ISPRS Journal of Photogrammetry and Remote Sensing 219 (2025) 22–37

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Pansharpening via predictive filtering with element-wise feature mixing

Yongchuan Cui [a,b], Peng Liu [a,b,*], Yan Ma [a,b], Lajiao Chen [a,b], Mengzhen Xu [c], Xingyan Guo [c]

[a] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[b] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 101408, China
[c] State Key Laboratory of Hydroscience and Engineering, Department of Hydraulic Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Keywords:
Pansharpening
Image fusion
Predictive filtering
Deep learning

ABSTRACT

Pansharpening is a crucial technique in remote sensing for enhancing spatial resolution by fusing low spatial resolution multispectral (LRMS) images with high spatial panchromatic (PAN) images. Existing deep convolutional networks often face challenges in capturing fine details due to the homogeneous operation of convolutional kernels. In this paper, we propose a novel predictive filtering approach for pansharpening to mitigate spectral distortions and spatial degradations. By obtaining predictive filters through the fusion of LRMS and PAN and conducting filtering operations using unique kernels assigned to each pixel, our method reduces information loss significantly. To learn more effective kernels, we propose an effective fine-grained fusion method for LRMS and PAN features, namely element-wise feature mixing. Specifically, features of LRMS and PAN will be exchanged under the guidance of a learned mask. The value of the mask signifies the extent to which the element will be mixed. Extensive experimental results demonstrate that the proposed method achieves better performances than the state-of-the-art models with fewer parameters and lower computations. Visual comparisons indicate that our model pays more attention to details, which further confirms the effectiveness of the proposed fine-grained fusion method. Codes are available at https://github.com/yc-cui/PreMix.

1. Introduction

Due to technological and physical constraints, the spatial resolution of images provided by various imaging sensors is often limited with respect to their spectral resolution. Numerous orbiting satellites, such as IKONOS, QuickBird, GaoFen, *etc.*, are capable of simultaneously providing panchromatic images and multispectral images. Generally, multispectral images contain multiple bands with abundant spectral information. In contrast, single-band panchromatic images have richer spatial details compared to multispectral images. Pansharpening aims at fusing panchromatic (PAN) images and low spatial resolution multispectral (LRMS) images to obtain high spatial and spectral resolution (HRMS) images.

In recent years, pansharpening based on deep learning (DL) have demonstrated notably superior performance compared to traditional approaches. Thanks to the nonlinear fitting capabilities of deep neural networks, HRMS obtained through deep nonlinear transformation and automatic parameter updating often exhibits higher quality than manual feature extraction or constraint formulation. Most of the existing DL-based neural networks used for pansharpening generally perform feature extraction and transformation on the input degraded image, and directly obtain the predicted pixels, as shown in Fig. 1(a). However, the direct strategy has shown to be suboptimal, and leads to issues such as slow convergence and increased errors (Bako et al., 2017). Particularly in the context of remote sensing imagery, the datasets encapsulate intricate ground object information with rich spectral and spatial details. Employing the direct strategy may suffer from severe information loss, consequently impacting the precision of the pansharpening results. In contrast, predictive filtering (Bako et al., 2017) does not directly synthesize the pixels of the image to be predicted, but first predicts a convolution kernel for each pixel. Then the original degraded image is filtered using the predictive kernels, thereby indirectly obtaining the image to be predicted. The illustration of how predictive filtering works is shown in Fig. 1(b). The predictive filtering technique has been applied to a variety of low-level vision tasks (Bako et al., 2017; Mildenhall et al., 2018; Xia et al., 2020; Guo et al., 2021b; Cho et al., 2021; Fu et al., 2021; Guo et al., 2021a; Li et al., 2022b) and proven successful.

This paper investigates the application of predictive filtering technique for pansharpening. We attempt to learn predictive kernels for LRMS and PAN simultaneously and filter them to generate HRMS.

* Corresponding author at: Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China.
E-mail addresses: cugcuiyc@cug.edu.cn (Y. Cui), liupeng202303@aircas.ac.cn (P. Liu), mayan@radi.ac.cn (Y. Ma), chenli@radi.ac.cn (L. Chen), mzxu@tsinghua.edu.cn (M. Xu).

https://doi.org/10.1016/j.isprsjprs.2024.10.029
Received 31 March 2024; Received in revised form 27 July 2024; Accepted 29 October 2024
Available online 26 November 2024

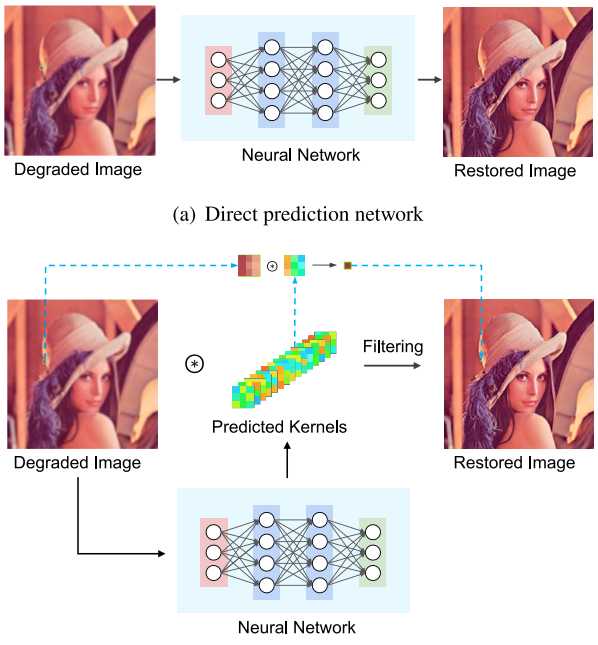


Fig. 1. Comparison of direct prediction network and kernel prediction network. The kernel prediction network employs predictive filtering to indirectly generate predicted pixels.

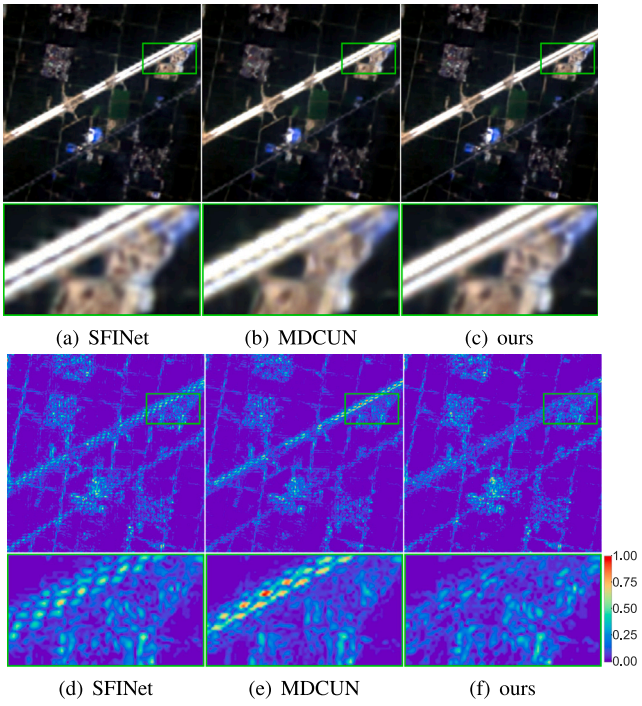


Fig. 2. Demonstration of the characteristics of the proposed model. The bright rectangular box represents the area zoomed in for display. (a)–(c) Prediction of different models. (e)–(f) The corresponding mean absolute error between the prediction and the ground truth.

The proposed model aims to minimize information loss compared to the deep nonlinear transformation of LRMS and PAN, thus reducing spatial degradations and spectral distortions. To demonstrate the effectiveness of our approach, we present a representative example of our fused result with comparison to two DL-based state-of-the-art (SOTA) methods, *i.e.*., SFINet (Zhou et al., 2022b) and MDCUN (Yang et al., 2022). Clearly, the fused outputs from SFINet (Zhou et al., 2022b) and MDCUN (Yang et al., 2022) show different levels of local spatial structure degradation and inaccurate reconstruction result, whereas our proposed model achieves a lower reconstruction error.

Our method demonstrates superiority due to the introduction of a fine-grained feature fusion approach, namely element-wise feature mixing (EWFM). Inspired by the customized kernel strategy of predictive filtering, EWFM utilizes a guiding mask to perform element exchange operations on the features of LRMS and PAN flow. We prove that this fusion strategy is actually equivalent to customizing a dynamically changing convolution kernel for each element in the feature map, akin to predictive filtering. Instead of convolving the entire feature map using a single same group of convolutional kernels as in regular convolution, EWFM utilizes a kernel-wise attention mechanism to assign a dynamic convolutional kernel to each element. We also proposed a multi-scale and multi-branch progressive filtering network to conduct multiple filtering to further improve the performance of pansharpening. Specifically, we leverage the histogram equalization and high pass filtering on LRMS and PAN as additional branches, conducting progressive predictive filtering across multiple resolution scales. Extensive experimental results show that this fully kernel-customized network outperforms SOTA DL-based models with reduced parameters and computations at both simulated reduced-resolution data and real-world full-resolution data.

In summary, our contributions are as follows:

- We explore predictive filtering for remote sensing multispectral and panchromatic image fusion for the first time.
- In order to predict effective filtering kernels, we propose a fine-grained feature fusion strategy. To obtain a more accurate result, We also propose a multi-scale and multi-branch progressive filtering model.
- We conducted extensive experiments, including numerous comparative analyses, ablation studies, and parameter analyses, using a variety of satellite data from GaoFen-1, WorldView-2, and IKONOS. The qualitative and quantitative results demonstrate that the model presented in this paper outperforms other models at both simulated and real data.

The remainder of this paper is organized as follows: Section 2 provides a review of pansharpening methods and predictive filtering. Section 3 explains the methods proposed in this paper. Section 4 presents a detailed comparison of the experimental results and provides an in-depth analysis of the experiments. Finally, Section 5 concludes the article and discusses future work.

2. Related work

This section briefly reviews traditional and DL-based pansharpening methods, while also introducing the development and application of kernel prediction neural networks.

2.1. Traditional pansharpening methods

Classic pansharpening methods can be roughly divided into three categories: Meng et al. (2019) and Vivone et al. (2021b,a): (i). component substitution (CS) approaches; (ii). multi-resolution analysis (MRA) approaches; (iii). variational optimization (VO) approaches.

The core idea of the CS-based methods is to utilize the relationship between the MS image and the PAN image to extract the high-frequency part of the PAN image, and then inject it into the upsampled LRMS image to obtain the final fused image HRMS. Commonly used methods are the principal component analysis (PCA) method (Chavez and Kwarteng, 1989), intensity-hue-saturation (IHS) fusion (Rahmani et al., 2010), the Gram–Schmidt (GS) spectral sharpening approach (Laben and Brower, 2000), the band-dependent spatial- detail (BDSD) method (Garzelli

et al., 2008). CS-based methods exhibit reduced computational time but may entail the loss of certain spectral details. Different from the CS-based methods, the MRA-based methods apply a multi-resolution transformation and extract high-frequency information from the PAN image, and then inject the information into the upsampled LRMS image to obtain the fused HRMS. Some instances of approaches include the Laplacian pyramid (LP) method (Burt and Adelson, 1983), smoothing filter-based intensity modulation (SFIM) (Liu, 2000), and the modulation transfer function generalized LP (MTF-GLP) (Aiazzi et al., 2002, 2006) with high-pass modulation injection (MTF-GLP-HPM) technique (Aiazzi et al., 2003). Unlike the CS algorithm, the MRA algorithm can avoid some spectral distortion, but some structural information will be lost during the filtering process (Sheng et al., 2023). VO-based methods typically involve formulating an optimization problem where the objective is to strike a balance between retaining spectral information from the multispectral image and enhancing spatial details from the panchromatic image. The pioneering work by Ballester et al. (2006) proposed the first VO algorithm, paving the way for subsequent algorithms that incorporate principles of sparsity (Zhang et al., 2019; Li et al., 2013), low rank (Zhang et al., 2021; Yang et al., 2018; Dian and Li, 2019), and variation priors (Liu et al., 2016b,a) to produce a more accurate HRMS image. However, VO-based methods often come with high computational complexity and rely heavily on specific model assumptions. In cases where the number of iterative optimizations is insufficient or the model assumptions are inaccurate, significant spectral or spatial distortions can arise. The abovementioned traditional methods are rooted in subjective assumptions during the super-resolution process, and their limited nonlinear capabilities may lead to spectral distortions during pansharpening.

2.2. Deep learning based pansharpening methods

Driven by the remarkable achievements of deep learning across various vision tasks, numerous DL-based approaches have emerged for pansharpening, especially convolutional neural network (CNN)-based methods. Leveraging their exceptional hierarchical feature representation capabilities, DL methods can efficiently learn robust priors and achieve competitive performance (Liu et al., 2022a).

Since the introduction of CNN for pansharpening by Masi et al. (2016), CNN-based data-driven techniques (Liu et al., 2022b) have widely emerged. Yang et al. (2017) incorporate domain-specific knowledge and employ residual block (He et al., 2016) to achieve spectral and spatial preservation. Then (Scarpa et al., 2018) proposed an improved PNN+ network. Xing et al. (2018) proposed a deep metric learning method which utilizes multiple nonlinear deep neural networks to learn a refined geometric multi-manifold neighbor embedding. Jiang et al. (2020) utilized a differential information mapping strategy and incorporated an attention module to enhance the spatial details in the fusion results. GTP-PNet (Zhang and Ma, 2021) seeks the nonlinear mapping between the gradients of the PAN and HRMS. HyperNet (Li et al., 2022a) utilized multiscale-attention-enhance blocks and dense-detail-insertion blocks to extract spatial details. The CMINet (Wang et al., 2024) framework integrated three modules to enhance modality-aware features and address modality misalignment issues. He et al. (2024) proposed to use implicit neural representations to parameterize images by neural networks for arbitrary-resolution pansharpening. Thanh Nhat Mai et al. (2024) developed a deep unfolding tensor rank minimization framework combined with a generalized detail injection approach for pansharpening, which effectively leverages the low-rank property of multispectral images and enhances spatial and spectral fidelity without relying on handcrafted formulations or empirical architectures.

