FusionMamba: Efficient Remote Sensing Image Fusion with State Space Model

Siran Peng, Xiangyu Zhu, *Senior Member, IEEE,* Haoyu Deng, Liang-Jian Deng, *Senior Member, IEEE,* and Zhen Lei, *Fellow, IEEE*

***Abstract*—Remote sensing image fusion aims to generate a high-resolution multi/hyper-spectral image by combining a high-resolution image with limited spectral data and a low-resolution image rich in spectral information. Current deep learning (DL) methods typically employ convolutional neural networks (CNNs) or Transformers for feature extraction and information integration. While CNNs are efficient, their limited receptive fields restrict their ability to capture global context. Transformers excel at learning global information but are computationally expensive. Recent advancements in the state space model (SSM), particularly Mamba, present a promising alternative by enabling global perception with low complexity. However, the potential of SSM for information integration remains largely unexplored. Therefore, we propose FusionMamba, an innovative method for efficient remote sensing image fusion. Our contributions are twofold. First, to effectively merge spatial and spectral features, we expand the single-input Mamba block to accommodate dual inputs, creating the FusionMamba block, which serves as a plug-and-play solution for information integration. Second, we incorporate Mamba and FusionMamba blocks into an interpretable network architecture tailored for remote sensing image fusion. Our designs utilize two U-shaped network branches, each primarily composed of four-directional Mamba blocks, to extract spatial and spectral features separately and hierarchically. The resulting feature maps are sufficiently merged in an auxiliary network branch constructed with FusionMamba blocks. Furthermore, we improve the representation of spectral information through an enhanced channel attention module. Quantitative and qualitative valuation results across six datasets demonstrate that our method achieves state-of-the-art (SOTA) performance, underscoring the effectiveness of FusionMamba. The code is available at https://github.com/PSRben/FusionMamba.**

***Index Terms*—Remote sensing image fusion, pansharpening, hyper-spectral pansharpening, deep learning (DL), convolutional neural networks (CNNs), Transformers, state space model (SSM).**

Siran Peng and Xiangyu Zhu are with the State Key Laboratory of Multimodel Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China; the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China. (Emails: pengsiran2023@ia.ac.cn, xiangyu.zhu@nlpr.ia.ac.cn)

Haoyu Deng is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan 611731, China. (Email: haoyu_deng@std.uestc.edu.cn)

Liang-Jian Deng is with the School of Mathematical Sciences/Multi-Hazard Early Warning Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan 611731, China. (Email: liangjian.deng@uestc.edu.cn)

Zhen Lei is with the State Key Laboratory of Multimodel Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China; the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China; the Centre for Artificial Intelligence and Robotics, Hong Kong Institute of Science and Innovation, Chinese Academy of Sciences, Hong Kong, China. (Email: zhen.lei@ia.ac.cn)

Corresponding author: Liang-Jian Deng.

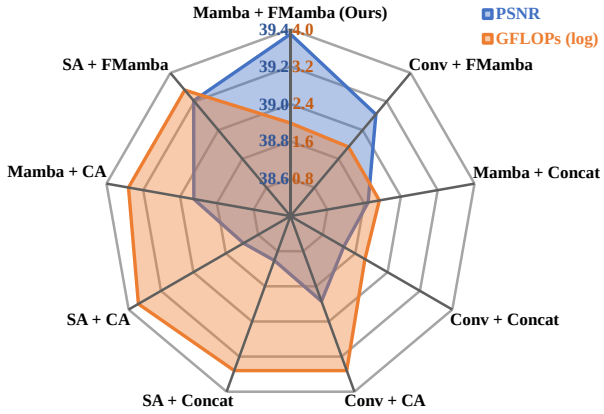


Fig. 1. Different combinations of feature extraction methods and information integration approaches for remote sensing image fusion. The candidate feature extraction methods include the convolution (Conv) layer, self-attention (SA) module [1], and four-directional Mamba (Mamba) block [2]. For information integration, the options comprise the concatenation (Concat) operation, cross-attention (CA) module, and the proposed FusionMamba (FMamba) block. For fairness, all combinations are designed with the same number of parameters. Quantitative evaluation results on 20 reduced-resolution samples from the WorldView-3 (WV3) dataset [3] demonstrate the superior efficacy and efficiency of our method. For precise values, please refer to Table VIII.

I. Introduction

Due to hardware limitations, satellite sensors often struggle to capture high-resolution multi/hyper-spectral images. As an alternative approach, they can simultaneously acquire a high-resolution image with limited spectral data and a low-resolution image with extensive spectral information. Remote sensing image fusion aims to merge these two types of images, generating a high-resolution result enriched with spectral characteristics. This study primarily investigates two remote sensing image fusion tasks: pansharpening [4] and hyper-spectral pansharpening [5]. Pansharpening involves creating a high-resolution multi-spectral (HRMS) image by combining a high-resolution panchromatic (PAN) image with a low-resolution multi-spectral (LRMS) image. Hyper-spectral pansharpening extends this process to hyper-spectral images, producing a high-resolution hyper-spectral (HRHS) image from a PAN image and a low-resolution hyper-spectral (LRHS) image.

Traditional (hyper-spectral) pansharpening studies can be broadly classified into three categories: component substitution (CS) methods, multi-resolution analysis (MRA) approaches, and variational optimization (VO) techniques. The CS-based methods [6]–[8] project the LRMS/LRHS image into a transformed domain, where the spatial information is treated as an independent component. By substituting this component with

the PAN image, a desired HRMS/HRHS result is produced. These methods are known for their simplicity, low computational requirements, and high spatial fidelity. However, they often lead to significant spectral distortions. The MRA-based approaches [9], [10] utilize an MRA framework to inject spatial details from the PAN image into the LRMS/LRHS image, thus generating an HRMS/HRHS output. These techniques are effective at preserving spectral characteristics but may experience issues with spatial distortions. The VO-based techniques [11]–[15] aim to uncover the intrinsic relationships between two types of images. They typically rely on various forms of prior information to construct optimization models that integrate spatial and spectral data. Despite their meticulous designs, VO-based techniques often fail to deliver satisfactory fusion results and are hindered by slow inference speeds.

Over the past few years, deep learning (DL) has become the leading solution for addressing image fusion problems within the remote sensing domain. By leveraging the powerful feature learning and non-linear fitting capabilities of neural networks, DL-based methods have consistently yielded impressive outcomes [16]–[34]. Analyzing these studies reveals two key insights. *First, networks with global perception abilities are particularly effective, as they leverage holistic information rather than solely relying on localized features. Second, since low-level tasks necessitate processing at relatively high resolutions, it is essential to keep computational complexity within manageable limits.* Most DL-based methods utilize convolutional neural networks (CNNs) [35] or Transformers [1] for feature extraction and information integration. Although CNNs are computationally efficient, they are hindered by limited receptive fields, restricting their ability to capture global context. On the contrary, Transformers excel at extracting global features but are burdened by quadratic complexity with respect to the length of input tokens. Recent breakthroughs in the state space model (SSM) [36]–[39], particularly Mamba [40], offer a promising solution to this issue by achieving global perception with linear complexity. The SSM has demonstrated remarkable success across a range of computer vision tasks [2], [41]–[44], delivering outstanding performance while requiring significantly fewer computational resources compared to Transformers. *However, there has been limited exploration into the potential of the SSM for integrating different types of information, which is a crucial aspect of image fusion.*

