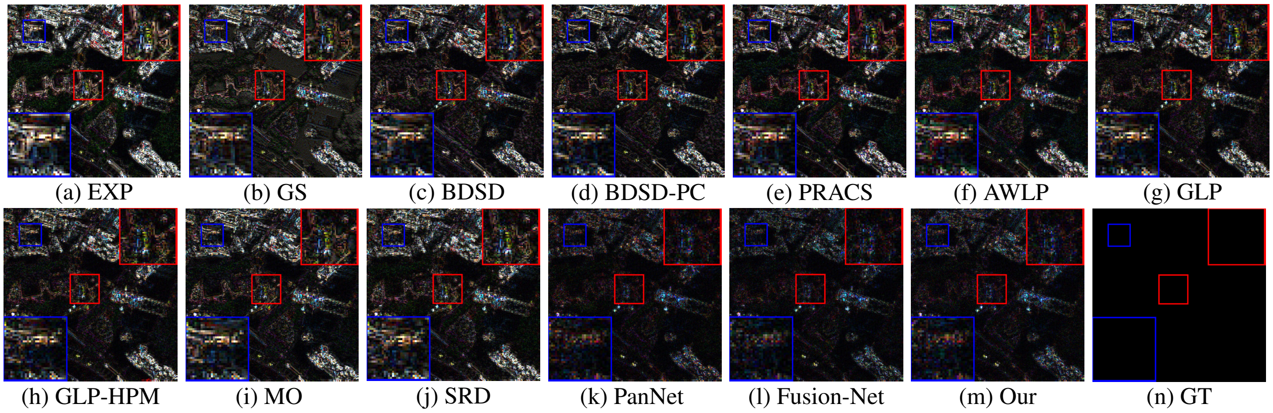

Figure 4: The absolute value of residuals between the true HRMS and the fused HRMS of Figure 3.

PAN image (256×256), one MS image (64×64×8) and one true HRMS image (256×256×8). The quantitative results are reported in Table 2, from which we can observe that our network achieves the best performance compared with other approaches. The visual comparison shown in Figure 3 further verifies the numerical results. For better illustration, we enlarge two regions of the fused image. Specifically, it can be seen from Figure 3 that the visual effect of classical methods contain obvious blur effects, while the three DL based methods produce better fused visual results. Additionally, we show the residual map of all the methods in Figure 4, from which we can observe that the residual image of our method contains fewer details.

Evaluation at Full Resolution

This section evaluates the performance of our network in the full resolution case. We conduct this experiment on 30 image pairs from the Worldview3 satellite. Each image pair contains one PAN image (256×256), and one MS image (64×64×8). Since the ground-truth HRMS image is not available, we thus adopt D_λ, D_s, and QNR (Vivone et al. 2014) as the indexes. The quantitative results are reported in Table 3. As can be seen, the DL-based methods (i.e., PanNet, Fusion-Net, and our network) perform better than other methods, which verifies the superiority of the DL methods. Among the three DL approaches, our network performs best. Additionally, we also show visual comparison of a full-resolution sample in Figure 5, which indicates that our network obtains the best visual effect both spatially and spectrally than the classical approaches, and also has a comparable visual effect with the other two DL methods.

Generalization to New Satellites

This section explores the generalization of our network on the images acquired by other satellites. Since our network is trained using the WolrdView-3 images, we thus test our network on the WorldView-2 images. We also adopt the Wald's protocol to generate 4 test image pairs, of which each contains one PAN image (256×256), one MS image (64×64×8), and one true HRMS image (256×256×8). The average quantitative results are reported in Table 4. As can be seen, our network obtains almost the best performance. These results indicate that our network performs almost the best in both spatial and spectral domain, which implies that our network can generalize well to new satelites. Besides, we show the visual comparisons in Figure 6, implying that