While supervised learning has garnered significant achievements for pansharpening, recent studies have highlighted several limitations, such as the requirement for a substantial amount of labeled training data and scale-related problems. Consequently, several unsupervised frameworks have been introduced to mitigate drawbacks inherent in supervised approaches. Shen et al. (2023) addressed the issue of scale-shift introduced by supervised learning-based methods with a general training framework. Liu et al. (2023) proposed a supervised–unsupervised combined network by integrating a supervised network based on Wald's protocol (Wald et al., 1997) and an unsupervised spatial–spectral compensation network to achieve high-fidelity pansharpening. The UAP-Net (Xiong et al., 2023) introduced an unsupervised pansharpening approach by leveraging a deep residual network augmented with a spatial texture attention mechanism that employs the high-frequency component of the input PAN as weights. PAN-MGDR (Lin et al., 2024) proposed an unsupervised pan-sharpening framework by adaptive blur kernel estimation and mutually guided detail restoration. Z-PNN (Ciotola et al., 2022) and λ-PNN (Ciotola et al., 2023) focused on designing spatial and spectral loss functions to facilitate effective unsupervised learning. Generative adversarial networks (Goodfellow et al., 2014; Cui et al., 2024; Liu et al., 2022a; Ozcelik et al., 2021) (GANs), renowned for their ability to produce realistic images through an unsupervised adversarial learning, have recently been integrated into pansharpening (Shao et al., 2020; Ozcelik et al., 2021; Liu et al., 2020; Zhou et al., 2022a; Xu et al., 2023). For example, Ma et al. (2020) proposed a GAN-based framework constrained by both spectral and textural loss. Xu et al. (2023) employed a coarse-to-fine framework and customized loss functions to enhance both spatial and spectral fidelity.

However, the aforementioned DL-based models directly extract features from LRMS and PAN to get HRMS. As stated in the introduction of this paper and as shown in Fig. 2, the direct manner may cause information loss and fail to effectively preserve spatial or spectral characteristics. Therefore, this paper attempts to explore kernel prediction by filtering LRMS and PAN to achieve better spatial and spectral preservation.

2.3. Kernel prediction network

In contrast to regular convolutions that directly operate on degraded image to obtain the final output, deep predictive filtering aims to learn a dynamic convolutional kernel for each pixel, which is then applied for filtering on degraded image. The advantage of kernel prediction network lies in that the predictive filtering allows more focused learning of the surrounding information for each pixel. Moreover, as the filtering operation is performed directly on the original image, it can significantly minimize information loss. The predictive filtering technique has been widely applied in various low-level vision tasks, *e.g.*, denoising (Bako et al., 2017; Mildenhall et al., 2018; Xia et al., 2020), deraining (Guo et al., 2021b), super-resolution (Cho et al., 2021), shadow removal (Fu et al., 2021), inpainting (Guo et al., 2021a; Li et al., 2022b). This paper investigates the use of predictive filtering for pansharpening and explores methods for fusing features to obtain effective kernels.

3. Methodology

In this section, we propose a kernel prediction network namely PreMix that incorporates predictive filtering (abbreviated as `PF`) for pansharpening and element-wise feature mixing (abbreviated as `EWFM`) for the fine-grained fusion of LRMS and PAN. The structure of the proposed base network is shown in Fig. 3.

3.1. Element-wise feature mixing

In order to perform fine-grained feature fusion to obtain highly precise predictive kernels used to filter LRMS and PAN, we propose element-wise feature mixing. The basic intuition is to customize a unique convolutional kernel for each element of the features, instead of homogeneously convolving the input features with the same group

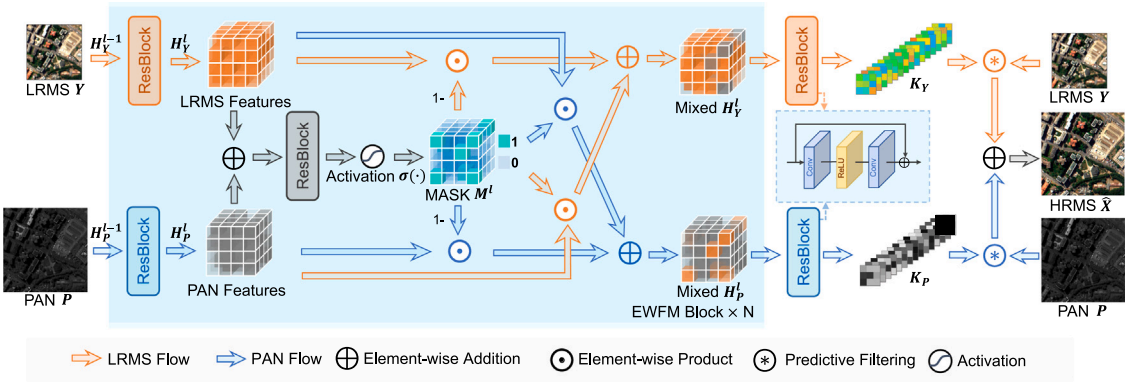


Fig. 3. Data flow of our proposed PreMix-Base version.

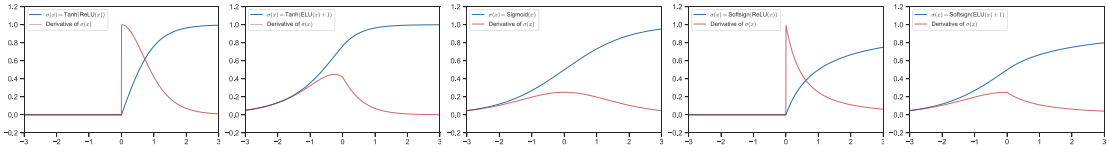


Fig. 4. Different mask generation functions and their gradients. Compared to the commonly used sigmoid function (c), the combination of Tanh and ReLU (a) has been experimentally proven to be superior.

of convolutional kernels. It is worth noting that the operation of customizing convolutional kernels in EWFM is actually implicit. We do not directly initialize learnable parameters of convolutional kernels for each element, as this would result in a huge number of parameters and computations. Instead, we use a learned mask to control the convolutional kernel to achieve dynamic weights.

Let the tensor version of upsampled LRMS be denoted as $\boldsymbol{Y} \in \mathbb{R}^{H\times W\times C}$, where H and W are the height and width and C is the number of channels of an LRMS image which is upsampled to the same size of a PAN image. $\boldsymbol{P} \in \mathbb{R}^{H\times W\times C}$ is a PAN image with height of H and width of W and its channel is duplicated for C times. Let the ground truth of HRMS be denoted as $\boldsymbol{X} \in \mathbb{R}^{H\times W\times C}$. The predicted HRMS obtained through the network is denoted as $\hat{\boldsymbol{X}} \in \mathbb{R}^{H\times W\times C}$. We first extract features from LRMS and PAN separately. Assuming $\boldsymbol{H}_Y^{l-1}$ and $\boldsymbol{H}_P^{l-1}$ are the outputs of the LRMS flow and PAN flow of the $l-1$th EWFM block. In the lth EWFM block, the features will be further extracted through convolution,

$$\boldsymbol{H}_Y^l = \mathrm{Conv}(\boldsymbol{H}_Y^{l-1}), \tag{1}$$

$$\boldsymbol{H}_P^l = \mathrm{Conv}(\boldsymbol{H}_P^{l-1}), \tag{2}$$

where Conv(·) represents the regular convolution operation which incorporates a residual block (ResBlock) (He et al., 2016) that sequentially features a convolution layer, a rectified linear unit (ReLU) (Glorot et al., 2011) activation function, and a subsequent convolution layer, culminating with a skip connection. Noting that the first EWFM block directly convolves the original LRMS and PAN, *i.e.*, when $l = 1$, $\boldsymbol{H}_Y^0 = \boldsymbol{Y}$ and $\boldsymbol{H}_P^0 = \boldsymbol{P}$. We then integrate the information of these two features and learn a mask,

$$\boldsymbol{M}^l = \sigma\left(\mathrm{Conv}(\boldsymbol{H}_Y^l + \boldsymbol{H}_P^l)\right), \tag{3}$$

where $\sigma(\cdot)$ is an activation function that maps its inputs to values ranging between 0 and 1. Typically, the function $\sigma(\cdot)$ can be the sigmoid function, but it may not be the best choice. The sigmoid function, characterized by a maximum gradient value of 0.25 (refer to Fig. 4(c)), can induce slow optimization and gradient vanishing problems as the number of layers increases. Moreover, the sigmoid has a smooth transition and lacks sparsity. The generated mask may be similar, thereby leading to insignificant features. Therefore, to explore effective mask generation methods, we explore another 5 activation functions with different properties (say, gradients, sparse activating, *etc.*). These functions correspond to three mixing strategies: soft mixing, sparse soft mixing, and hard mixing. Detailed explanations will be provided in Section 3.2.

After obtaining $\boldsymbol{M}^l$, we perform a mixing operation on the original features,

$$\boldsymbol{H}_Y^l \leftarrow (1-\boldsymbol{M}^l)\odot \boldsymbol{H}_Y^l + \boldsymbol{M}^l \odot \boldsymbol{H}_P^l, \tag{4}$$

$$\boldsymbol{H}_P^l \leftarrow (1-\boldsymbol{M}^l)\odot \boldsymbol{H}_P^l + \boldsymbol{M}^l \odot \boldsymbol{H}_Y^l, \tag{5}$$

where $\odot$ is the element-wise product. The mixed features will be used as the inputs for the next layer of the EWFM block. Note that when carrying out the convolution operation using the mixed result, it implicitly includes customizing the convolution kernel for each element. In the next, we will analyze the mixing operation from both the perspective of features and kernels.

3.1.1. Feature perspective

From the perspective of features, the value of $\boldsymbol{M}^l$ in Eqs. (4) and (5) indicates the extent to which features are exchanged. Taking LRMS flow as an example, when certain elements in $\boldsymbol{M}^l$ are equal to 1, it means that these elements of LRMS will be exchanged by the features extracted by PAN flow. Supposing $\boldsymbol{h_y}'$ and $\boldsymbol{h_p}'$ are the local neighborhoods of a certain element of the mixed features. Let $\boldsymbol{k_1}$ and $\boldsymbol{k_2}$ be the convolutional kernels for LRMS flow and PAN flow, respectively. Convolving $\boldsymbol{h_y}'$ with $\boldsymbol{k_1}$ yields,

$$\begin{aligned}\boldsymbol{h_y}' * \boldsymbol{k_1} &= \sum_{j\in\mathcal{N}_i}\left(\left(1-\boldsymbol{m}^j\right)\times \boldsymbol{h_y}^j\right)\times \boldsymbol{k_1}^{j-i} \\ &+ \sum_{j\in\mathcal{N}_i}\left(\boldsymbol{m}^j\times \boldsymbol{h_p}^j\right)\times \boldsymbol{k_1}^{j-i} \\ &= \left((1-\boldsymbol{m})\odot \boldsymbol{h_y}\right) * \boldsymbol{k_1} + \left(\boldsymbol{m}\odot \boldsymbol{h_p}\right) * \boldsymbol{k_1},\end{aligned} \tag{6}$$

where $*$ is the convolution operation and $\mathcal{N}_i$ are neighbors of the ith element. $\boldsymbol{m}$, $\boldsymbol{h_y}$, $\boldsymbol{h_p}$ represent the local elements of $\boldsymbol{M}^l$, $\boldsymbol{H_Y}$, $\boldsymbol{H_P}$ during convolution, respectively. Note that the convolution operation in neural networks is actually performing a cross-correlation for elements and

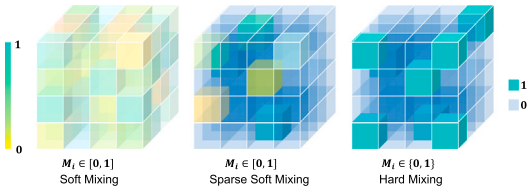


Fig. 5. Different mixing strategies. Left: Soft mixing, each element $\boldsymbol{M}_i$ varies from 0 to 1. Middle: Sparse soft mixing, *i.e.*, sparse version of soft mixing, the majority of elements in $\boldsymbol{M}$ are 0. Right: Hard mixing, $\boldsymbol{M}_i$ approaches 0 or 1.

kernel. This operation is inverted to the traditional convolution, *i.e.*, the kernel is rotated by 180°, but since the parameters of the convolution kernel are trainable, both operations are equivalent. In this paper, Eq. (6) is expressed by traditional convolution. Similarly, the kernel $\boldsymbol{k}_2$ for convolution on $\boldsymbol{h}'_p$ gives,

$$\boldsymbol{h}'_p * \boldsymbol{k}_2 = \left((1-\boldsymbol{m}) \odot \boldsymbol{h}_p\right) * \boldsymbol{k}_2 + \left(\boldsymbol{m} \odot \boldsymbol{h}_y\right) * \boldsymbol{k}_2. \tag{7}$$

From the perspective of features, taking LRMS as an example, $\boldsymbol{h}_y$ will be masked into two parts, where one part of $(1-\boldsymbol{m})$ will be convoluted by $\boldsymbol{k}_1$, and the other part of $\boldsymbol{m}$ will be convoluted by $\boldsymbol{k}_2$.