Given the aforementioned situation, we propose FusionMamba, a novel method for efficient remote sensing image fusion. Our innovations focus on two aspects. First, we expand the single-input Mamba block to support dual inputs, creating the FusionMamba block. This new module effectively merges spatial and spectral features, demonstrating superiority over existing fusion techniques like concatenation and cross-attention [1], as illustrated in Fig. 1. Moreover, experimental results presented in Table VI indicate that the FusionMamba block can function as a plug-and-play module for information integration. Second, based on the intrinsic properties of image fusion, we meticulously design an interpretable network architecture that incorporates Mamba and FusionMamba blocks. For feature extraction, we embed four-directional Mamba blocks [2] into two U-shaped network branches: the spatial branch and the spectral branch. The former emphasizes capturing spatial details from the PAN image, while the latter focuses on extracting spectral features from the LRMS/LRHS image. This design allows for the separate and hierarchical learning of spatial and spectral information. The resulting feature maps from both branches are sufficiently merged in an auxiliary combination branch, which is constructed using several FusionMamba blocks. To further improve the representation of spectral information, we introduce a Mamba-driven channel attention (MCA) module, where the traditional multi-layer perceptron (MLP) is replaced with a bidirectional Mamba block [41]. The contributions of this study are as follows:

1) To effectively merge spatial and spectral information, we expand the Mamba block to accommodate dual inputs, resulting in the innovative FusionMamba block. This module demonstrates superior effectiveness over existing fusion techniques, representing a significant advancement in the application of the SSM for information integration.
2) According to the properties of image fusion, we develop an interpretable network architecture that incorporates Mamba and FusionMamba blocks. Our designs employ two U-shaped network branches to extract spatial and spectral features separately and hierarchically. The resulting feature maps are sufficiently merged in a combination branch. Additionally, an enhanced channel attention module is utilized to improve spectral representation.
3) To the best of our knowledge, this study represents the first application of the SSM in hyper-spectral pansharpening and hyper-spectral image super-resolution (HISR) tasks. The proposed method achieves state-of-the-art (SOTA) performance across six datasets, thereby convincingly demonstrating the superiority of FusionMamba.

The rest of this paper is structured as follows. Section II reviews the related works and outlines our motivations. Section III provides a detailed explanation of our method. In Section IV, we present the experimental results for both pansharpening and hyper-spectral pansharpening tasks, accompanied by comprehensive ablation studies. Finally, Sections V and VI cover the discussion and conclusion, respectively.

II. Related Works and Motivations

A. DL Methods for Remote Sensing Image Fusion

In recent years, DL-based methods have dominated the remote sensing image fusion community. These techniques leverage the powerful feature learning and non-linear fitting capabilities inherent in neural networks, significantly outperforming traditional approaches. Broadly speaking, DL-based methods for remote sensing image fusion can be classified into two categories: CNN-based approaches and Transformer-based techniques. Notable examples in the first category include PNN [16], PanNet [17], and FusionNet [23]. PNN represents a pioneering advancement by integrating DL into the pansharpening field. It employs three stacked convolutional layers to achieve SOTA performance at the time of its publication. PanNet creatively uses high-pass filters to capture edge information and incorporates residual network (ResNet) blocks [45] to extract spatial and spectral features. FusionNet

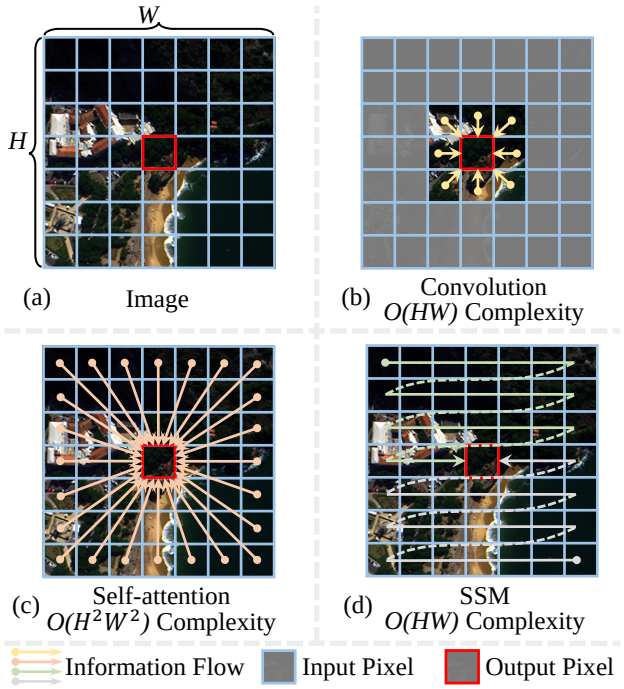


Fig. 2. Comparison among the convolution layer in CNNs, the self-attention module in Transformers, and the SSM in bidirectional Mamba [41]. (a) Suppose we have an image with a resolution of $H \times W$. (b) The convolution operation integrates pixels within a limited receptive field, resulting in a computational complexity of $O(HW)$. (c) The self-attention mechanism uniformly integrates all pixels, which leads to a significantly higher computational complexity of $O(H^2W^2)$. (d) The SSM integrates all pixels along specific directions, with those closer to the output pixel contributing more significantly to the final result. Additionally, its computational complexity is $O(HW)$.

embeds CNNs into the architecture of traditional algorithms, yielding remarkable outcomes. However, due to the limited receptive field size of convolutional kernels, these CNN-based approaches often struggle to capture global information, resulting in significant spatial distortions. Transformer-based techniques address this problem by calculating the correlation between any two pixels. Noteworthy works in this category include INNformer [25] and U2Net [30]. The former pioneers the application of Transformers for pansharpening, utilizing modified cross-attention blocks to sufficiently merge spatial and spectral feature maps. The latter incorporates enhanced cross-attention modules into a U-shaped network, attaining SOTA results in pansharpening. Despite their exceptional performance, these techniques are burdened by a high volume of floating-point operations (FLOPs) due to the quadratic complexity. Some methods in the remote sensing domain, such as STT [46] and LeMeViT [47], attempt to alleviate this computational burden by discarding redundant tokens. However, these approaches are not suitable for image fusion tasks and still exhibit quadratic complexity. Over the past two years, the development of DL-based methods for remote sensing image fusion has encountered a bottleneck, with none successfully achieving both global perception and low computational cost.

B. State Space Model

The SSM is a foundational scientific model primarily utilized in control theory and econometrics. Recently, its application has extended into the field of DL, thanks to the pioneering research of LSSL [36] and S4 [37]. Based on a series of mathematical derivations, LSSL approaches the SSM as a foundational DL framework. Building on this, S4 introduces the concept of *normal plus low-rank*, substantially reducing the computational complexity during the training phase of the SSM. Subsequent studies, such as S5 [39] and H3 [38], further explore the potential of the SSM in DL, effectively narrowing the performance gap between the SSM and Transformers. This line of research has culminated in the development of Mamba [40], which synthesizes pivotal findings from earlier works and proposes a selection mechanism for dynamic feature extraction. Mamba not only outperforms Transformers across various 1D tasks but also demands significantly fewer computational resources. The success of Mamba has captivated the computer vision community, leading to its widespread adoption in a variety of 2D vision tasks. Since Mamba was originally designed for 1D tasks with inherent directionality, applying it directly to 2D vision tasks, which typically lack such directionality, will result in incomplete global perception. To address this limitation, Vision Mamba [41] flattens spatial feature maps from both positive and negative directions, introducing a bidirectional Mamba approach that ensures complete global perception. VMamba [2] further improves upon this by proposing a four-directional Mamba technique, which enables the discovery of more spatial connections. In the remote sensing domain, notable contributions include RSCaMa [42], RSMamba [43], and Pan-Mamba [44]. RSCaMam introduces the SSM into remote sensing change captioning, achieving commendable performance by employing Mamba for joint spatial-temporal modeling. RSMamba proposes a shuffle flattening method to explore unconventional spatial connections, yielding SOTA results in remote sensing image classification tasks. Additionally, Pan-Mamba represents a pioneering effort in utilizing Mamba for pansharpening, demonstrating impressive performance even with the original Mamba blocks. However, the methods discussed above primarily concentrate on the application and directionality of the SSM, leaving its potential for information integration largely unexplored.