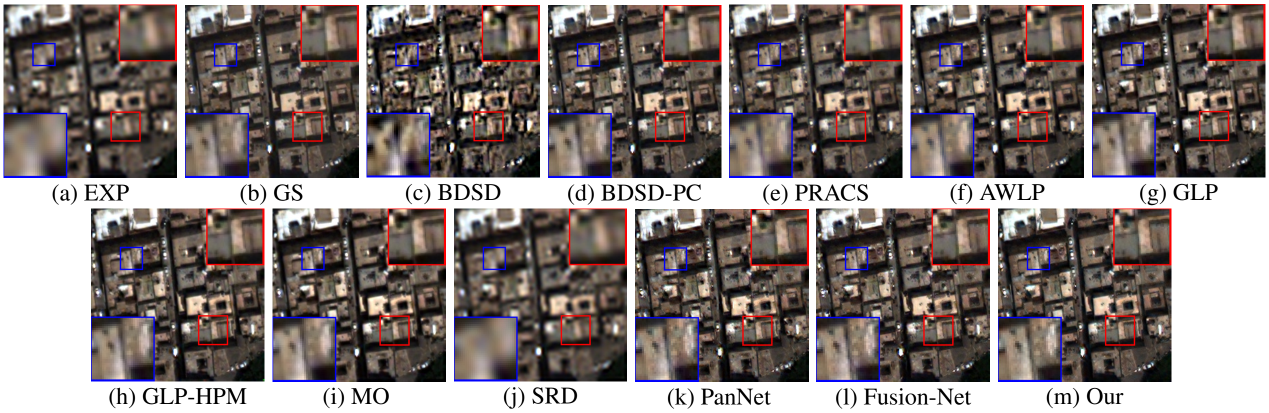

Figure 5: Visual comparisons of the fused HRMS image for all the methods on a full resolution sample.

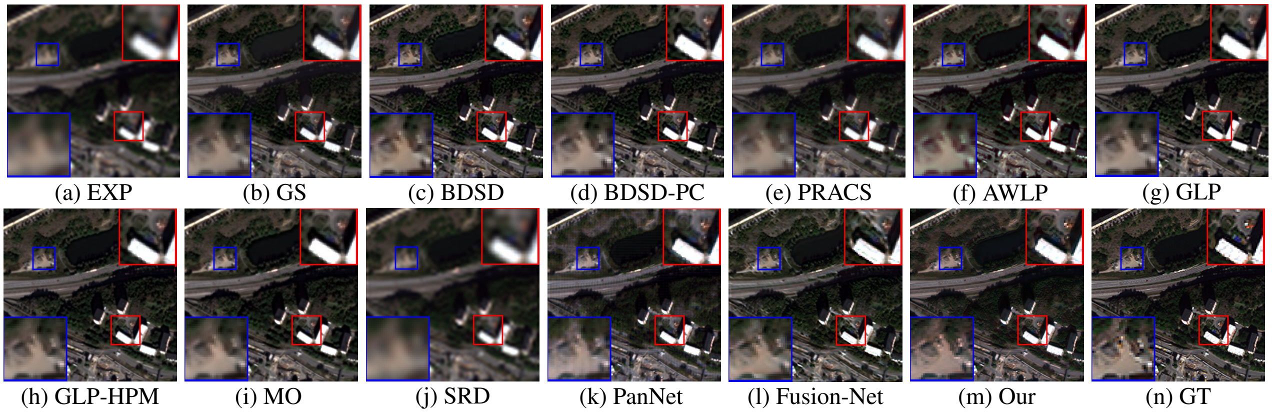

Figure 6: Visual comparisons of the fused HRMS image for all the methods on the WorldView-2 dataset.

Methods	**Q8**	**SAM**	**ERGAS**
EXP	0.147±0.099	8.469±0.690	8.448±0.858
GS	0.369±0.097	8.900±1.137	6.050±0.328
BDSD	0.381±0.079	8.141±0.851	5.624±0.635
BDSD-PC	0.226±0.079	8.555±0.851	6.765±0.635
PRACS	0.234±0.127	8.238±0.807	6.770±0.538
AWLP	**0.404±0.133**	8.033±0.728	5.828±0.600
GLP	0.377±0.065	7.901±0.820	5.584±0.536
GLP-HPM	0.317±0.074	8.094±0.948	5.843±0.601
MO	0.338±0.110	7.883±0.979	5.921±0.513
SRD	0.147±0.099	8.469±0.690	8.448±0.858
PanNet	0.374±0.105	7.577±0.667	5.540±0.174
Fusion-Net	0.368±0.079	8.214±1.132	6.140±0.359
Our	0.388±0.133	**7.523±0.989**	**5.410±0.184**

Table 4: Average result on WorldView-2 data (4 samples).

our network has clearer visual results.

Parameters and Time Comparisons

Table 5 compares the time, the parameter number, and the giga floating-point operations per second (GFLOPs) of our network with the other two DL-based methods. From Table 5, we can find that our network has a comparable efficiency with other methods while contains fewer parameters.

	Image Size	**PanNet**	**Fusion-Net**	**Our**
Time (s)	256×256 64×64×8	2.63	2.31	2.79
GFLOPs	256×256 64×64×8	10.3	10.2	11.2
# Par($\times10^4$)	-	7.76	7.63	**7.03**

Table 5: Parameters and time comparisons.

Conclusion

This paper proposes a novel deep network for pansharpening using the model-based deep learning method. Specifically, we first propose a new pansharpening model based on the convolutional sparse coding. Then, we design a proximal gradient algorithm to solve this model, and this algorithm is unfolded into a network. Since each network module corresponds to an operation of the algorithm, our network has a clear physical interpretation. This helps us to better understand how the network works. Experimental results on some benchmark datasets illustrate our network performs better than other approaches both quantitatively and qualitatively.