3.1.2. Kernel perspective

From the perspective of convolution kernels, this operation is actually equivalent to kernel-wise attention. The kernel for every single element is dynamically changed by the weight which is controlled by the learned mask,

$$\begin{aligned}\boldsymbol{h}'_y * \boldsymbol{k}_1 = &\sum_{j\in\mathcal{N}_i} \boldsymbol{h}_y^j \times \left(\left(1-\boldsymbol{m}^j\right) \times \boldsymbol{k}_1^{j-i}\right) \\ &+ \sum_{j\in\mathcal{N}_i} \boldsymbol{h}_p^j \times (\boldsymbol{m}^j \times \boldsymbol{k}_1^{j-i}) \\ &= \boldsymbol{h}_y * \left(\mathtt{rot180}\,(1-\boldsymbol{m}) \odot \boldsymbol{k}_1\right) \\ &+ \boldsymbol{h}_p * \left(\mathtt{rot180}\,(\boldsymbol{m}) \odot \boldsymbol{k}_1\right),\end{aligned} \tag{8}$$

where $\mathtt{rot180}(\cdot)$ represents flipping the input by 180°. $\mathtt{rot180}(\cdot)$ aligns the superscripts of $\boldsymbol{m}$ and $\boldsymbol{k}_1$. For $\boldsymbol{m}$ with superscript j, it needs to be multiplied by $\boldsymbol{k}_1$ with superscript $j-i$, which corresponds to the operation of rotating $\boldsymbol{m}$ by 180° and then performing a dot product. Similarly, Eq. (7) can be rewritten as,

$$\begin{aligned}\boldsymbol{h}'_p * \boldsymbol{k}_2 = &\boldsymbol{h}_p * \left(\mathtt{rot180}\,(1-\boldsymbol{m}) \odot \boldsymbol{k}_2\right) \\ &+ \boldsymbol{h}_y * \left(\mathtt{rot180}\,(\boldsymbol{m}) \odot \boldsymbol{k}_2\right).\end{aligned} \tag{9}$$

According to Eqs. (8) and (9), although each set of convolution kernels $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ are fixed, we assign different weights to the elements in the convolution kernels through a learnable mask. As the proposed model is an LRMS and PAN dual-stream feature extraction network, $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ can be controlled by $\boldsymbol{m}$ and $1-\boldsymbol{m}$, respectively. This dynamic convolution endows diverse weights to each element of the convolution kernel, allowing for extremely fine-grained feature fusion. Results from ablation experiments (see Table 4) demonstrate that the proposed `EWFM` operation is more effective than only using predictive filtering, and the combination of both achieves the best performance.

3.2. Mixing strategy

We control $\sigma(\cdot)$ in Eq. (3) to generate masks corresponding to different mixing strategies. Typically, the sigmoid function is used to map inputs to the range between 0 and 1. This function actually corresponds to a soft fusion strategy, as shown in Fig. 5 (left). However, this strategy may not be effective because the masks activated by the sigmoid function may become homogeneous due to the smoothness of the function. Remote sensing images are complex and diverse, thus we want to learn a more flexible mask. Inspired by the sparse activation of the `ReLU` (Glorot et al., 2011) function, we propose other functions to replace the sigmoid function to explore more feasible mixing strategies, as shown in Fig. 5 (middle and right). Experimental results show that sparsity enables focusing on different features in each layer, avoiding the attention homogenization of the sigmoid. Below, we will introduce three mixing strategies as shown in Fig. 5.

3.2.1. Soft mixing

Soft mixing maps all its inputs to between 0 and 1. The representative function is sigmoid activation:

$$\begin{aligned}\sigma(x) &= \mathtt{Sigmoid}(x) \\ &= \frac{1}{1+e^{-x}}.\end{aligned} \tag{10}$$

As mentioned before, the over-smoothness of the sigmoid potentially leads to redundancy and inefficiency in the learned mask. We explore alternative activation functions for generating masks. One of them is the combination of `Tanh` and `ELU` (Clevert et al., 2015),

$$\begin{aligned}\sigma(x) &= \mathtt{Tanh}\left(\mathtt{ELU}(x)+1\right) \\ &= \begin{cases} \frac{2}{1+e^{-2e^x}} - 1 & \text{if } x < 0, \\ \frac{2}{1+e^{-2(x+1)}} - 1 & \text{if } x \geq 0, \end{cases}\end{aligned} \tag{11}$$

where $\mathtt{ELU}(x)+1$ is to translate the inputs to a value greater than 0. Then it will be sent to `Tanh` to learn a mask from 0 to 1. As shown in Fig. 4(b), it has stronger gradients than sigmoid. Also, the asymmetry will learn a non-homogeneous mask. However, Eq. (11) involves more exponential operations. The alternative function is to replace `Tanh` with `Softsign`,

$$\begin{aligned}\sigma(x) &= \mathtt{Softsign}\left(\mathtt{ELU}(x)+1\right) \\ &= \begin{cases} \frac{e^x}{e^x+1} & \text{if } x < 0, \\ \frac{x+1}{x+2} & \text{if } x \geq 0. \end{cases}\end{aligned} \tag{12}$$

Eq. (12) is not computationally expensive and also has asymmetry, but its gradient is smaller than sigmoid (see Fig. 4(e)).

3.2.2. Sparse soft mixing

The above mentioned functions are not sparsity-inducing. Due to their smooth transition, very similar masks may be learned. Therefore, we propose sparse soft mixing (Fig. 5 (middle)) to learn a significantly differentiating mask.

$$\begin{aligned}\sigma(x) &= \mathtt{Tanh}\left(\mathtt{ReLU}(x)\right) \\ &= \begin{cases} 0 & \text{if } x < 0, \\ \frac{2}{1+e^{-2x}} - 1 & \text{if } x \geq 0. \end{cases}\end{aligned} \tag{13}$$

Eq. (13) is to truncating Eq. (11) where x is less than 0. Similarly, use a combination of `Softsign` and `ReLU` to reduce the computational load,

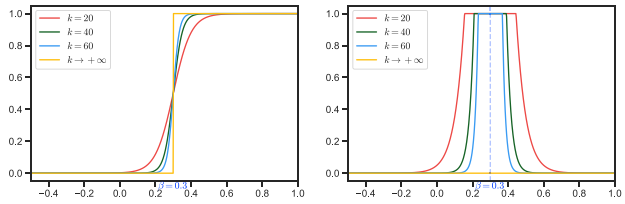
$$\begin{aligned}\sigma(x) &= \mathtt{Softsign}\left(\mathtt{ReLU}(x)\right) \\ &= \begin{cases} 0 & \text{if } x < 0, \\ \frac{x}{1+x} & \text{if } x \geq 0. \end{cases}\end{aligned} \tag{14}$$

Induce sparsity in the activations learn a more discriminative mask. The plot of sparse soft mixing is shown in Figs. 4(a) and 4(d).

3.2.3. Hard mixing

The extreme case of sparse soft mixing is hard mixing, *i.e.*, the mask either approaches 0 or 1. This forces the network to learn important features and suppress unimportant ones. In order to achieve hard mixing, the simplest method is manually set a threshold β and use a step function to map the input value to 0 or 1,

$$h(x) = \mathtt{Sign}\left(\sigma(x)\right) = \begin{cases} 0 & \text{if } \sigma(x) \leq \beta, \\ 1 & \text{otherwise}, \end{cases} \tag{15}$$



(a) Plots of Eq. (16) (b) Gradients of Eq. (16)

Fig. 6. Approximate the step function using the sigmoid function. Note that in (b), to prevent gradient explosion, we truncate the maximum gradient to 1.

where β is the threshold. However, the hard truncating approach has two drawbacks. Firstly, the mask is constrained to values of 0 or 1, potentially reducing diversity and leading to the learning of overly homogeneous features. Secondly, the parameter β is arbitrarily set, making its determination a time-consuming and laborious task. To address these issues, we suggest using the s-shaped function (Iliev et al., 2017, 2015), such as the sigmoid, to approximate the step function (see Fig. 6). It allows for the learned mask to be closer to 0 or 1, while also enabling the threshold parameter to be trainable,

$$\begin{aligned}\hat{h}(x) &= \texttt{Sigmoid}\left(k \times (\sigma(x) - \beta)\right) \\ &= \frac{1}{1 - e^{-k(\sigma(x)-\beta)}},\end{aligned} \tag{16}$$

where k is an amplification factor. Compared to $h(x)$, $\hat{h}(x)$ is differentiable. But the drawback of Eq. (16) is the introduction of an adjustable amplification factor k. Fortunately, the value of k can be empirically determined by examining the shape of the s-shaped function. In this paper, we fix k to 20.

3.3. Predictive filtering

After conducting element-wise feature mixing in each EWFM block, we utilize convolution to obtain predictive kernels $\boldsymbol{K}_Y$ and $\boldsymbol{K}_P$ at the final Lth layer of the network,

$$\boldsymbol{K}_Y = \text{Conv}(\boldsymbol{H}_Y^L), \tag{17}$$

$$\boldsymbol{K}_P = \text{Conv}(\boldsymbol{H}_P^L). \tag{18}$$

Afterwards, the predictive kernels from the LRMS flow and the PAN flow will be used to filter LRMS and PAN, respectively. We then add the filtered results to obtain the final prediction of HRMS,

$$\hat{\boldsymbol{X}} = \boldsymbol{Y} \circledast \boldsymbol{K}_Y + \boldsymbol{P} \circledast \boldsymbol{K}_P, \tag{19}$$

where $\circledast$ represents predictive filtering and the ith pixel of the prediction is computed as follows,

$$\begin{aligned}\hat{\boldsymbol{X}}[i] = &\sum_{j \in \mathcal{N}_i} \boldsymbol{K}_Y[j-i] \times \boldsymbol{Y}[j] \\ &+ \sum_{j \in \mathcal{N}_i} \boldsymbol{K}_P[j-i] \times \boldsymbol{P}[j].\end{aligned} \tag{20}$$

3.4. Multi-scale multi-branch progressive filtering

In Eq. (20), HRMS will be obtained by filtering LRMS and PAN a single time. We denote the abovementioned network architecture as PreMix-B (Base). In this section, we propose a multi-scale and multi-branch progressive filtering network, *i.e.*, PreMix-H (Huge). Compared to PreMix-B, PreMix-H is more powerful to conduct fine-grained feature fusion.

The architecture of PreMix-H is shown in Fig. 7. In addition to the original branch, we added histogram equalization (HE) branch and high pass (HP) filtering branch. The HE branch works towards achieving a more uniform data distribution (refer to Fig. 8(b)), thereby lowering the complexity of network training. The HP filtering branch, on the other hand, is designed to highlight the edges and details of the image (see Fig. 8(c)), enriching the spatial information of the filtering result. For implementing, we firstly apply a Gaussian low-pass filter to the original image using a 3×3 kernel with a standard deviation of 1.5×1.5 to obtain a low-pass filtered image. To ensure the filtered image retains the same dimensions as the original, we employ a reflection padding mode at the image boundaries. Subsequently, the high-pass filtered image is obtained by subtracting the low-pass filtered image from the original.

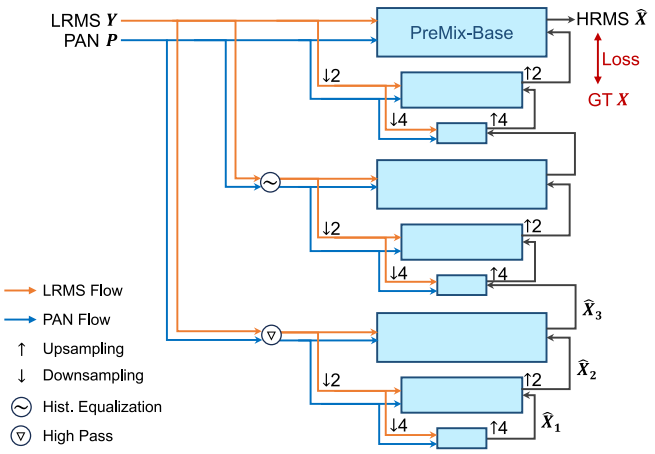
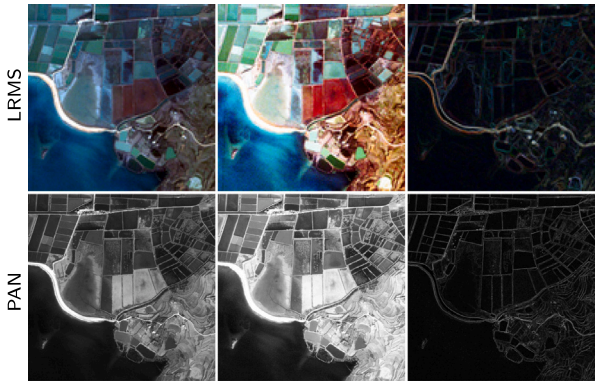


Fig. 7. Data flow of our proposed PreMix-Huge version.



(a) Original image (b) Hist. equalization (c) High pass

Fig. 8. Three branches used as inputs for PreMix-Huge.

To learn more diverse feature representations, each branch adopts multi-scale inputs, *i.e.*, the original image resolution is reduced by 2× and by 4× to serve as additional inputs. In each branch, the result of lower-resolution filtering will be passed to the upper layer, so that each layer will cascade filter the results from the previous layer. Through multi-scale and multi-branch filtering, we will obtain predicted images with abundant spectral information and rich spatial details. Taking the original branch as an example,

$$\hat{\boldsymbol{X}}_1 = f(\boldsymbol{Y}_{\downarrow 4}, \boldsymbol{P}_{\downarrow 4}, \boldsymbol{Y}, \boldsymbol{P}), \tag{21}$$

where $\hat{\boldsymbol{X}}_1$ is the predicted result obtained at the 4× scale, $f(\cdot)$ represents the PreMix-Base network, which takes in four inputs. The former two arguments are LRMS and PAN at the 4× scale for extracting predictive kernels, while the latter two are the original LRMS and PAN to be filtered. Then the LRMS and PAN at the 2× scale are used as inputs for filtering $\hat{\boldsymbol{X}}_1$ and PAN.

$$\hat{\boldsymbol{X}}_2 = f(\boldsymbol{Y}_{\downarrow 2}, \boldsymbol{P}_{\downarrow 2}, \hat{\boldsymbol{X}}_1, \boldsymbol{P}). \tag{22}$$

Similarly, obtain the final filtering result by filtering $\hat{\boldsymbol{X}}_2$ in progressive manner,

$$\hat{\boldsymbol{X}}_3 = f(\boldsymbol{Y}, \boldsymbol{P}, \hat{\boldsymbol{X}}_2, \boldsymbol{P}). \tag{23}$$

For other branches, the process is similar. As shown in Fig. 7, we take the final filtering results of each branch and feed them into the next branch for further filtering. At the same time, in order to maintain continuity of features, we add the features from the final layer of each PreMix-Base to the next layer.