C. Motivations

Existing DL-based methods for remote sensing image fusion primarily utilize CNNs or Transformers for feature extraction and information integration. While CNNs are efficient, they often struggle to capture global information. Conversely, Transformers exhibit outstanding global perception but come with significant computational costs. Fortunately, recent advancements in the SSM, particularly Mamba, offer a promising solution to this dilemma, achieving both global perception and high efficiency. For a clearer understanding, Fig. 2 provides a visual comparison of the convolution layer in CNNs, the self-attention module in Transformers, and the SSM in bidirectional Mamba. It is evident that the SSM integrates the strengths of both CNNs and Transformers. Although the SSM has demonstrated notable success in a range of computer vision tasks, its potential for information integration, an essential aspect of image fusion, remains largely untapped. Therefore,

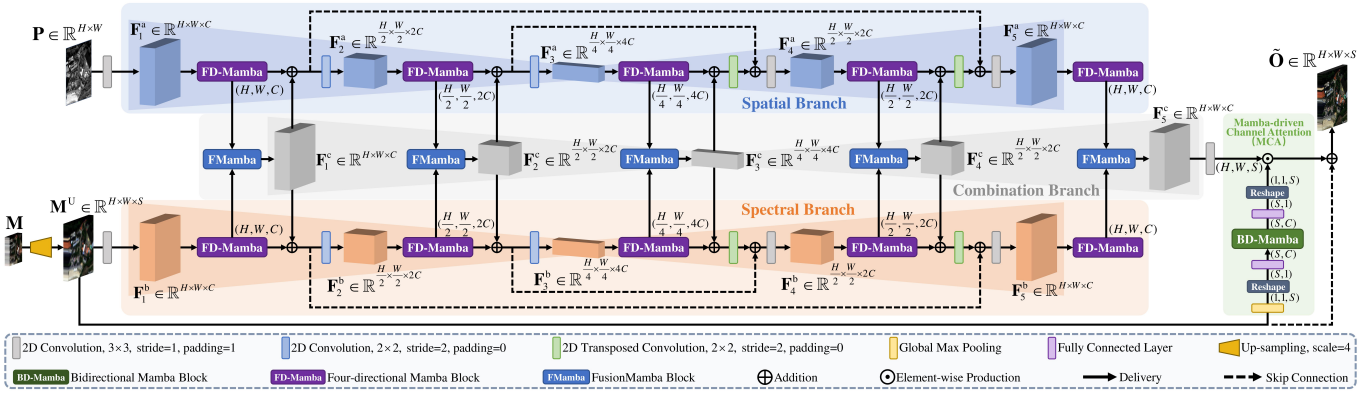


Fig. 3. The proposed network architecture. Our designs comprise two U-shaped network branches dedicated to feature extraction, a combination branch for information integration, and an MCA module for spectral enhancement. Detailed structures of the Mamba and FusionMamba blocks are depicted in Fig. 4.

we expand the single-input Mamba block to support dual inputs, resulting in the FusionMamba block, which effectively merges spatial and spectral information. Next, we need to determine an interpretable network architecture that leverages Mamba and FusionMamba blocks for feature extraction and information integration, respectively. Given that images from different sources exhibit distinct characteristics, we employ two U-shaped network branches, each primarily composed of Mamba blocks, to separately and hierarchically learn spatial and spectral features. Additionally, we utilize an auxiliary network branch built with FusionMamba blocks to achieve sufficient information integration. Moreover, to mitigate the distortion caused by encoding spectral data into the channels of feature maps, we develop an enhanced channel attention module to improve the representation of spectral information.

III. Methodology

In this section, we first introduce the notations (Section III-A) and explore the mathematical foundations of the SSM (Section III-B). Next, we provide a detailed explanation of the network architecture (Section III-C), followed by an in-depth discussion of the Mamba and FusionMamba blocks (Section III-D). Finally, we describe the loss function (Section III-E).

A. Notations

The PAN image is denoted as $\mathbf{P} \in \mathbb{R}^{H \times W}$, where H and W represent its height and width. In addition, $\mathbf{M} \in \mathbb{R}^{h \times w \times S}$ denotes the LRMS/LRHS image, with S representing the number of spectral bands, $h = \frac{H}{4}$, and $w = \frac{W}{4}$. Furthermore, the up-sampled LRMS/LRHS image, the generated HRMS/HRHS image, and the ground-truth (GT) image are defined as $\mathbf{M}^{\mathrm{U}} \in \mathbb{R}^{H \times W \times S}$, $\tilde{\mathbf{O}} \in \mathbb{R}^{H \times W \times S}$, and $\mathbf{O} \in \mathbb{R}^{H \times W \times S}$, respectively. Our network takes $\mathbf{P}$ and $\mathbf{M}$ as inputs to produce an output $\tilde{\mathbf{O}}$, which is supervised by the GT image $\mathbf{O}$. The network performs feature extraction and information integration through five cascading stages. At the i-th stage, the spatial, spectral, and fusion feature maps are denoted as $\mathbf{F}_i^{\mathrm{a}}$, $\mathbf{F}_i^{\mathrm{b}}$, and $\mathbf{F}_i^{\mathrm{c}}$, respectively. The dimensions of $\mathbf{F}_{\{1,5\}}^{\{\mathrm{a,b,c}\}}$, $\mathbf{F}_{\{2,4\}}^{\{\mathrm{a,b,c}\}}$, and $\mathbf{F}_3^{\{\mathrm{a,b,c}\}}$ are $H \times W \times C$, $\frac{H}{2} \times \frac{W}{2} \times 2C$, and $\frac{H}{4} \times \frac{H}{4} \times 4C$, where C represents the number of channels. Additionally, N denotes the size of hidden states in the SSM.

B. Preliminaries

1) State Space Model: The SSM is a continuous system that maps a 1D input $x(t) \in \mathbb{R}$ into an output $y(t) \in \mathbb{R}$ via intermediate hidden states $h(t) \in \mathbb{R}^N$. This process is frequently described using ordinary differential equations (ODEs), as illustrated below:

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t), \\ y(t) &= \mathbf{C}h(t). \end{aligned} \tag{1}$$

Here, $\mathbf{A} \in \mathbb{R}^{N \times N}$ denotes the state matrix governing the system's evolution. $\mathbf{B} \in \mathbb{R}^{N \times 1}$ and $\mathbf{C} \in \mathbb{R}^{1 \times N}$ are projection parameters that regulate the system updates. Eq. 1 indicates that the SSM possesses global perception, as its current output is influenced by all preceding inputs. When $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are constant, this equation characterizes a linear time-invariant (LTI) system, as exemplified in LSSL [36] and S4 [37]. Conversely, when these parameters change over time, the equation describes a linear time-varying (LTV) system, which is the case in Mamba [40]. LTI systems inherently lack the ability to perceive input content, whereas input-aware LTV systems are designed to possess this capability.