Acknowledgements

This work was supported by National Key Research and Development Project of China (2021ZD0110700), National Natural Science Foundation of China (61906151, 12101384, 62050194, 62037001), the Fundamental Research Funds for the Central Universities (GK202103001), Innovative Research Group of the National Natural Science Foundation of China(61721002), Innovation Research Team of Ministry of Education (IRT_17R86), Project of China Knowledge Centre for Engineering Science and Technology, and Project of XJTU Undergraduate Teaching Reform (20JX04Y).

References

Aiazzi, B.; Alparone, L.; Baronti, S.; and Garzelli, A. 2002. Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 40(10): 2300–2312.

Beck, A.; and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1): 183–202.

Cao, X.; Fu, X.; Hong, D.; Xu, Z.; and Meng, D. 2021. PanCSC-Net: A model-driven deep unfolding method for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, DOI: 10.1109/TGRS.2021.3115501.

Carper, W.; Lillesand, T.; and Kiefer, R. 1990. The use of intensity-hue-saturation transformations for merging SPOT panchromatic and multispectral image data. *Photogrammetric Engineering and Remote Sensing*, 56(4): 459–467.

Choi, J.; Yu, K.; and Kim, Y. 2010. A new adaptive component-substitution-based satellite image fusion by using partial replacement. *IEEE Transactions on Geoscience and Remote Sensing*, 49(1): 295–309.

Deng, L.-J.; Feng, M.; and Tai, X.-C. 2019. The fusion of panchromatic and multispectral remote sensing images via tensor-based sparse modeling and hyper-Laplacian prior. *Information Fusion*, 52: 76–89.

Deng, L.-J.; Vivone, G.; Guo, W.; Dalla Mura, M.; and Chanussot, J. 2018. A variational pansharpening approach based on reproducible kernel Hilbert space and heaviside function. *IEEE Transactions on Image Processing*, 27(9): 4330–4344.

Deng, L.-J.; Vivone, G.; Jin, C.; and Chanussot, J. 2020. Detail injection-based deep convolutional neural networks for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, DOI: 10.1109/TGRS.2020.3031366.

Deng, X.; and Dragotti, P. L. 2020. Deep convolutional neural network for multi-modal image restoration and fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, DOI: 10.1109/TPAMI.2020.2984244.

Dong, W.; Zhou, C.; Wu, F.; Wu, J.; Shi, G.; and Li, X. 2021. Model-guided deep hyperspectral image super-resolution. *IEEE Transactions on Image Processing*, DOI: 10.1109/TIP.2021.3078058.

Fang, F.; Li, F.; Shen, C.; and Zhang, G. 2013a. A variational approach for pan-sharpening. *IEEE Transactions on Image Processing*, 22(7): 2822–2834.

Fang, F.; Li, F.; Zhang, G.; and Shen, C. 2013b. A variational method for multisource remote-sensing image fusion. *International Journal of Remote Sensing*, 34(7): 2470–2486.

Fu, X.; Lin, Z.; Huang, Y.; and Ding, X. 2019. A variational pan-sharpening with local gradient constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10265–10274.

Fu, X.; Wang, W.; Huang, Y.; Ding, X.; and Paisley, J. 2020. Deep multiscale detail networks for multiband spectral image sharpening. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5): 2090–2104.

Garzelli, A.; Nencini, F.; and Capobianco, L. 2007. Optimal MMSE pan sharpening of very high resolution multispectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 46(1): 228–236.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

He, L.; Rao, Y.; Li, J.; Chanussot, J.; Plaza, A.; Zhu, J.; and Li, B. 2019. Pansharpening via detail injection based convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(4): 1188–1204.

Huang, W.; Xiao, L.; Wei, Z.; Liu, H.; and Tang, S. 2015. A new pan-sharpening method with deep neural networks. *IEEE Geoscience and Remote Sensing Letters*, 12(5): 1037–1041.

Laben, C. A.; and Brower, B. V. 2000. Process for enhancing the spatial resolution of multispectral imagery using pansharpening. US Patent 6,011,875.

Li, M.; Cao, X.; Zhao, Q.; Zhang, L.; and Meng, D. 2021. Online rain/snow removal from surveillance videos. *IEEE Transactions on Image Processing*, 30: 2029–2044.

Li, S.; and Yang, B. 2010. A new pan-sharpening method using a compressed sensing technique. *IEEE Transactions on Geoscience and Remote Sensing*, 49(2): 738–746.

Liu, J. 2000. Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details. *International Journal of Remote Sensing*, 21(18): 3461–3472.

Masi, G.; Cozzolino, D.; Verdoliva, L.; and Scarpa, G. 2016. Pansharpening by convolutional neural networks. *Remote Sensing*, 8(7): 594.