4. Experiments

We compare the proposed PreMix model with several competitive methods containing 6 commonly-recognized traditional methods including Brovey (Gillespie et al., 1987), IHS (Rahmani et al., 2010), SFIM (Liu, 2000), GS (Laben and Brower, 2000), GSA (Aiazzi et al., 2007), CNMF (Yokoya et al., 2012), and 7 SOTA DL-based methods including 5 supervised methods, GPPNN (Xu et al., 2021), SFINet (Zhou et al., 2022b), MDCUN (Yang et al., 2022), PGCU (Zhu et al., 2023), UTSN (Sheng et al., 2023), and 2 unsupervised methods including Z-PNN (Ciotola et al., 2022) and λ-PNN (Ciotola et al., 2023). All codes are open source and can be found in the official repositories.

4.1. Steups

4.1.1. Datasets

To comprehensively evaluate the superiority of the proposed method in this study, we use a publicly available large-scale pansharpening dataset named NBU_PansharpRSData (Meng et al., 2021). We employed the panchromatic and multispectral images from the GaoFen-1, WorldView-2, and IKONOS satellite sensors for evaluation. The spatial resolution of LRMS and PAN in GaoFen-1 are 8 m and 2 m, respectively. The spatial resolution of LRMS and PAN in WorldView-2 are 2 m and 0.5 m, respectively. The spatial resolution of LRMS and PAN in IKONOS are 4 m and 1 m, respectively. The size of all multispectral images is 256 × 256, while the size of panchromatic images is 1024 × 1024. We partitioned the dataset into training, validation, and test sets in a ratio of 5:2:3.

In our experimentation, we conducted simulation experiments at reduced-resolution and real-world pansharpening experiments at full-resolution. For the simulation experiments, we utilized the widely adopted Wald's protocol (Wald et al., 1997) to generate the training data. Specifically, the PAN and LRMS images will be downsampled with a ratio r (where r is the ratio of the resolution from PAN to LRMS, which in this study is 4) as model inputs, with the original LRMS image considered as the ground truth (GT). In the real-world experiments, the original LRMS and PAN images were used as inputs and there is no GT for reference.

4.1.2. Implementation details

The network architecture is implemented using PyTorch version v2.1.2. The network is trained with a batch size of 16 and a learning rate of 1e−3 optimized by the Adam (Kingma and Ba, 2015) optimizer for 300 epochs. The learning rate decays by a factor of 0.8 every 50 epochs. We utilize the commonly used ℓ_1 loss as the loss function. All experiments were conducted on an NVIDIA RTX A5000 GPU with 24 GB of memory. The same settings were employed for all DL-based methods to validate the superiority of the network. In the experiments below, PreMix-B employs 2 layers of `EWFM` blocks, with mask generation using Eq. (13), and utilizes trainable hard mixing (Eq. (16)). In PreMix-H, the number of `EWFM` in PreMix-B is set to 1, while the other configurations remain unchanged. Full codes can be accessed at https://github.com/yc-cui/PreMix.

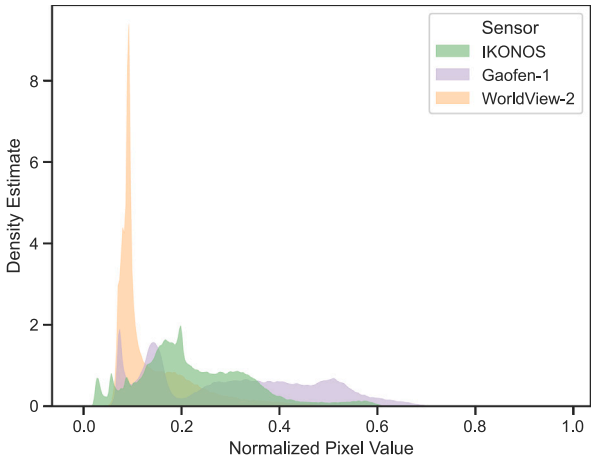


Fig. 9. The kernel density estimation (KDE) plots. The pixel values are normalized to 0 and 1.

4.1.3. Evaluation metrics

For simulated reduced-resolution experiments, we compare the differences between prediction and GT using the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM) (Wang et al., 2004), the erreur relative globale adimensionnelle de synthèse (ERGAS) (Wald et al., 1997), and the spectral angle mapper (SAM) (Yuhas et al., 1992). For full-resolution real-world data, the spectral distortion index D_λ (Alparone et al., 2008), the spatial distortion index D_S (Alparone et al., 2008), and the quality without reference (QNR) (Alparone et al., 2008) are employed as non-reference metrics.

4.2. Quantitative comparison

This section conducts a thorough quantitative comparison of the proposed method against existing models, providing a detailed assessment of performance metrics to evaluate the effectiveness of the proposed method.

4.2.1. Evaluation at reduced-resolution

Table 1 presents the comparison results of full-reference metrics for the simulated experiments. Traditional methods lag significantly behind DL-based approaches due to the inability to leverage large datasets for training. Among all DL-based models, our proposed PreMix-H outperforms others on all metrics across all satellite sensors, demonstrating the superiority of the proposed method. Specifically, taking PSNR as an example, PreMix-H achieves 43.62 dB (+1.83), 36.54 dB (+0.74), and 40.23 dB (+1.24) on the GaoFen-1, WorldView-2, and IKONOS datasets, respectively, compared to other methods. For SSIM, PreMix-H achieves 96.97%, 94.66% and 95.29%, the improvements are 1%, 0.63%, and 0.98% over other methods. PreMix-B performs similarly to SFINet (Zhou et al., 2022b) on GaoFen-1, surpasses it on all metrics on WorldView-2, but performs slightly inferior on IKONOS, where MDCUN (Yang et al., 2022) and SFINet (Zhou et al., 2022b) achieve better results.

The main reason for PreMix-B achieving relatively inferior results lies in the inherent distribution differences among datasets from different satellites. Each remote sensing dataset possesses unique spectral features and spatial details. The performance of PreMix-B on the IKONOS dataset is slightly less satisfactory, indicating that the model's fitting capability for IKONOS satellite data is marginally inferior compared to its fitting performance on other datasets. To statistically analyzed the reasons, Fig. 9 displays the 1d probability kernel density estimation (KDE) plots of the red band for each satellite, depicting the univariate distribution of pixel values sampled from the satellite data (normalized to 0 to 1). It is observable that datasets from the two distinct satellite

Table 1
Quantitative comparison results on simulated reduced-resolution data. ↓: Lower is better. ↑: Higher is better. Color convention: The best, 2nd-best, and 3rd-best among all algorithms.

Model	GaoFen-1				WorldView-2				IKONOS			
	PSNR↑	SSIM↑	ERGAS↓	SAM↓	PSNR↑	SSIM↑	ERGAS↓	SAM↓	PSNR↑	SSIM↑	ERGAS↓	SAM↓
Brovey	34.63	0.8414	2.238	0.0256	32.25	0.8646	5.269	0.0985	35.82	0.9006	2.295	0.0524
IHS	34.80	0.8329	2.356	0.0302	32.22	0.8498	5.392	0.1051	35.68	0.8950	2.371	0.0544
SFIM	34.95	0.8221	2.278	0.0307	31.90	0.8782	5.611	0.0913	35.93	0.9078	2.277	0.0458
GS	34.16	0.8330	2.491	0.0385	32.63	0.8790	5.039	0.0958	36.28	0.9131	2.199	0.0487
GSA	34.56	0.8359	2.189	0.0359	33.53	0.8911	4.538	0.0900	37.53	0.9204	1.939	0.0437
CNMF	36.79	0.8794	1.811	0.0303	33.34	0.8988	4.586	0.0849	37.14	0.9242	1.968	0.0446
GPPNN	39.03	0.9390	1.397	0.0277	35.39	0.9368	3.699	0.0720	37.70	0.9301	1.874	0.0448
SFINet	41.79	0.9597	1.036	0.0198	35.80	0.9403	3.524	0.0677	38.99	0.9422	1.630	0.0370
MDCUN	41.69	0.9589	1.048	0.0194	33.06	0.8827	4.717	0.0888	38.98	0.9431	1.629	0.0367
UTSN	39.79	0.9504	1.271	0.0244	34.88	0.9293	3.952	0.0782	37.75	0.9295	1.885	0.0456
PGCU	39.90	0.9525	1.254	0.0266	35.31	0.9369	3.732	0.0735	37.43	0.9350	1.939	0.0463
Z-PNN	37.68	0.9022	1.629	0.0296	32.28	0.8545	5.016	0.1054	37.09	0.9224	1.964	0.0494
λ-PNN	38.14	0.9376	1.497	0.0229	33.96	0.9028	4.319	0.0913	37.91	0.9318	1.931	0.0442
PreMix-B	41.77	0.9605	1.031	0.0196	35.97	0.9420	3.464	0.0664	38.94	0.9412	1.637	0.0374
PreMix-H	43.62	0.9697	0.863	0.0169	36.54	0.9466	3.411	0.0635	40.23	0.9529	1.460	0.0326

Table 2
Quantitative comparison results on real-world full-resolution data. ↓: Lower is better. ↑: Higher is better. Color convention: The best, 2nd-best, and 3rd-best among all algorithms.

Model	GaoFen-1			WorldView-2			IKONOS		
	D_λ ↓	D_S ↓	QNR↑	D_λ ↓	D_S ↓	QNR↑	D_λ ↓	D_S ↓	QNR↑
Brovey	0.0665	0.1889	0.7591	0.0286	0.0702	0.9033	0.0291	0.1014	0.8729
IHS	0.0553	0.2071	0.7516	0.0312	0.0697	0.9013	0.0311	0.1069	0.8657
SFIM	0.0228	0.0712	0.9085	0.0270	0.0615	0.9135	0.0388	0.0878	0.8784
GS	0.0483	0.1932	0.7712	0.0202	0.0695	0.9117	0.0204	0.0933	0.8886
GSA	0.0479	0.1479	0.8143	0.0172	0.0723	0.9120	0.0255	0.0857	0.8918
CNMF	0.0190	0.1059	0.8776	0.0259	0.0748	0.9016	0.0300	0.0807	0.8924
GPPNN	0.0211	0.0752	0.9053	0.0221	0.0688	0.9107	0.0296	0.0812	0.8923
SFINet	0.0063	0.0417	0.9524	0.0091	0.0598	0.9317	0.0176	0.0723	0.9120
MDCUN	0.0035	0.0243	0.9722	0.0179	0.0490	0.9342	0.0192	0.0704	0.9122
UTSN	0.0246	0.0875	0.8901	0.0243	0.0685	0.9091	0.0366	0.0935	0.8742
PGCU	0.1531	0.0918	0.7707	0.1445	0.0997	0.7718	0.0779	0.0872	0.8437
Z-PNN	0.0194	0.0681	0.9138	0.0195	0.0691	0.9127	0.0285	0.0722	0.9014
λ-PNN	0.0092	0.0294	0.9617	0.0381	0.0717	0.8929	0.0207	0.0769	0.9040
PreMix-B	0.0065	0.0452	0.9486	0.0149	0.0642	0.9221	0.0197	0.0736	0.9088
PreMix-H	0.0042	0.0141	0.9817	0.0109	0.0612	0.9287	0.0133	0.0584	0.9296

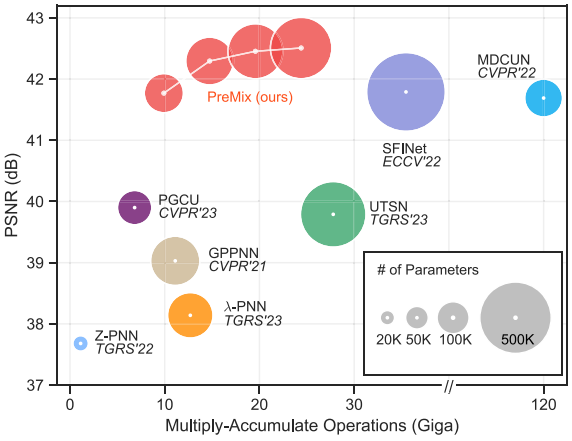


Fig. 10. PNSR v.s. Number of params (K) v.s. MACs (G). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sources exhibit significant distributional differences. Such variations are anticipated to engender discrepancies in the model's capacity to fit the different datasets. This may be attributed to the fact that in comparison to the other two datasets, IKONOS exhibits a multimodal distribution with more distinct peaks, which indicates a higher degree of variation within the data. Since we have constrained PreMix-B to the smallest scale, the limited number of parameters and single-scale filtering may struggle to capture its intrinsic patterns. In contrast to IKONOS, the distribution of the WorldView-2 dataset is more smoothing and highly concentrated. Such a distribution confers a distinct advantage for neural network optimization, which also elucidates why our model performs relatively better on the WorldView-2 dataset.

On the other hand, compared to most of other models, PreMix-B boasts a lower parameter count, aiming to evaluate the performance of the model at the smallest scale. The PreMix model introduced in this paper necessitates the customization of a specific convolutional kernel for each pixel and its neighbors to execute predictive filtering. However, if the model is endowed with an insufficient number of parameters, it may fail to capture the intricate patterns and relationships within the data, hindering the learning of inherent features and characteristics. Consequently, this could impede the model's capacity to learn effective kernels, potentially leading to suboptimal results.