2) Discretization: When employing the SSM in the field of DL, discretization is required. To facilitate this process, a timescale parameter, denoted as $\mathbf{\Delta} \in \mathbb{R}$, is introduced to convert the continuous parameters $\mathbf{A}$ and $\mathbf{B}$ into their discrete counterparts, represented as $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$. Using the zero-order hold (ZOH) method as the transformation algorithm, the discrete parameters are calculated as follows:

$$\begin{aligned} \overline{\mathbf{A}} &= \exp(\mathbf{\Delta A}), \\ \overline{\mathbf{B}} &= (\mathbf{\Delta A})^{-1}(\exp(\mathbf{\Delta A}) - \mathbf{I}) \cdot \mathbf{\Delta B} \approx \mathbf{\Delta B}. \end{aligned} \tag{2}$$

Then, the discrete form of Eq. 1 can be expressed as:

$$\begin{aligned} h_t &= \overline{\mathbf{A}}h_{t-1} + \overline{\mathbf{B}}x_t, \\ y_t &= \mathbf{C}h_t. \end{aligned} \tag{3}$$

In practice, x_t is a feature vector with C components, and Eq. 3 processes each of these components independently.

3) Selective Scan: In Mamba, the variability of parameters with the input prevents the reformulation of Eq. 3 into a convolutional form, thereby impeding the parallelization of the SSM. To overcome this obstacle, Mamba introduces the

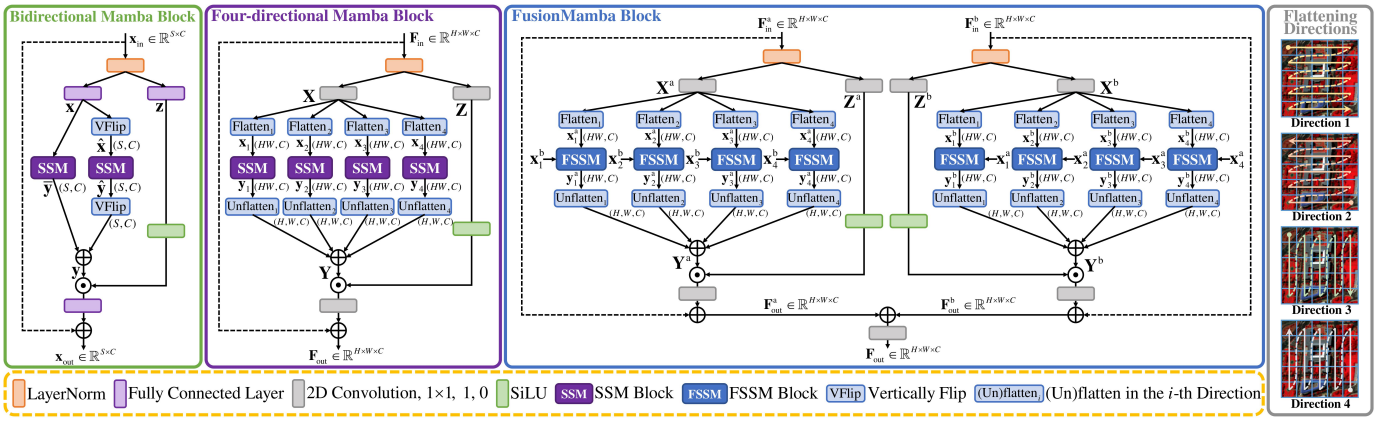


Fig. 4. The schematic diagram of the bidirectional Mamba block (first from the left), the four-directional Mamba block (second from the left), and the proposed FusionMamba block (second from the right), along with an illustration depicting the four flattening directions (first from the right). FSSM stands for the fusion state space model. Additionally, the specifics of the SSM and FSSM blocks are detailed in Algorithms 1 and 2, respectively.

Algorithm 1 SSM Block

Input: $\mathbf{x} : (HW, C)$
Output: $\mathbf{y} : (HW, C)$
$\mathbf{A} : (C, N) \leftarrow \mathbf{Parameter}_{\mathbf{A}}$
/* $\mathbf{A}$ represents C sets of structured $N \times N$ matrices [37] */
$\mathbf{B} : (HW, N) \leftarrow \mathbf{Linear}_{\mathbf{B}}(\mathbf{x})$
$\mathbf{C} : (HW, N) \leftarrow \mathbf{Linear}_{\mathbf{C}}(\mathbf{x})$
$\mathbf{\Delta} : (HW, C) \leftarrow \log(1 + \exp(\mathbf{Linear}_{\Delta}(\mathbf{x}) + \mathbf{Parameter}_{\Delta}))$
/* $\mathbf{Parameter}_{\Delta}$ is a bias vector with a size of C */
$\overline{\mathbf{A}} : (HW, C, N) \leftarrow \exp(\mathbf{\Delta} \otimes \mathbf{A})$
$\overline{\mathbf{B}} : (HW, C, N) \leftarrow \mathbf{\Delta} \otimes \mathbf{B}$
$\mathbf{y} \leftarrow \mathrm{SSM}(\overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{C})(\mathbf{x})$
/* SSM represents Eq. 3 implemented using selective scan */
return $\mathbf{y}$

Algorithm 2 FSSM Block

Inputs: $\mathbf{x}^{\mathrm{a}}, \mathbf{x}^{\mathrm{b}} : (HW, C)$
Output: $\mathbf{y}^{\mathrm{a}} : (HW, C)$
$\mathbf{A} : (C, N) \leftarrow \mathbf{Parameter}_{\mathbf{A}}$
/* $\mathbf{A}$ represents C sets of structured $N \times N$ matrices [37] */
$\mathbf{B} : (HW, N) \leftarrow \mathbf{Linear}_{\mathbf{B}}(\mathbf{x}^{\mathrm{b}})$
$\mathbf{C} : (HW, N) \leftarrow \mathbf{Linear}_{\mathbf{C}}(\mathbf{x}^{\mathrm{b}})$
$\mathbf{\Delta} : (HW, C) \leftarrow \log(1 + \exp(\mathbf{Linear}_{\Delta}(\mathbf{x}^{\mathrm{b}}) + \mathbf{Parameter}_{\Delta}))$
/* $\mathbf{Parameter}_{\Delta}$ is a bias vector with a size of C */
$\overline{\mathbf{A}} : (HW, C, N) \leftarrow \exp(\mathbf{\Delta} \otimes \mathbf{A})$
$\overline{\mathbf{B}} : (HW, C, N) \leftarrow \mathbf{\Delta} \otimes \mathbf{B}$
$\mathbf{y}^{\mathrm{a}} \leftarrow \mathrm{SSM}(\overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{C})(\mathbf{x}^{\mathrm{a}})$
/* SSM represents Eq. 3 implemented using selective scan */
return $\mathbf{y}^{\mathrm{a}}$

selective scan mechanism, which incorporates three hardware-related techniques: kernel fusion, parallel scan, and recomputation. This selective scan enables Mamba to achieve impressive speed while maintaining a relatively low memory requirement.

C. Network Architecture

To fully exploit the potential of the SSM in remote sensing image fusion, we design an interpretable network architecture, which consists of two U-shaped network branches (the spatial branch and the spectral branch) for feature extraction, a combination branch for information integration, and an MCA module for spectral enhancement, as shown in Fig. 3. Next, we will provide a detailed explanation of these network components.

1) U-shaped Network Branches: This design facilitates the efficient learning of spatial and spectral information in a separate and hierarchical manner. Specifically, the spatial branch focuses on extracting spatial details from $\mathbf{P}$, while the spectral branch is dedicated to capturing spectral characteristics from $\mathbf{M}$. To acquire sufficient deep-level information without significantly increasing network parameters, we extract features at three different scales. This means that each U-shaped network branch comprises a total of five stages. At each stage, the spatial or spectral feature map is initially processed by a four-directional Mamba block. The resulting feature map then passes through a FusionMamba block to generate a fusion output, which is subsequently added back to the original input. Finally, a convolution layer of varying types is employed to adjust both the spatial resolution and the number of channels.

2) Combination Branch: This design enables the comprehensive integration of spatial and spectral information. To align with the U-shaped network branches, it incorporates five FusionMamba blocks. Each block receives its corresponding spatial and spectral feature maps as inputs, generating a fusion output that is subsequently added back to the original inputs. From a holistic perspective, the combination branch effectively simulates the progressive merging of different features.