Otazu, X.; González-Audícana, M.; Fors, O.; and Núñez, J. 2005. Introduction of sensor spectral response into image fusion methods. Application to wavelet-based methods. *IEEE Transactions on Geoscience and Remote Sensing*, 43(10): 2376–2385.

Restaino, R.; Vivone, G.; Addesso, P.; and Chanussot, J. 2019. A pansharpening approach based on multiple linear regression estimation of injection coefficients. *IEEE Geoscience and Remote Sensing Letters*, 17(1): 102–106.

Restaino, R.; Vivone, G.; Dalla Mura, M.; and Chanussot, J. 2016. Fusion of multispectral and panchromatic images based on morphological operators. *IEEE Transactions on Image Processing*, 25(6): 2882–2895.

Shah, V. P.; Younan, N. H.; and King, R. L. 2008. An efficient pan-sharpening method via a combined adaptive PCA approach and contourlets. *IEEE Transactions on Geoscience and Remote Sensing*, 46(5): 1323–1335.

Tian, X.; Chen, Y.; Yang, C.; and Ma, J. 2021. Variational pansharpening by exploiting cartoon-texture similarities. *IEEE Transactions on Geoscience and Remote Sensing*, DOI: 10.1109/TGRS.2020.3048257.

Vicinanza, M. R.; Restaino, R.; Vivone, G.; Dalla Mura, M.; and Chanussot, J. 2014. A pansharpening method based on the sparse representation of injected details. *IEEE Geoscience and Remote Sensing Letters*, 12(1): 180–184.

Vivone, G. 2019. Robust band-dependent spatial-detail approaches for panchromatic sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9): 6421–6433.

Vivone, G.; Alparone, L.; Chanussot, J.; Dalla Mura, M.; Garzelli, A.; Licciardi, G. A.; Restaino, R.; and Wald, L. 2014. A critical comparison among pansharpening algorithms. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5): 2565–2586.

Vivone, G.; Marano, S.; and Chanussot, J. 2020. Pansharpening: context-based generalized Laplacian pyramids by robust regression. *IEEE Transactions on Geoscience and Remote Sensing*, 58(9): 6152–6167.

Vivone, G.; Restaino, R.; and Chanussot, J. 2018. Full scale regression-based injection coefficients for panchromatic sharpening. *IEEE Transactions on Image Processing*, 27(7): 3418–3431.

Vivone, G.; Restaino, R.; Dalla Mura, M.; Licciardi, G.; and Chanussot, J. 2013. Contrast and error-based fusion schemes for multispectral image pansharpening. *IEEE Geoscience and Remote Sensing Letters*, 11(5): 930–934.

Wald, L.; Ranchin, T.; and Mangolini, M. 1997. Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images. *Photogrammetric Engineering and Remote Sensing*, 63(6): 691–699.

Wang, T.; Fang, F.; Li, F.; and Zhang, G. 2018. High-quality Bayesian pansharpening. *IEEE Transactions on Image Processing*, 28(1): 227–239.

Xie, Q.; Zhou, M.; Zhao, Q.; Xu, Z.; and Meng, D. 2020a. MHF-net: An interpretable deep network for multispectral and hyperspectral image fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, DOI: 10.1109/TPAMI.2020.3015691.

Xie, W.; Cui, Y.; Li, Y.; Lei, J.; Du, Q.; and Li, J. 2020b. HPGAN: Hyperspectral pansharpening using 3-D generative adversarial networks. *IEEE Transactions on Geoscience and Remote Sensing*, 59(1): 463–477.

Yang, J.; Fu, X.; Hu, Y.; Huang, Y.; Ding, X.; and Paisley, J. 2017. PanNet: A deep network architecture for pansharpening. In *Proceedings of the IEEE International Conference on Computer Vision*, 5449–5457.

Yang, Y.; Tu, W.; Huang, S.; Lu, H.; Wan, W.; and Gan, L. 2021. Dual-stream convolutional neural network with residual information enhancement for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, DOI: 10.1109/TGRS.2021.3098752.

Yin, H. 2021. PSCSC-Net: A deep coupled convolutional sparse coding network for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, DOI: 10.1109/TGRS.2021.3088313.

Yuan, Q.; Wei, Y.; Meng, X.; Shen, H.; and Zhang, L. 2018. A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3): 978–989.

Zhang, H.; and Ma, J. 2021. GTP-PNet: A residual learning network based on gradient transformation prior for pansharpening. *ISPRS Journal of Photogrammetry and Remote Sensing*, 172: 223–239.

Zhou, H.; Liu, Q.; and Wang, Y. 2021. Pgman: An unsupervised generative multiadversarial network for pansharpening. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14: 6316–6327.