The two unsupervised models, Z-PNN (Ciotola et al., 2022) and λ-PNN (Ciotola et al., 2023), demonstrate a diminished comparability with supervised methods when evaluated at reduced-resolution. This stems from their training on full-resolution dataset, which does not align with the data distribution at lower resolutions. In contrast, supervised learning models are trained at reduced-resolution and thus can naturally adapt to the distribution at reduced scales. Unlike supervised models, unsupervised learning models do not directly learn the mapping from inputs to predictions; instead, they indirectly learn spatial super-resolution from PAN images and spectral preservation from LRMS

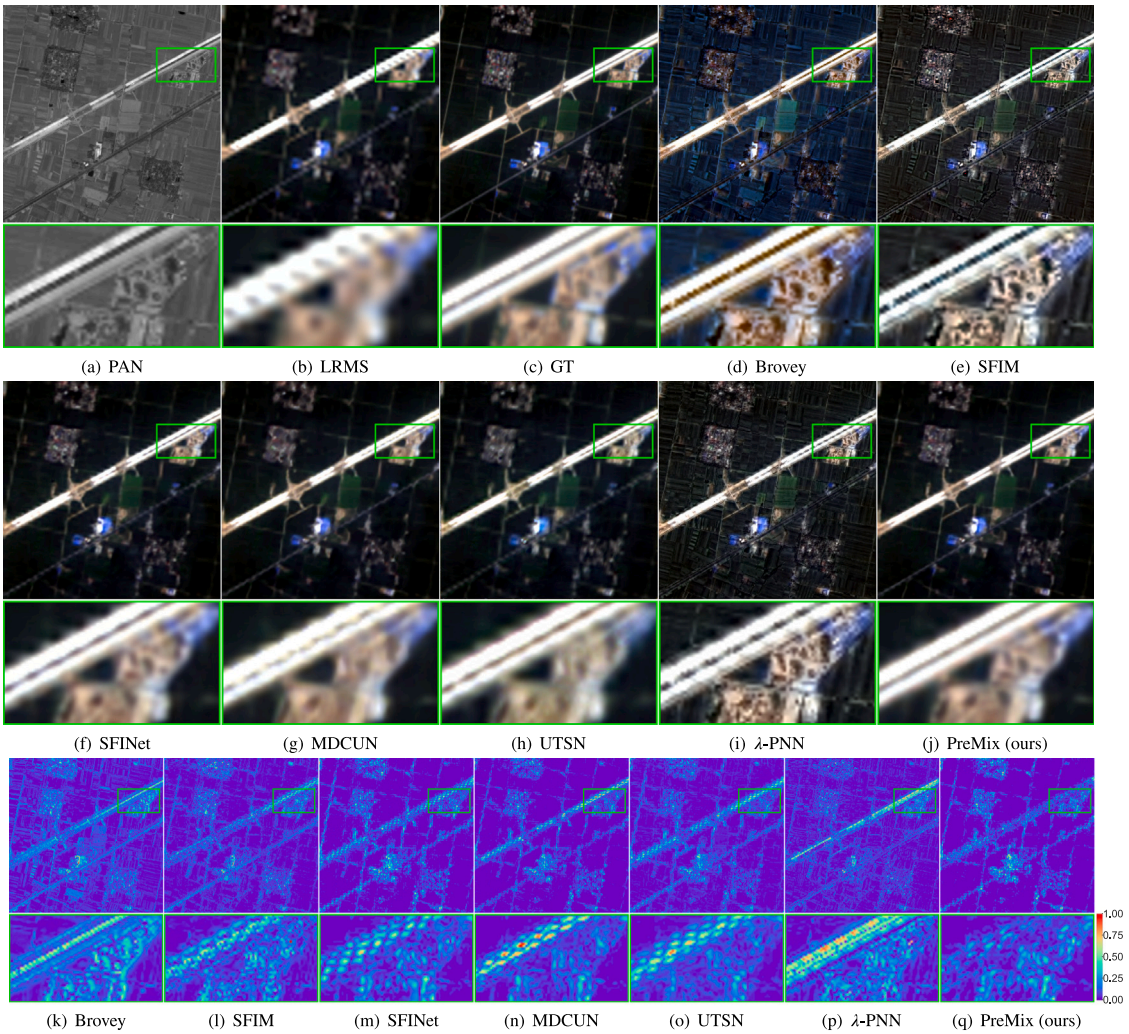


Fig. 11. Visual comparison on GaoFen-1 dataset. The bright rectangular box represents the area zoomed in for display. The last row is the mean absolute error between prediction and GT.

images. The divergence in learning strategies underscores a great gap between the performance capabilities of supervised and unsupervised models.

4.2.2. Evaluation at full-resolution

Table 2 presents the comparison results of non-reference metrics for the full-resolution real-data experiments. Our PreMix-B does not generalize well to full scale compared to other models. The PreMix-B model employs single-scale filtering and is equipped with only a single layer of the EWFM module. Simplified models may lack the requisite parameters or architectural complexity to capture the intricacies and patterns present in the data, thus having limited expressive capability and generalizability. Experimental results on varying the number of EWFM layers indicate that performance can indeed be further improved by adding more layers (refer to Section 4.4.1 Fig. 20). This finding underscores the model's potential to enhance its predictive accuracy and generalizability through increased complexity. However, we have intentionally constrained the parameter count of PreMix-B to be relatively small. One layer and single filtering is intended to validate the model's performance at a minimal scale. This is also primarily to maintain a comparable parameter volume with models like MDCUN (Yang et al., 2022) and PGCU (Zhu et al., 2023), thereby ensuring a fair comparison. On the other hand, PreMix-H achieves better results, indicating the effectiveness of the proposed multi-scale and multi-branch progressive filtering. Compared to the second-best model, PreMix-H improves QNR by 0.95% and 1.74% on the GaoFen-1 and IKONOS datasets, respectively. However, on the WorldView-2 dataset, SFINet (Zhou et al., 2022b) and MDCUN (Yang et al., 2022) remain competitive models and achieve better results compared to PreMix-H. Compared to supervised models, the unsupervised Z-PNN (Ciotola et al., 2022) λ-PNN (Ciotola et al., 2023) excel in spatial preservation, but its spectral preservation is less satisfactory. The discrepancy may be attributed to the meticulous design of the loss function.

4.2.3. Evaluation of complexity

Table 3 presents the number of parameters, the computational cost (multiply-accumulate operations, MACs), and processing time per image for all DL-based models. In Table 1, PreMix-B exhibits similar performance to SFINet (Zhou et al., 2022b) and MDCUN (Yang et al., 2022), but outperforms them in terms of lower computation and parameters as well as shorter runtime, according to Table 3. This demonstrates the superiority of the proposed method in this paper. Although PreMix-H has more parameters, its computational cost and inference time are lower than that of SFINet (Zhou et al., 2022b) and MDCUN (Yang et al., 2022). Taking SFINet (Zhou et al., 2022b) as an example, the inference time for our model is 8.734 ms per image, significantly outperforming 23.90 ms per image of SFINet (Zhou et al., 2022b). This is mainly attributed to the extensive use of operations in the image frequency domain and various attention tricks

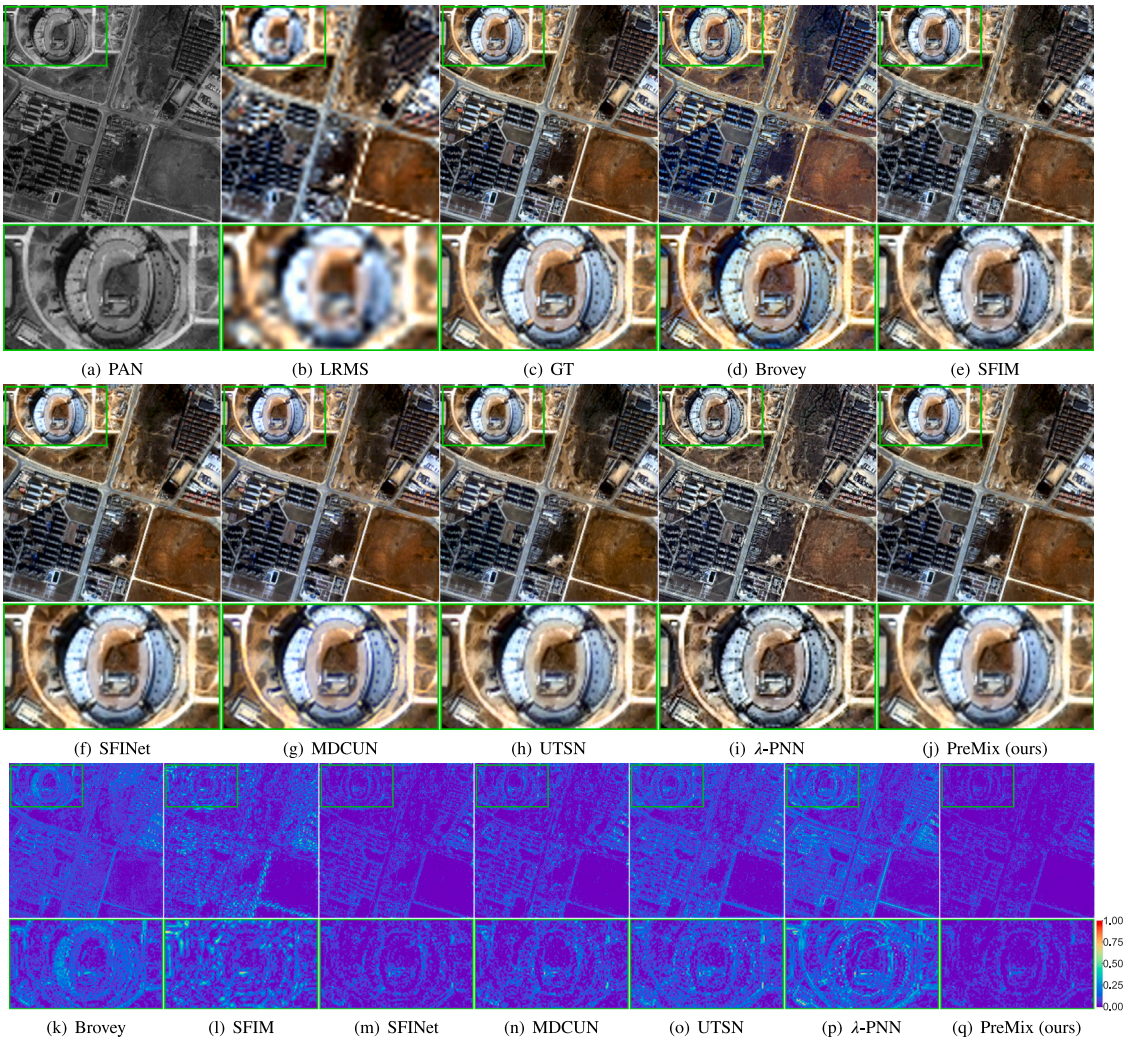


Fig. 12. Visual comparison on IKONOS dataset. The bright rectangular box represents the area zoomed in for display. The last row is the mean absolute error between prediction and GT.

Table 3
Comparison of model complexity and inference time. MACs was obtained under the condition of predicting images with 4 bands 256 × 256 resolution and a batch of 1. The inference time was obtained by calculating the average time taken by the model to perform forward inference on all images in the testing set.

Model	Params(M)	MACs(G)	Time (ms/img)
GPPNN	0.239	11.11	7.586
SFINet	0.611	35.47	23.90
MDCUN	0.141	117.8	15.59
UTSN	0.424	27.79	2.454
PGCU	0.116	6.823	8.991
Z-PNN	0.023	1.123	1.816
λ-PNN	0.204	12.70	2.983
PreMix-B	0.151	9.909	2.800
PreMix-H	0.865	35.03	8.734

of SFINet (Zhou et al., 2022b), which are more computationally intensive. In contrast, the model presented in this paper relies predominantly on convolutions, which are efficiently supported by PyTorch and CUDA, thus providing a substantial speed advantage despite the higher parameter count.

To further illustrate the superiority of the proposed algorithm in this paper and provide a clearer visual contrast, we plotted the relationship between model parameters, MACs and PSNR, as shown in Fig. 10. Each colored point in the graph corresponds to a distinct model, with the point's diameter indicative of the model's parameter count. The closer a point is to the upper left corner and the smaller its diameter, the more superior the model's performance is considered. The red circles represents the performance of PreMix-B equipped with 2 to 5 `EWFM` layers. It can be observed from the figure that the proposed model in this paper achieves superior performance compared to other models with fewer parameters and computational costs. To enhance the model's capacity for learning effective and robust kernels, it is necessary to enable deeper layers and more branches, which inevitably leads to increased complexity and longer inference time. A large number of parameters can lead to overfitting when directly learning the mapping from LRMS and PAN to HRMS (Fig. 1(a)). Because the network may directly memorize the mapping from inputs to predictions. However, in the method proposed in this paper, the increase in the number of parameters is primarily utilized to learn more effective predictive kernels that are applied to LRMS and PAN (Fig. 1(b)), thereby indirectly obtaining HRMS. The proposed strategy facilitates an indirect acquisition of HRMS and can mitigate the risk of overfitting. This is because the complexity of memorizing the kernels for filtering LRMS and PAN is inherently greater than that of memorizing HRMS.

4.3. Visual comparison

In this section, we examine the visual quality and fidelity of the output, and offer a qualitative analysis of spatial and spectral characteristics of different methods.