3) Mamba-driven Channel Attention: The MCA module is designed to improve the representation of spectral information. Based on the widely used channel attention mechanism [48], the MCA module replaces the MLP with a bidirectional Mamba block. Additionally, several modifications are made to better accommodate the characteristics of the SSM in data processing. Specifically, we first utilize global max pooling to eliminate spatial information from $\mathbf{M}^{\mathrm{U}}$, resulting in a $1 \times 1 \times S$ feature map. This map is then reshaped into a 1D sequence of size $S \times 1$. Next, we employ a fully connected layer to increase the channel number of this sequence to C. The augmented 1D sequence is subsequently passed through a bidirectional Mamba block to extract spectral features. Finally, the output is projected and reshaped back into a $1 \times 1 \times S$ feature map, which is multiplied with $\mathbf{F}_5^{\mathrm{c}}$ to complete the spectral enhancement.

D. Mamba and FusionMamba Blocks

In this section, we detail the bidirectional Mamba block, the four-directional Mamba block, and the proposed FusionMamba block, all of which are depicted in Fig. 4. Additionally, we compare the FLOPs required by different DL models.

1) Bidirectional Mamba Block: For an input 1D sequence $\mathbf{x}_{\text{in}} \in \mathbb{R}^{S\times C}$, we first normalize it using layer normalization. Next, it is processed by two parallel fully connected layers, producing two distinct sequences, denoted as $\mathbf{x} \in \mathbb{R}^{S\times C}$ and $\mathbf{z} \in \mathbb{R}^{S\times C}$. This procedure can be expressed as follows:

$$\mathbf{x}, \mathbf{z} = \mathbf{Linear}_{\text{x}}(\mathbf{Norm}(\mathbf{x}_{\text{in}})), \mathbf{Linear}_{\text{z}}(\mathbf{Norm}(\mathbf{x}_{\text{in}})). \quad (4)$$

Here, $\mathbf{Norm}$ denotes layer normalization, while $\mathbf{Linear}_{\text{x}}$ and $\mathbf{Linear}_{\text{z}}$ represent two separate fully connected layers. After that, we flip $\mathbf{x}$ vertically, generating $\hat{\mathbf{x}}$ of size $S \times C$. Subsequently, $\mathbf{x}$ and $\hat{\mathbf{x}}$ are processed separately through two SSM blocks for feature extraction, resulting in two output sequences, denoted as $\overline{\mathbf{y}}$ and $\hat{\mathbf{y}}$. This process is expressed as:

$$\begin{aligned} \hat{\mathbf{x}} &= \text{VFlip}(\mathbf{x}), \\ \overline{\mathbf{y}}, \hat{\mathbf{y}} &= \mathbf{SSM}_1(\mathbf{x}), \mathbf{SSM}_2(\hat{\mathbf{x}}). \end{aligned} \quad (5)$$

Here, VFlip refers to the operation of flipping a matrix vertically. $\mathbf{SSM}_1$ and $\mathbf{SSM}_2$ represent two separate SSM blocks, which are thoroughly detailed in Algorithm 1. Next, we vertically flip $\hat{\mathbf{y}}$ and add it to $\overline{\mathbf{y}}$, producing a new sequence denoted as $\mathbf{y} \in \mathbb{R}^{S\times C}$. After gating by $\mathbf{z}$, this sequence undergoes a fully connected layer and is added to $\mathbf{x}_{\text{in}}$, resulting in the final output represented as $\mathbf{x}_{\text{out}} \in \mathbb{R}^{S\times C}$:

$$\begin{aligned} \mathbf{y} &= \overline{\mathbf{y}} + \text{VFlip}(\hat{\mathbf{y}}), \\ \mathbf{x}_{\text{out}} &= \mathbf{Linear}_{\text{o}}(\mathbf{y} \cdot \text{SiLU}(\mathbf{z})) + \mathbf{x}_{\text{in}}. \end{aligned} \quad (6)$$

Here, $\mathbf{Linear}_{\text{o}}$ denotes the fully connected layer, and SiLU stands for the "SiLU" activation function.

2) Four-directional Mamba Block: Given an input feature map $\mathbf{F}_{\text{in}} \in \mathbb{R}^{H\times W\times C}$, we normalize it using layer normalization and process it via two parallel 1×1 convolution layers, yielding two distinct feature maps, denoted as $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ and $\mathbf{Z} \in \mathbb{R}^{H\times W\times C}$. This process can be expressed as follows:

$$\mathbf{X}, \mathbf{Z} = \mathbf{Conv}_{\text{x}}(\mathbf{Norm}(\mathbf{F}_{\text{in}})), \mathbf{Conv}_{\text{z}}(\mathbf{Norm}(\mathbf{F}_{\text{in}})). \quad (7)$$

Here, $\mathbf{Conv}_{\text{x}}$ and $\mathbf{Conv}_{\text{z}}$ represents two separate convolution layers. Following this, $\mathbf{X}$ is flattened in four directions, producing $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$, each with dimension of $HW \times C$. These sequences are then separately processed by SSM blocks, resulting in four outputs denoted as $\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{y}_3$, and $\mathbf{y}_4$:

$$\begin{cases} \mathbf{x}_i = \text{Flatten}_i(\mathbf{X}), \\ \mathbf{y}_i = \mathbf{SSM}_i(\mathbf{x}_i). \end{cases} \quad i = 1, 2, 3, 4. \quad (8)$$

Here, Flatten_i represents the flattening operation along the i-th direction. Subsequently, we unflatten the outputs of SSM blocks and combine them to obtain a new feature map, denoted as $\mathbf{Y} \in \mathbb{R}^{H\times W\times C}$. After gating by $\mathbf{Z}$, this feature map undergoes a 1×1 convolution layer and is added to $\mathbf{F}_{\text{in}}$, yielding the final output represented as $\mathbf{F}_{\text{out}} \in \mathbb{R}^{H\times W\times C}$:

$$\begin{aligned} \mathbf{Y} &= \sum_{i=1}^{4} \text{Unflatten}_i(\mathbf{y}_i), \\ \mathbf{F}_{\text{out}} &= \mathbf{Conv}_{\text{o}}(\mathbf{Y} \cdot \text{SiLU}(\mathbf{Z})) + \mathbf{F}_{\text{in}}. \end{aligned} \quad (9)$$

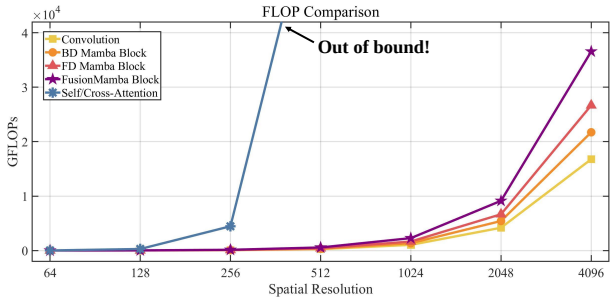


Fig. 5. Comparison of FLOPs among the convolution layer, bidirectional (BD) Mamba block, four-directional (FD) Mamba block, FusionMamba block, and self/cross-attention module at various spatial resolutions. For optimal visual effects, we configure D, C, and N to be 0.5M, 256, and 64, respectively.

Here, Unflatten_i denotes the operation of unflattening along the i-th direction and $\mathbf{Conv}_{\text{o}}$ represents the convolution layer.