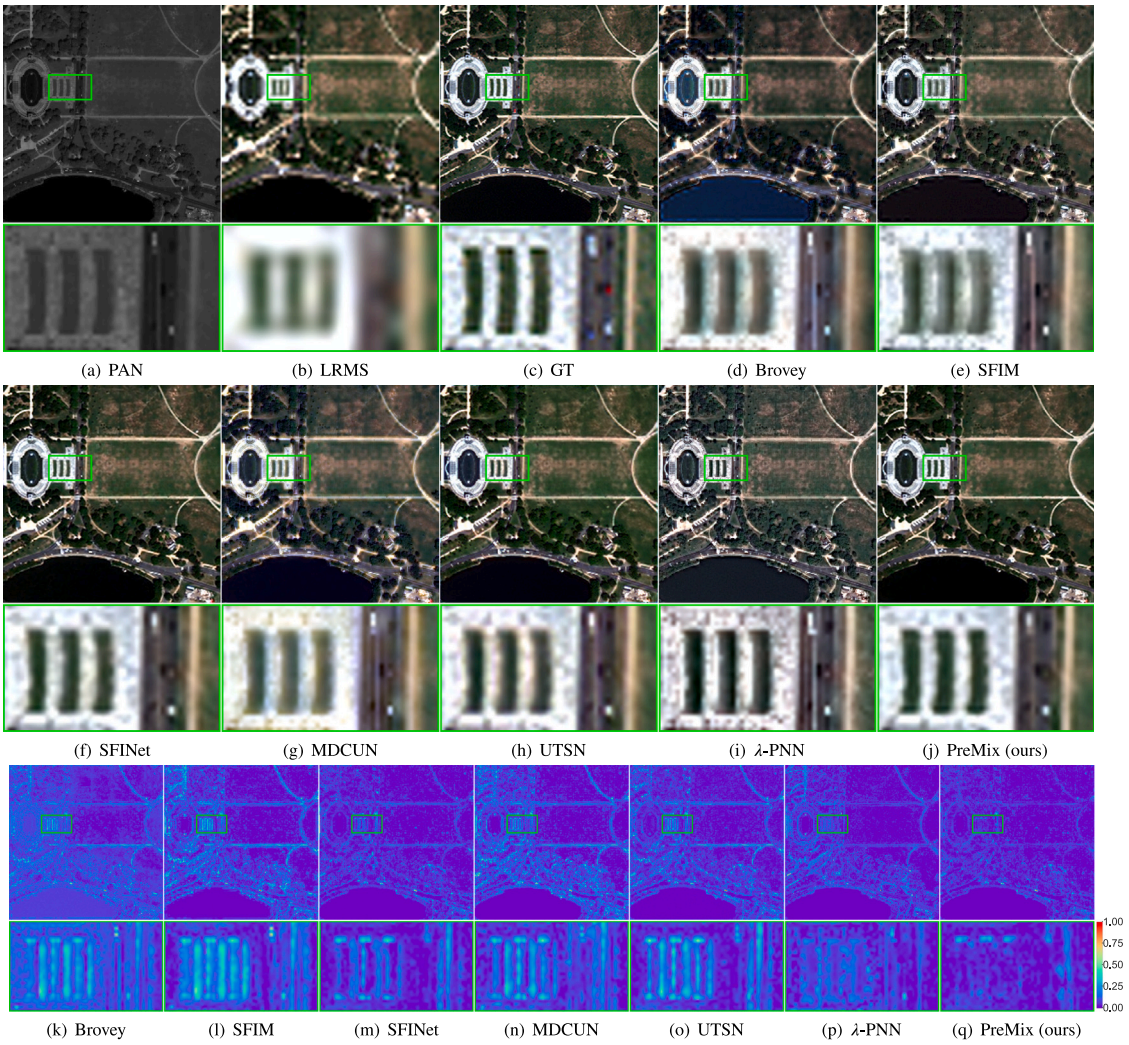


Fig. 13. Visual comparison on WorldView-2 dataset. The bright rectangular box represents the area zoomed in for display. The last row is the mean absolute error between prediction and GT. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3.1. Evaluation at reduced-resolution

Figs. 11, 12, and 13 present the visual comparison of reduced-resolution on the GaoFen-1, IKONOS and WorldView-2, respectively. To further clarify the superiority of the proposed model in this paper, we also generate the mean absolute error images and local zoomed-in images between the prediction and GT. Clearly, the model proposed in this paper focuses more on spatial details (*e.g.*, edges in the zoomed-in area), demonstrating smaller reconstruction error. While SFINet (Zhou et al., 2022b) also shows good performance, it slightly lags behind the proposed PreMix model in terms of image detail preservation.

4.3.2. Evaluation at full-resolution

Figs. 14, 15, and 16 present the visual comparison results at full-resolution of WorldView-2, GaoFen-1, and IKONOS, respectively. To assess the generalizability and practicality on out-of-dataset images, Fig. 15 is sourced from another high-resolution dataset (Zhang et al., 2023). In traditional algorithms, Brovey (Gillespie et al., 1987), IHS (Rahmani et al., 2010), and CNMF (Yokoya et al., 2012) exhibit noticeable spectral distortions, while GS (Laben and Brower, 2000) and GSA (Aiazzi et al., 2007) yield more blurring results. Among all DL-based models, GPPNN (Xu et al., 2021), UTSN (Sheng et al., 2023), and PGCU (Zhu et al., 2023) lack the ability to generalize to full-resolution and lead to significant spectral distortions, especially PGCU (Zhu et al., 2023). The results produced by MDCUN (Yang et al., 2022) are relatively blurred. Both SFINet (Zhou et al., 2022b) and the proposed PreMix model perform well, but compared to SFINet (Zhou et al., 2022b), our model excels in details, such as the *tennis court* in the top left corner of the zoomed-in image in Fig. 14.

Table 4
Ablation of EWFM and PF. ↓: Lower is better. ↑: Higher is better.

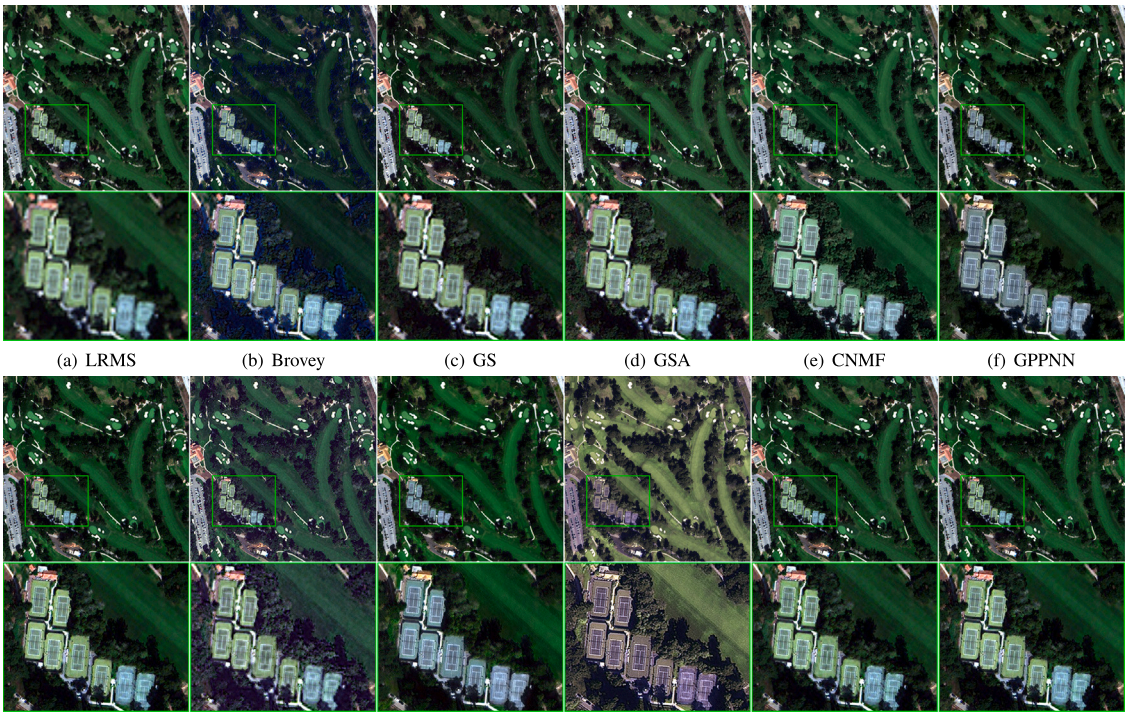
Configuration		Metric			
EWFM	PF	PSNR↑	SSIM↑	ERGAS↓	SAM↓
✗	✗	40.76	0.9530	1.144	0.0221
✓	✗	41.66	0.9617	1.040	0.0205
✗	✓	41.41	0.9569	1.066	0.0199
✓	✓	**42.51**	**0.9655**	**0.953**	**0.0184**

4.4. Ablation study

We evaluate the contribution of every individual component to the overall performance via ablation study, including EWFM and PF, different branches, and different scales.

4.4.1. Ablation of EWFM and PF

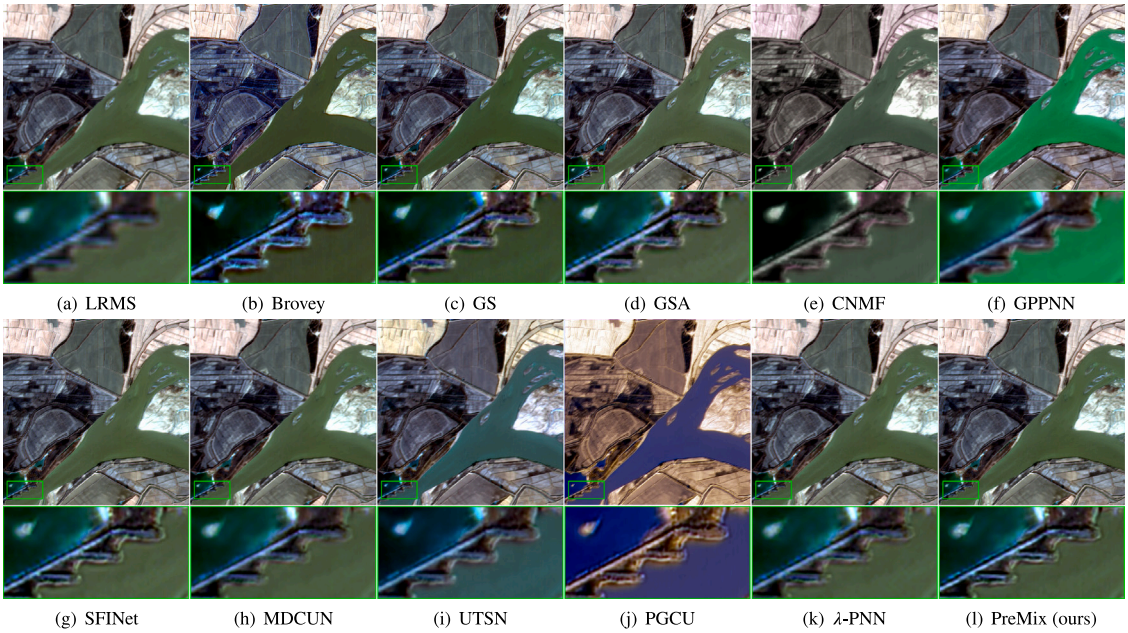
To validate the effectiveness of the proposed EWFM and PF in this study, we conducted ablation experiments on GaoFen-1 dataset. The experiment utilized PreMix-B with mask generation by Eq. (13), and employed hard mixing with 4 layers of EWFM blocks. The experimental



(a) LRMS (b) Brovey (c) GS (d) GSA (e) CNMF (f) GPPNN

(g) SFINet (h) MDCUN (i) UTSN (j) PGCU (k) λ-PNN (l) PreMix (ours)

Fig. 14. Visual comparison on WorldView-2 dataset at full-resolution. The bright rectangular box represents the area zoomed in for display.



(a) LRMS (b) Brovey (c) GS (d) GSA (e) CNMF (f) GPPNN

(g) SFINet (h) MDCUN (i) UTSN (j) PGCU (k) λ-PNN (l) PreMix (ours)

Fig. 15. Visual comparison on GaoFen-1 dataset at full-resolution. The bright rectangular box represents the area zoomed in for display.

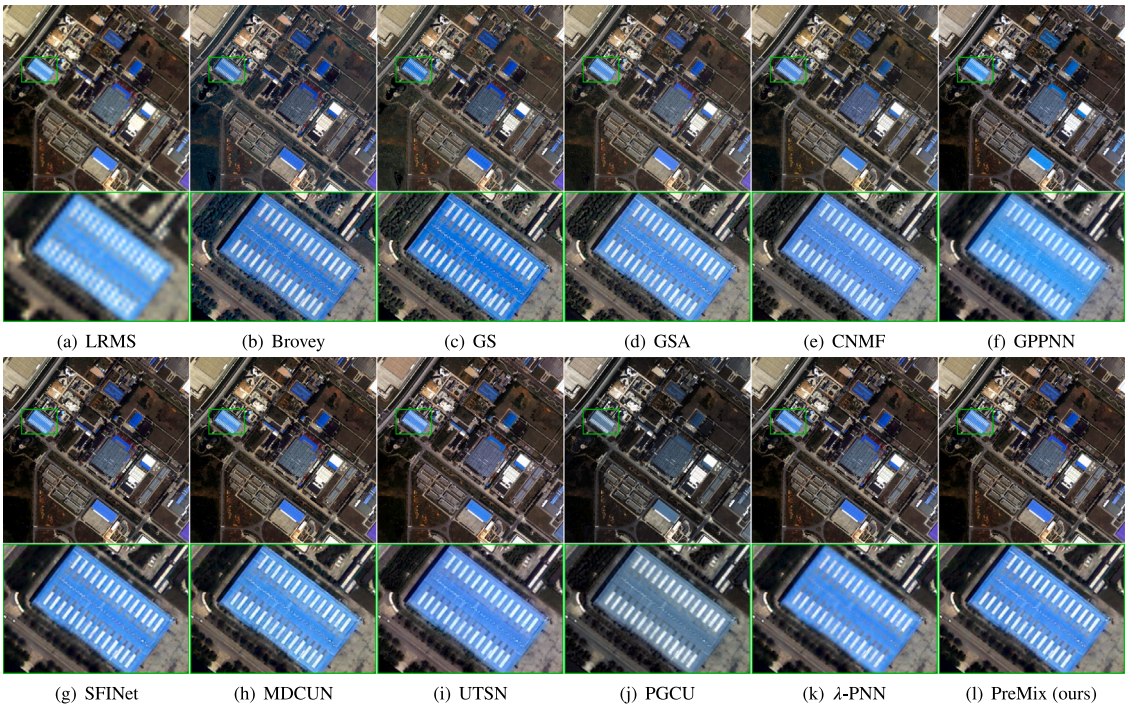
results are shown in Table 4. Clearly, the performance is poorest when neither EWFM nor PF are used. Adding only EWFM module without PF yields better results than adding only PF without EWFM. The best performance is achieved when both are incorporated.