3) FusionMamba Block: The original SSM can only handle a single input. To effectively integrate different types of information, we expand it to accommodate dual inputs, resulting in the fusion state space model (FSSM), as detailed in Algorithm 2. Within the FSSM block, one input is responsible for generating the projection and timescale parameters, while the other input is the sequence to be processed. The FusionMamba block, consisting of eight FSSM blocks, is designed with a symmetrical structure. For the input spatial and spectral feature maps, denoted as $\mathbf{F}^{\text{a}}_{\text{in}} \in \mathbb{R}^{H\times W\times C}$ and $\mathbf{F}^{\text{b}}_{\text{in}} \in \mathbb{R}^{H\times W\times C}$, we employ a method similar to the four-directional Mamba block to generate two sets of feature maps as follows:

$$\begin{aligned} \mathbf{X}^{\text{a}}, \mathbf{Z}^{\text{a}} &= \mathbf{Conv}^{\text{a}}_{\text{x}}(\mathbf{Norm}(\mathbf{F}^{\text{a}}_{\text{in}})), \mathbf{Conv}^{\text{a}}_{\text{z}}(\mathbf{Norm}(\mathbf{F}^{\text{a}}_{\text{in}})); \\ \mathbf{X}^{\text{b}}, \mathbf{Z}^{\text{b}} &= \mathbf{Conv}^{\text{b}}_{\text{x}}(\mathbf{Norm}(\mathbf{F}^{\text{b}}_{\text{in}})), \mathbf{Conv}^{\text{b}}_{\text{z}}(\mathbf{Norm}(\mathbf{F}^{\text{b}}_{\text{in}})). \end{aligned} \quad (10)$$

Since this equation is a direct extension of Eq. 7, explanations for the symbols are omitted. Next, $\mathbf{X}^{\text{a}}$ and $\mathbf{X}^{\text{b}}$ are flattened separately in four directions. The resulting 1D sequences are then forwarded to FSSM blocks for information integration:

$$\begin{cases} \mathbf{x}^{\text{a}}_i, \mathbf{x}^{\text{b}}_i = \text{Flatten}_i(\mathbf{X}^{\text{a}}), \text{Flatten}_i(\mathbf{X}^{\text{b}}), \\ \mathbf{y}^{\text{a}}_i, \mathbf{y}^{\text{b}}_i = \mathbf{FSSM}^{\text{a}}_i(\mathbf{x}^{\text{a}}_i, \mathbf{x}^{\text{b}}_i), \mathbf{FSSM}^{\text{b}}_i(\mathbf{x}^{\text{b}}_i, \mathbf{x}^{\text{a}}_i). \end{cases} \quad i = 1, 2, 3, 4. \quad (11)$$

Here, $\mathbf{FSSM}^{\text{a}}$ and $\mathbf{FSSM}^{\text{b}}$ refer to the FSSM blocks on the left and right halves of the FusionMamba block in Fig. 4. After that, we process the two sets of outputs separately, producing two new feature maps denoted as $\mathbf{Y}^{\text{a}} \in \mathbb{R}^{H\times W\times C}$ and $\mathbf{Y}^{\text{b}} \in \mathbb{R}^{H\times W\times C}$. These maps are finally combined to form $\mathbf{F}_{\text{out}}$:

$$\begin{aligned} \mathbf{Y}^{\text{a}}, \mathbf{Y}^{\text{b}} &= \sum_{i=1}^{4} \text{Unflatten}_i(\mathbf{y}^{\text{a}}_i), \sum_{i=1}^{4} \text{Unflatten}_i(\mathbf{y}^{\text{b}}_i), \\ \mathbf{F}^{\text{a}}_{\text{out}} &= \mathbf{Conv}^{\text{a}}_{\text{o}}(\mathbf{Y}^{\text{a}} \cdot \text{SiLU}(\mathbf{Z}^{\text{a}})) + \mathbf{F}^{\text{a}}_{\text{in}}, \\ \mathbf{F}^{\text{b}}_{\text{out}} &= \mathbf{Conv}^{\text{b}}_{\text{o}}(\mathbf{Y}^{\text{b}} \cdot \text{SiLU}(\mathbf{Z}^{\text{b}})) + \mathbf{F}^{\text{b}}_{\text{in}}, \\ \mathbf{F}_{\text{out}} &= \mathbf{Conv}_{\text{o}}(\mathbf{F}^{\text{a}}_{\text{out}} + \mathbf{F}^{\text{b}}_{\text{out}}). \end{aligned} \quad (12)$$

Here, $\mathbf{Conv}^{\text{a}}_{\text{o}}$, $\mathbf{Conv}^{\text{b}}_{\text{o}}$, and $\mathbf{Conv}_{\text{o}}$ represent 1×1 convolution layers that generate $\mathbf{F}^{\text{a}}_{\text{out}}$, $\mathbf{F}^{\text{b}}_{\text{out}}$, and $\mathbf{F}_{\text{out}}$, respectively.

4) Analysis of FLOPs: In a convolution layer with D parameters, the FLOP count is commonly calculated as $2HWD$. Given that a selective scan costs $9HWCN$ FLOPs [40], the

TABLE I
QUANTITATIVE EVALUATION RESULTS ON 20 REDUCED-RESOLUTION AND 20 FULL-RESOLUTION SAMPLES FROM THE WV3 DATASET, WHICH BELONGS TO THE PANSHARPENING TASK. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**, AND THE SECOND-BEST RESULTS ARE UNDERLINED. ADDITIONALLY, THE METHODS ABOVE THE DIVIDING LINE REPRESENT TRADITIONAL APPROACHES, WHILE THE METHODS BELOW IT ARE DL-BASED TECHNIQUES.

Methods	Params	Reduced-Resolution				Full-Resolution (Real Data)		
		PSNR(±std)	Q2n(±std)	SAM(±std)	ERGAS(±std)	D_λ(±std)	D_s(±std)	QNR(±std)
TV [11]	–	32.381±2.328	0.795±0.120	5.692±1.808	4.855±1.434	0.0234±0.0061	0.0393±0.0227	0.9383±0.0269
GLP-HPM [9]	–	33.095±2.800	0.835±0.092	5.333±1.761	4.616±1.503	0.0206±0.0082	0.0630±0.0284	0.9180±0.0346
GLP-FS [10]	–	32.963±2.753	0.833±0.092	5.315±1.765	4.700±1.597	0.0197±0.0078	0.0630±0.0289	0.9187±0.0347
BDSD-PC [8]	–	32.970±2.784	0.829±0.097	5.428±1.822	4.697±1.617	0.0625±0.0235	0.0730±0.0356	0.8698±0.0531
PanNet [17]	0.08M	37.346±2.688	0.891±0.093	3.613±0.766	2.664±0.688	**0.0165**±0.0074	0.0470±0.0210	0.9374±0.0271
MSDCNN [49]	0.23M	37.068±2.686	0.890±0.090	3.777±0.803	2.760±0.689	0.0230±0.0091	0.0467±0.0199	0.9316±0.0271
BDPN [18]	1.49M	36.191±2.702	0.871±0.100	4.201±0.857	3.046±0.732	0.0364±0.0142	0.0459±0.0192	0.9196±0.0308
FusionNet [23]	0.08M	38.047±2.589	0.904±0.090	3.324±0.698	2.465±0.644	0.0239±0.0090	0.0364±0.0137	0.9406±0.0197
MUCNN [50]	2.32M	38.262±2.703	0.911±0.089	3.206±0.681	2.400±0.617	0.0258±0.0111	0.0327±0.0140	0.9424±0.0205
LAGNet [51]	0.15M	38.592±2.778	0.910±0.091	3.103±0.558	2.292±0.607	0.0368±0.0148	0.0418±0.0152	0.9230±0.0247
PMACNet [52]	0.94M	38.595±2.882	0.912±0.092	3.073±0.623	2.293±0.532	0.0540±0.0232	0.0336±0.0115	0.9143±0.0281
U2Net [30]	0.66M	39.117±3.009	0.920±0.085	2.888±0.581	2.149±0.525	0.0178±0.0072	0.0313±0.0075	0.9514±0.0115
Pan-Mamba [44]	0.48M	39.012±2.986	0.920±0.085	2.914±0.592	2.184±0.521	0.0183±0.0071	0.0307±0.0108	0.9516±0.0146
CANNet [34]	0.78M	39.003±2.900	0.919±0.084	2.941±0.590	2.175±0.530	0.0196±0.0083	0.0301±0.0074	0.9510±0.0126
FusionMamba	0.73M	**39.374**±2.973	**0.922**±0.084	**2.843**±0.577	**2.092**±0.510	0.0186±0.0078	**0.0269**±0.0058	**0.9550**±0.0110
Ideal Values	–	**+∞**	**1**	**0**	**0**	**0**	**0**	**1**