4.4.2. Ablation of multi-branch

To validate the importance of the multi-branch approach proposed in this study, Table 5 presents the ablation results of the histogram equalization (HE) branch and the high pass (HP) filtering branch in PreMix-H. The results indicate that the best performance is achieved when both branches are enabled simultaneously.

Table 5
Ablation of different branches. ↓: Lower is better. ↑: Higher is better.

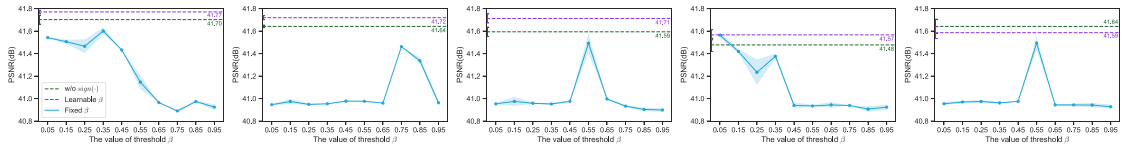
Configuration		Metric			
HP	HE	PSNR↑	SSIM↑	ERGAS↓	SAM↓
✗	✗	42.45	0.9634	0.967	0.0186
✓	✗	43.17	0.9679	0.897	0.0175
✗	✓	43.49	0.9685	0.879	0.0172
✓	✓	**43.62**	**0.9697**	**0.863**	**0.0169**



(a) LRMS (b) Brovey (c) GS (d) GSA (e) CNMF (f) GPPNN

(g) SFINet (h) MDCUN (i) UTSN (j) PGCU (k) λ-PNN (l) PreMix (ours)

Fig. 16. Visual comparison on IKONOS dataset at full-resolution. The bright rectangular box represents the area zoomed in for display..



(a) $\texttt{Tanh}(\texttt{ReLU}(x))$ (b) $\texttt{Tanh}(\texttt{ELU}(x)+1)$ (c) $\texttt{Sigmoid}(x)$ (d) $\texttt{Softsign}(\texttt{ReLU}(x))$ (e) $\texttt{Softsign}(\texttt{ELU}(x)+1)$

Fig. 17. On the influence of the threshold β.

Table 6
Ablation of different scales. ↓: Lower is better. ↑: Higher is better.

Configuration	Metric			
	PSNR↑	SSIM↑	ERGAS↓	SAM↓
1x	41.72	0.9590	1.039	0.0198
1x,2x	43.18	0.9679	0.891	0.0175
1x,2x,4x	**43.62**	**0.9697**	**0.863**	**0.0169**

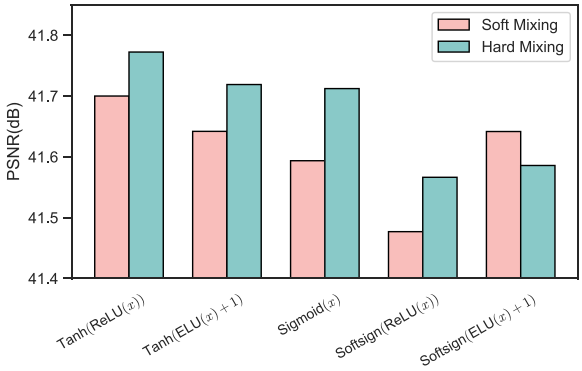


Fig. 18. On the influence of mask generation functions.

4.4.3. Ablation of multi-scale

To validate the importance of multi-scale inputs, Table 6 presents the ablation results on different scales of inputs in PreMix-H. The results indicate that the best performance is achieved when the original scale, scale of 2×, and scale of 4× are used simultaneously.

4.5. Parameter analysis

To validate that the mask generation method proposed in this study is superior to the commonly used sigmoid function and that hard mixing is an effective strategy, we compared the impact of different mask generation methods on the predicted results. Additionally, we analyzed the influence of the number of layers in EWFM and the size of the prediction kernel in PF in PreMix-B on the results.

Fig. 17 presents a comparison between manually setting the thresholds in the step function and utilizing the proposed trainable thresholds. Enabling the hard mixing strategy with trainable thresholds yields even better results, demonstrating the effectiveness of the proposed method. Fig. 18 compares the PSNR under different mixing strategies on GaoFen-1, with EWFM set to a depth of 2. It can be observed that, compared to the sigmoid, $\texttt{Tanh}(\texttt{ReLU}(x))$ exhibits better performance, highlighting the importance of sparse activation. To further analyze the impacts of different mask generation methods, we provide visual images of masks averaged across all channels generated by representative approaches, as shown in Fig. 19. β in $\texttt{Sign}(\sigma(x))$ is manually set to 0.3. The results align with our previous analysis: masks generated by sigmoid tend to have similar values at each layer, while $\texttt{Tanh}(\texttt{ReLU}(x))$ produce more distinctive values at each layer. The poor performance of manually set thresholds is attributed to the masks gradually converging to 0 or 1 as the depth increases.

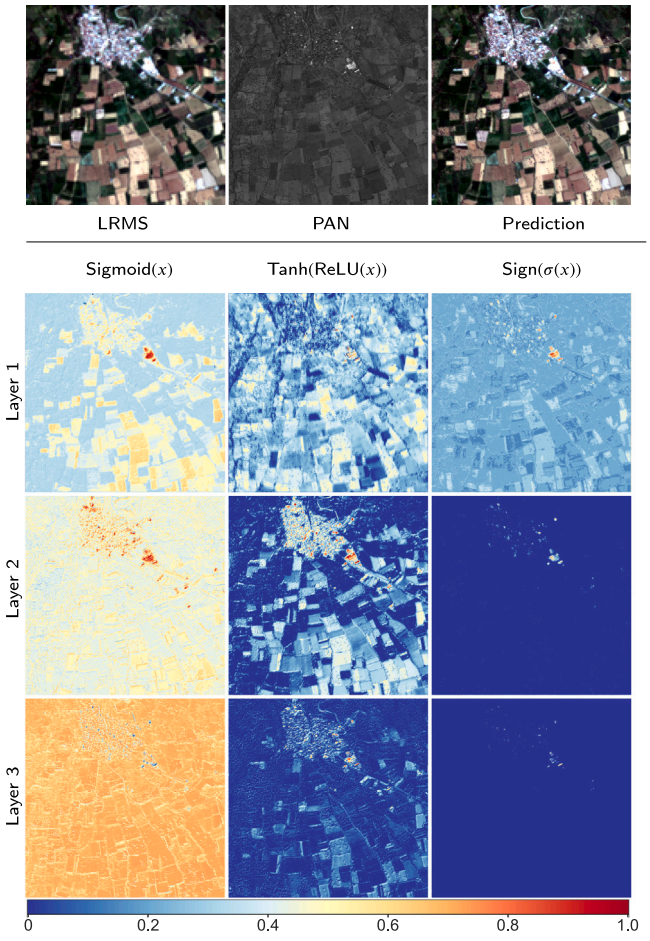


Fig. 19. Visualization of masks. Masks generated by the `Sigmoid` function exhibit a tendency to possess similar values, whereas the composition `Tanh(ReLU(`x`))` yields a broader range of values. The suboptimal performance of manually set thresholds results from converging towards binary values of 0 or 1.

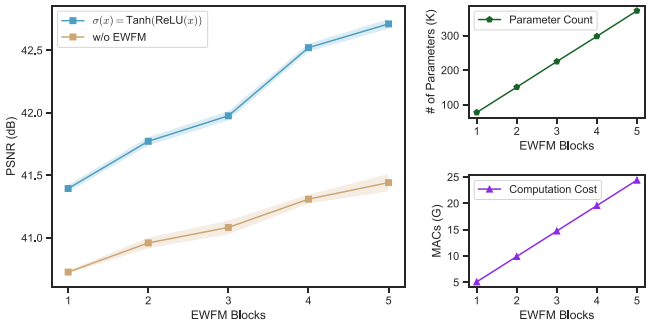


Fig. 20. On the influence of the number of `EWFM` layers.

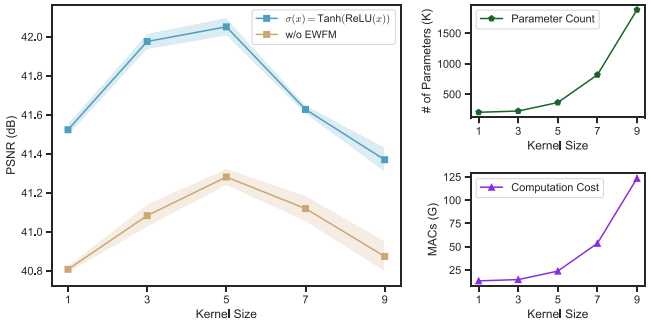


Fig. 21. On the influence of the size of predictive kernels.

Figs. 20 and 21 show the impact of different numbers of `EWFM` layers and different sizes of predictive kernels on the predicted results. Generally, as the number of layers increases, the model performance improves. However, a similar assertion does not hold true for the size of the predictive kernel. The model achieves the best performance when the predictive kernel size is 5, but a significant performance drop occurs with further increases in kernel size. This could be due to the local spatial correlation of images, where excessively large kernels struggle to learn relationships across larger areas, leading to suboptimal results. On the other hand, the number of parameters and computational cost of the proposed model increase linearly with the number of `EWFM` blocks. However, it exhibit a quadratic complexity with the increase in kernel size. In summary, increasing the number of `EWFM` layers is a more economical and effective choice compared to enlarging the kernel size.

Regarding the determination of the optimal number of `EWFM` layers, it is related to multiple factors, such as the distribution of datasets and the available computational resources. Firstly, larger datasets with greater complexity may require a higher number of `EWFM` blocks to effectively process the complicated information. Conversely, smaller and less complex datasets could suffer from overfitting if an excessive number of `EWFM` layers are employed. According to Occam's Razor, a trade-off must be made between the complexity of the model and its effectiveness. The more complex the model, the stronger its fitting capability, but also the stronger its memorization ability, which can lead to a significant decline in generalizability. In such cases, simpler models may demonstrate superior performance. Secondly, neural networks are inherently classified as black-box optimization models. The number of layers of `EWFM` block is a key hyperparameter which typically necessitates fine-tuning based on empirical evidence. Alternatively, neural architecture search (NAS) (Elsken et al., 2019) can be utilized to autonomously determine the optimal count of `EWFM` blocks. Nonetheless, this approach inevitably requires additional time and computational resources. Compared with automatic search, another solution is to determine the number of EWFMs through empirical experiments as well as considering preconditions such as dataset characteristics and computility.

4.6. Discussion of limitation

One inherent limitation observed in the pansharpening process is the loss of detailed spectral information of small objects (see *the red car* in Fig. 13). Despite advancements in predictive filtering for HRMS generation, the challenge of accurately recovering the correct spectrum of small objects remains a common open issue across all models. The LRMS may lack sufficient spatial resolution to capture fine details of small objects, consequently affecting the ability of the predicted HRMS to accurately restore their spectral characteristics. This inherent limitation highlights the need for further research and innovative techniques to address the accurate reconstruction of spectral information for small objects. Future works may consider strategies for spectral preservation and strive to design a framework to enhance the recovery of spectral information of small objects.

5. Conclusion

In this study, we proposed a pansharpening method based on predictive filtering technique aimed at minimizing information loss and reducing spatial and spectral distortions. By introducing the fine-grained feature fusion method `EWFM` and a multi-scale multi-branch progressive filtering network, we achieved efficient feature mixing of LRMS and PAN, as well as more accurate prediction kernel acquisition, and

ultimately resulted in superior performance. In both simulated experiments and full-resolution experiments, our proposed model excelled in various evaluation metrics, particularly showing significant improvements in PSNR and SSIM. Compared to other DL-based models, our model demonstrated outstanding performances in terms of effectiveness, parameter count, and computational cost. Further ablation studies confirmed the effectiveness of the proposed approach and underscored the importance of EWFM and PF modules. Our model showcases innovation and superiority in the field of pansharpening, providing valuable insights for future research and applications.

For future work, we aim to explore even more refined feature fusion methods and continue optimizing the model structure to enhance efficiency and accuracy. Additionally, we plan to consider extending the application of this predictive filtering technique to other remote sensing domains, aiming for broader impact and utility.

CRediT authorship contribution statement

Yongchuan Cui: Writing – original draft, Visualization, Methodology, Investigation, Conceptualization. **Peng Liu:** Resources, Project administration, Funding acquisition, Methodology, Data curation, Conceptualization. **Yan Ma:** Validation, Software. **Lajiao Chen:** Validation, Software. **Mengzhen Xu:** Writing – review & editing, Visualization, Formal analysis. **Xingyan Guo:** Writing – review & editing, Validation, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under Grant 41971397, Grant 41471368, and Grant U2243222, and the project Y1H103101A and Y5J0100.

References

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

Aiazzi, B., Alparone, L., Baronti, S., Garzelli, A., Selva, M., 2003. An MTF-based spectral distortion minimizing model for pan-sharpening of very high resolution multispectral images of urban areas. In: 2003 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas. pp. 90–94.

Aiazzi, B., Alparone, L., Baronti, S., Garzelli, A., Selva, M., 2006. MTF-tailored multiscale fusion of high-resolution MS and pan imagery. Photogramm. Eng. Remote Sens. 72, 591–596.