overall FLOP counts for a bidirectional Mamba block, a four-directional block, and a FusionMamba block, each with D parameters, are $2HWD+18HWCN$, $2HWD+36HWCN$, and $2HWD+72HWCN$, respectively. As for a self/cross-attention block in Transformers, the total FLOP count is estimated to be around $2HWD+4H^2W^2C$. The FLOP comparison among these modules, as depicted in Fig. 5, indicates that the Mamba and FusionMamba blocks possess FLOP costs comparable to that of the convolution layer and are significantly more efficient than the self/cross-attention block.

E. Loss Function

The main contributions of this study lie in the application and innovation of the SSM. Therefore, we employ the simplest ℓ_1 loss function for network training, as shown below:

$$\mathcal{L}oss = \frac{1}{T}\sum_{i=1}^{T} \|f_{\mathbf{\Theta}}(\mathbf{P}_i, \mathbf{M}_i) - \mathbf{O}_i\|_1. \tag{13}$$

Here, T denotes the total number of training examples, and $f_{\mathbf{\Theta}}$ represents our network with learnable parameters $\mathbf{\Theta}$. Additionally, $\mathbf{P}_i$, $\mathbf{M}_i$, and $\mathbf{O}_i$ refer to the i-th PAN image, LRMS/LRHS image, and GT image in the training dataset, respectively. Furthermore, $\|\cdot\|_1$ defines the ℓ_1 normalization.

IV. EXPERIMENTS

In this section, we present the quantitative and qualitative evaluation results for representative remote sensing image fusion approaches on the pansharpening and hyper-spectral pansharpening tasks. Additionally, we conduct comprehensive ablation studies to demonstrate the superiority of our method.

A. Pansharpening

1) Datasets: For the pansharpening task [4], we conduct experiments using the widely recognized WorldView-3 (WV3) and GaoFen-2 (GF2) datasets. The WV3 dataset consists of instances acquired by the sensor aboard the WV3 satellite. This sensor captures data across eight spectral bands, covering wavelengths from 0.4 to 1 μm, with a spatial resolution of 1.2 meters. The images in the GF2 dataset are collected by the sensor onboard the GF2 satellite, which records data across four spectral bands within the wavelength range of 0.4 to 0.9 μm. Additionally, this sensor provides a spatial resolution of 4 meters. Both datasets utilized in this study are sourced from the PanCollection[1]. The data generation process adheres strictly to Wald's protocol [53], with comprehensive details provided in [3]. Specifically, the WV3 dataset includes 10000 training samples, with 90% allocated for training and 10% for validation. Additionally, it contains 20 reduced-resolution and 20 full-resolution testing samples. Each training sample comprises an image triplet in the PAN/LRMS/GT format, with dimensions of 64×64, $16\times16\times8$, and $64\times64\times8$, respectively. The reduced-resolution testing samples include PAN/LRMS/GT image triplets sized 256×256, $64\times64\times8$, and $256\times256\times8$, respectively. Additionally, the full-resolution testing samples consist of image pairs in the PAN/LRMS format, with sizes of 512×512 and $128\times128\times8$, respectively. In the GF2 dataset, there are 22010 training samples, divided into 90% for training and 10% for validation. Additionally, this dataset includes 20 reduced-resolution and 20 full-resolution testing samples. Each training sample contains a PAN/LRMS/GT image triplet of sizes 64×64, $16\times16\times4$, and $64\times64\times4$, respectively. The reduced-resolution testing samples comprise PAN/LRMS/GT image triplets sized 256×256, $64\times64\times4$, and $256\times256\times4$, respectively. Additionally, the full-resolution testing samples consist of PAN/LRMS image pairs sized 512×512 and $128\times128\times4$. The primary distinction between the WV3 and GF2 datasets lies in the number of spectral bands included in their multi-spectral images.

[1] https://github.com/liangjiandeng/PanCollection

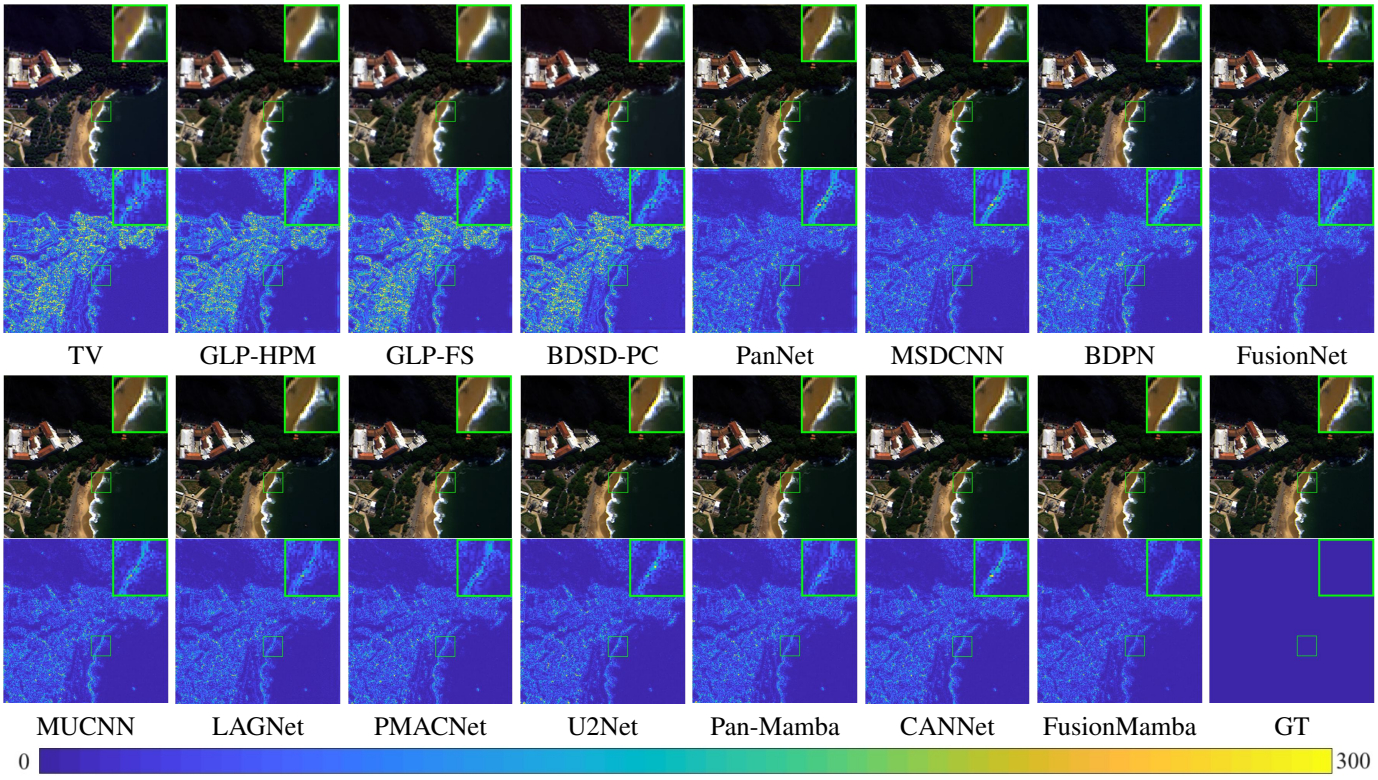


Fig. 6. Qualitative results on a reduced-resolution example from the WV3 dataset. Rows 1 and 3: Pseudo-color images representing spectral bands 1, 3, and 5. Rows 2 and 4: The corresponding absolute error maps (AEMs) for spectral band 7. The values on both sides of the color bar indicate the degree of errors.