Aiazzi, B., Baronti, S., Selva, M., 2007. Improving component substitution pansharpening through multivariate regression of MS +pan data. IEEE Trans. Geosci. Remote Sens. 45 (10), 3230–3239.

Alparone, L., Aiazzi, B., Baronti, S., Garzelli, A., Nencini, F., Selva, M., 2008. Multispectral and panchromatic data fusion assessment without reference. Photogramm. Eng. Remote Sens. 74, 193–200.

Bako, S., Vogels, T., Mcwilliams, B., Meyer, M., NováK, J., Harvill, A., Sen, P., Derose, T., Rousselle, F., 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. ACM Trans. Graph. 36 (4).

Ballester, C., Caselles, V., Igual, L., Verdera, J., Rougé, B., 2006. A variational model for P+XS image fusion. Int. J. Comput. Vis. 69 (1), 43–58.

Burt, P., Adelson, E., 1983. The Laplacian pyramid as a compact image code. IEEE Trans. Commun. 532–540.

Chavez, P., Kwarteng, A., 1989. Extracting spectral contrast in landsat thematic mapper image data using selective principal component analysis. Photogramm. Eng. Remote Sens..

Cho, W., Son, S., Kim, D.-S., 2021. Weighted multi-kernel prediction network for burst image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 404–413.

Ciotola, M., Poggi, G., Scarpa, G., 2023. Unsupervised deep learning-based pansharpening with jointly enhanced spectral and spatial fidelity. IEEE Trans. Geosci. Remote Sens. 61, 1–17.

Ciotola, M., Vitale, S., Mazza, A., Poggi, G., Scarpa, G., 2022. Pansharpening by convolutional neural networks in the full resolution framework. IEEE Trans. Geosci. Remote Sens. 1.

Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In: ArXiv.

Cui, Y., Liu, P., Song, B., Zhao, L., Ma, Y., Chen, L., 2024. Reconstruction of large-scale missing data in remote sensing images using Extend-GAN. IEEE Geosci. Remote Sens. Lett. 21, 1–5.

Dian, R., Li, S., 2019. Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization. IEEE Trans. Image Process. 5135–5146.

Elsken, T., Metzen, J.H., Hutter, F., 2019. Neural architecture search: A survey. J. Mach. Learn. Res. 20 (55), 1–21.

Fu, L., Zhou, C., Guo, Q., Juefei-Xu, F., Yu, H., Feng, W., Liu, Y., Wang, S., 2021. Auto-exposure fusion for single-image shadow removal. In: IEEE Conference on Computer Vision and Pattern Recognition.

Garzelli, A., Nencini, F., Capobianco, L., 2008. Optimal MMSE pan sharpening of very high resolution multispectral images. IEEE Trans. Geosci. Remote Sens. 228–236.

Gillespie, A.R., Kahle, A.B., Walker, R.E., 1987. Color enhancement of highly correlated images. II. Channel ratio and "chromaticity" transformation techniques. Remote Sens. Environ. 22 (3), 343–365.

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. In: Gordon, G., Dunson, D., Dudí k, M. (Eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. In: Proceedings of Machine Learning Research, vol. 15, PMLR, Fort Lauderdale, FL, USA, pp. 315–323.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial networks. Adv. Neural Inf. Process. Syst. 3.

Guo, Q., Li, X., Juefei-Xu, F., Yu, H., Liu, Y., Wang, S., 2021a. JPGNet: Joint predictive filtering and generative network for image inpainting. In: Proceedings of the 29th ACM International Conference on Multimedia. In: ACM International Conference on Multimedia, Association for Computing Machinery, New York, NY, USA, pp. 386–394.

Guo, Q., Sun, J., Juefei-Xu, F., Ma, L., Xie, X., Feng, W., Liu, Y., 2021b. EfficientDeRain: Learning pixel-wise dilation filtering for high-efficiency single-image deraining. In: AAAI Conference on Artificial Intelligence.

He, L., Fang, Z., Li, J., Chanussot, J., Plaza, A., 2024. Two spectral–spatial implicit neural representations for arbitrary-resolution hyperspectral pansharpening. IEEE Trans. Geosci. Remote Sens. 62, 1–21.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA, USA, pp. 770–778.

Iliev, A.I., Kyurkchiev, N., Markov, S., 2015. On the approximation of the cut and step functions by logistic and gompertz functions. BIOMATH.

Iliev, A., Kyurkchiev, N., Markov, S., 2017. On the approximation of the step function by some sigmoid functions. Math. Comput. Simulation 223–234.

Jiang, M., Shen, H., Li, J., Yuan, Q., Zhang, L., 2020. A differential information residual convolutional neural network for pansharpening. ISPRS J. Photogramm. Remote Sens. 163, 257–271.

Kingma, D., Ba, J., 2015. Adam: A method for stochastic optimization. In: International Conference on Learning Representations. San Diega, CA, USA.

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

Li, X., Guo, Q., Lin, D., Li, P., Feng, W., Wang, S., 2022b. MISF:Multi-level interactive siamese filtering for high-fidelity image inpainting. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1859–1868.

Li, S., Yin, H., Fang, L., 2013. Remote sensing image fusion via sparse representations over learned dictionaries. IEEE Trans. Geosci. Remote Sens. 51 (9), 4779–4789.

Li, K., Zhang, W., Yu, D., Tian, X., 2022a. HyperNet: A deep network for hyperspectral, multispectral, and panchromatic image fusion. ISPRS J. Photogramm. Remote Sens. 188, 30–44.

Lin, H., Dong, Y., Ding, X., Liu, T., Liu, Y., 2024. Unsupervised pan-sharpening via mutually guided detail restoration. AAAI Conf. Artif. Intell. 38, 3386–3394.

Liu, J.G., 2000. Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details. Int. J. Remote Sens. 21 (18), 3461–3472.

Liu, P., Li, J., Wang, L., He, G., 2022a. Remote sensing data fusion with generative adversarial networks: State-of-the-art methods and future research directions. IEEE Geosci. Remote Sens. Mag. 10 (2), 295–328.

Liu, Q., Meng, X., Shao, F., Li, S., 2023. Supervised-unsupervised combined deep convolutional neural networks for high-fidelity pansharpening. Inf. Fusion 89, 292–304.

Liu, P., Wang, L., Ranjan, R., He, G., Zhao, L., 2022b. A survey on active deep learning: from model driven to data driven. ACM Comput. Surv. 54 (10s), 1–34.

Liu, P., Xiao, L., Tang, S., 2016a. A new geometry enforcing variational model for pan-sharpening. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 9 (12), 5726–5739.

Liu, P., Xiao, L., Zhang, J., Naz, B., 2016b. Spatial-hessian-feature-guided variational model for pan-sharpening. IEEE Trans. Geosci. Remote Sens. 54 (4), 2235–2253.

Liu, Q., Zhou, H., Xu, Q., Liu, X., Wang, Y., 2020. PSGAN: A generative adversarial network for remote sensing image pan-sharpening. IEEE Trans. Geosci. Remote Sens. 59 (12), 10227–10242.

Ma, J., Yu, W., Chen, C., Liang, P., Guo, X., Jiang, J., 2020. Pan-GAN: An unsupervised pan-sharpening method for remote sensing image fusion. Inf. Fusion 62, 110–120.
Masi, G., Cozzolino, D., Verdoliva, L., Scarpa, G., 2016. Pansharpening by convolutional neural networks. Remote Sens. 8 (7).
Meng, X., Shen, H., Li, H., Zhang, L., Fu, R., 2019. Review of the pansharpening methods for remote sensing images based on the idea of meta-analysis: Practical discussion and challenges. Inf. Fusion 102–113.
Meng, X., Xiong, Y., Shao, F., Shen, H., Sun, W., Yang, G., Yuan, Q., Fu, R., Zhang, H., 2021. A large-scale benchmark data set for evaluating pansharpening performance: Overview and implementation. IEEE Geosci. Remote Sens. Mag. 9 (1), 18–52.
Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R., 2018. Burst denoising with kernel prediction networks. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2502–2510.
Ozcelik, F., Alganci, U., Sertel, E., Unal, G., 2021. Rethinking CNN-based pansharpening: Guided colorization of panchromatic images via GANs. IEEE Trans. Geosci. Remote Sens. 59 (4), 3486–3501.
Rahmani, S., Strait, M., Merkurjev, D., Moeller, M., Wittman, T., 2010. An adaptive IHS pan-sharpening method. IEEE Geosci. Remote Sens. Lett. 7 (4), 746–750.
Scarpa, G., Vitale, S., Cozzolino, D., 2018. Target-adaptive CNN-based pansharpening. IEEE Trans. Geosci. Remote Sens. 56 (9), 5443–5457.
Shao, Z., Lu, Z., Ran, M., Fang, L., Zhou, J., Zhang, Y., 2020. Residual encoder–decoder conditional generative adversarial network for pansharpening. IEEE Geosci. Remote Sens. Lett. 17 (9), 1573–1577.
Shen, K., Yang, X., Lolli, S., Vivone, G., 2023. A continual learning-guided training framework for pansharpening. ISPRS J. Photogramm. Remote Sens. 196, 45–57.
Sheng, Z., Zhang, F., Sun, J., Tan, Y., Zhang, K., Bruzzone, L., 2023. A unified two-stage spatial and spectral network with few-shot learning for pansharpening. IEEE Trans. Geosci. Remote Sens. 61, 1–17.
Thanh Nhat Mai, T., Lam, E.Y., Lee, C., 2024. Deep unfolding tensor rank minimization with generalized detail injection for pansharpening. IEEE Trans. Geosci. Remote Sens. 62, 1–18.
Vivone, G., Dalla Mura, M., Garzelli, A., Pacifici, F., 2021a. A benchmarking protocol for pansharpening: Dataset, preprocessing, and quality assessment. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 6102–6118.
Vivone, G., Dalla Mura, M., Garzelli, A., Restaino, R., Scarpa, G., Ulfarsson, M.O., Alparone, L., Chanussot, J., 2021b. A new benchmark based on recent advances in multispectral pansharpening: Revisiting pansharpening with classical and emerging pansharpening methods. IEEE Geosci. Remote Sens. Mag. 53–81.
Wald, L., Ranchin, T., Mangolini, M., 1997. Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images. Photogramm. Eng. Remote Sens. 63, 691–699.
Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.
Wang, Y., He, X., Dong, Y., Lin, Y., Huang, Y., Ding, X., 2024. Cross-modality interaction network for pan-sharpening. IEEE Trans. Geosci. Remote Sens. 62, 1–16.
Xia, Z., Perazzi, F., Gharbi, M., Sunkavalli, K., Chakrabarti, A., 2020. Basis prediction networks for effective burst denoising with large kernels. In: IEEE Conference on Computer Vision and Pattern Recognition.
Xing, Y., Wang, M., Yang, S., Jiao, L., 2018. Pan-sharpening via deep metric learning. ISPRS J. Photogramm. Remote Sens. 145, 165–183, Deep Learning RS Data.
Xiong, Z., Liu, N., Wang, N., Sun, Z., Li, W., 2023. Unsupervised pansharpening method using residual network with spatial texture attention. IEEE Trans. Geosci. Remote Sens. 61, 1–12.
Xu, Q., Li, Y., Nie, J., Liu, Q., Guo, M., 2023. UPanGAN: Unsupervised pansharpening based on the spectral and spatial loss constrained generative adversarial network. Inf. Fusion 91, 31–46.
Xu, S., Zhang, J., Zhao, Z., Sun, K., Liu, J., Zhang, C., 2021. Deep gradient projection networks for pan-sharpening. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1366–1375.
Yang, J., Fu, X., Hu, Y., Huang, Y., Ding, X., Paisley, J., 2017. PanNet: a deep network architecture for pan-sharpening. In: IEEE International Conference on Computer Vision. pp. 1753–1761.
Yang, S., Zhang, K., Wang, M., 2018. Learning low-rank decomposition for pan-sharpening with spatial-spectral offsets. IEEE Trans. Neural Netw. Learn. Syst. 29 (8), 3647–3657.
Yang, G., Zhou, M., Yan, K., Liu, A., Fu, X., Wang, F., 2022. Memory-augmented deep conditional unfolding network for pansharpening. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1778–1787.
Yokoya, N., Yairi, T., Iwasaki, A., 2012. Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. IEEE Trans. Geosci. Remote Sens. 50 (2), 528–537.
Yuhas, R.H., Goetz, A.F.H., Boardman, J.W., 1992. Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm. In: JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop.
Zhang, Y., Liu, P., Chen, L., Xu, M., Guo, X., Zhao, L., 2023. A new multi-source remote sensing image sample dataset with high resolution for flood area extraction: GF-FloodNet. Int. J. Digit. Earth 16 (1), 2522–2554.
Zhang, H., Ma, J., 2021. GTP-PNet: A residual learning network based on gradient transformation prior for pansharpening. ISPRS J. Photogramm. Remote Sens. 172, 223–239.
Zhang, K., Wang, M., Yang, S., Jiao, L., 2019. Convolution structure sparse coding for fusion of panchromatic and multispectral images. IEEE Trans. Geosci. Remote Sens. 1117–1130.
Zhang, F., Zhang, H., Zhang, K., Xing, Y., Sun, J., Wu, Q., 2021. Exploiting low-rank and sparse properties in strided convolution matrix for pansharpening. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 2649–2661.
Zhou, H., Hou, J., Zhang, Y., Ma, J., Ling, H., 2022a. Unified gradient-and intensity-discriminator generative adversarial network for image fusion. Inf. Fusion 88, 184–201.
Zhou, M., Huang, J., Yan, K., Yu, H., Fu, X., Liu, A., Wei, X., Zhao, F., 2022b. Spatial-frequency domain information integration for pan-sharpening. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (Eds.), European Conference on Computer Vision. Springer Nature Switzerland, Cham, pp. 274–291.
Zhu, Z., Cao, X., Zhou, M., Huang, J., Meng, D., 2023. Probability-based global cross-modal upsampling for pansharpening. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 14039–14048.