2) Benchmarks: We compare FusionMamba with representative pansharpening techniques, including four traditional approaches: TV [11], GLP-HPM [9], GLP-FS [10], and BDSD-PC [8]; and ten DL-based methods: PanNet [17], MSDCNN [49], BDPN [18], FusionNet [23], MUCNN [50], LAGNet [51], PMACNet [52], U2Net [30], Pan-Mamba [44], and CANNet [34]. For fairness, all DL-based methods are trained using the same Nvidia GPU 3090 and PyTorch environment.

3) Quality Indices: In accordance with the research standards of pansharpening, we utilize four quality indices, namely PSNR [54], Q2n [55], SAM [56], and ERGAS [53], to evaluate the results on reduced-resolution samples. The ideal values for these indices are $+\infty$, 1, 0, and 0, respectively. For full-resolution samples, we employ D_λ, D_s, and QNR [57] as evaluation metrics, with ideal values of 0, 0, and 1, respectively. Notably, QNR, which combines D_λ and D_s, provides a comprehensive measure of overall fusion quality.

4) Settings: In the pansharpening task, we set C to 32 and N to 8. Additionally, we utilize the PixelShuffle technique [58] for up-sampling. During the training of our networks on the WV3 and GF2 datasets, the number of epochs is configured as 420 and 300, respectively. Besides, the batch size and initial learning rate are uniformly configured as 32 and 5×10^{-4}, respectively. Furthermore, we employ the Adam optimizer, with the learning rate halved every 200 epochs. As for other DL-based methods, we adhere to the default settings specified in the related papers or source codes.

5) Results: The quantitative evaluation results for the WV3 and GF2 datasets, respectively presented in Tables I and II, indicate that FusionMamba achieves the best overall performance on both the reduced-resolution and full-resolution testing samples. Given that the indicator values are approaching their limits, our method demonstrates significant improvements over other techniques. Additionally, the qualitative evaluation results on both datasets, as depicted in Figs. 6 and 7, reveal that the FusionMamba's absolute error maps (AEMs) are the closest to the GT images. Consequently, our method exhibits superior effectiveness in the pansharpening task.

B. Hyper-spectral Pansharpening

1) Datasets: We conduct experiments on three widely used hyper-spectral pansharpening datasets [5]: Pavia, Botswana, and Washington D.C. (WDC). The Pavia dataset includes images acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor, which records data across 115 spectral bands within the wavelength range of 0.4 to 0.9 μm. Additionally, this sensor offers a spatial resolution of 1.3 meters. The images in the Botswana dataset are collected by the Hyperion sensor aboard the Earth Observing-1 (EO-1) satellite, operated by the National Aeronautics and Space Administration (NASA). This sensor captures data across 242 spectral bands, spanning wavelengths from 0.4 to 2.5 μm, with a spatial resolution of 30 meters. The WDC dataset comprises images captured by the Hyper-spectral Digital Imagery Collection Experiment (HY-DICE) sensor, which records data across 210 spectral bands, covering a wavelength range from 0.4 to 2.4 μm, with a spatial resolution of 1 meter. The hyper-spectral pansharpening datasets used in this study are sourced from

TABLE II
RESULTS ON 20 REDUCED-RESOLUTION AND 20 FULL-RESOLUTION SAMPLES FROM THE GF2 DATASET, WHICH BELONGS TO PANSHARPENING.

Methods	Params	Reduced-Resolution				Full-Resolution (Real Data)		
		PSNR(±std)	Q2n(±std)	SAM(±std)	ERGAS(±std)	D_λ(±std)	D_s(±std)	QNR(±std)
TV [11]	–	41.262±2.264	0.907±0.029	1.911±0.447	1.737±0.447	0.0553±0.0430	0.1118±0.0226	0.8392±0.0441
GLP-HPM [9]	–	41.582±2.217	0.900±0.034	1.650±0.392	1.588±0.405	0.0336±0.0129	0.1404±0.0277	0.8309±0.0334
GLP-FS [10]	–	41.565±2.125	0.897±0.035	1.655±0.385	1.589±0.395	0.0346±0.0137	0.1429±0.0282	0.8276±0.0348
BDSD-PC [8]	–	41.205±2.317	0.892±0.035	1.681±0.360	1.667±0.445	0.0759±0.0301	0.1548±0.0280	0.7812±0.0409
PanNet [17]	0.08M	46.268±2.031	0.967±0.010	0.997±0.212	0.919±0.191	0.0179±0.0110	0.0799±0.0178	0.9036±0.0198
MSDCNN [49]	0.23M	45.247±2.228	0.961±0.011	1.047±0.221	1.041±0.231	0.0243±0.0133	0.0730±0.0093	0.9044±0.0126
BDPN [18]	1.49M	42.080±2.625	0.923±0.024	1.481±0.326	1.546±0.432	0.0330±0.0223	0.0765±0.0199	0.8929±0.0250
FusionNet [23]	0.08M	45.663±2.270	0.964±0.009	0.974±0.212	0.988±0.222	0.0350±0.0124	0.1013±0.0134	0.8673±0.0179
MUCNN [50]	2.32M	48.256±1.930	0.979±0.008	0.808±0.171	0.731±0.146	0.0181±0.0093	0.0515±0.0088	0.9312±0.0107
LAGNet [51]	0.15M	48.760±1.447	0.980±0.009	0.786±0.148	0.687±0.113	0.0284±0.0130	0.0792±0.0136	0.8947±0.0200
PMACNet [52]	0.94M	45.041±2.135	0.963±0.011	1.359±0.133	1.248±0.204	0.0981±0.0215	0.0474±0.0115	0.8590±0.0171
U2Net [30]	0.66M	49.404±1.730	0.982±0.009	0.714±0.138	0.632±0.117	0.0236±0.0172	0.0510±0.0101	0.9265±0.0172
Pan-Mamba [44]	0.48M	48.931±1.811	0.982±0.008	0.743±0.156	0.684±0.129	0.0231±0.0110	0.0573±0.0116	0.9209±0.0148
CANNet [34]	0.78M	49.520±1.932	0.983±0.006	0.708±0.156	0.630±0.128	0.0194±0.0101	0.0630±0.0094	0.9188±0.0110
FusionMamba	0.73M	**49.678**±1.708	**0.984**±0.007	**0.705**±0.137	**0.615**±0.108	**0.0174**±0.0094	**0.0295**±0.0073	**0.9536**±0.0086
Ideal Values	–	**+∞**	**1**	**0**	**0**	**0**	**0**	**1**

the HyperPanCollection[2]. The data generation process follows Wald's protocol [53], with a detailed explanation provided in [59]. Specifically, the Pavia dataset contains 1680 training samples, of which 90% are allocated for training and 10% for validation. Additionally, this dataset includes two testing samples. Each training sample consists of an image triplet in the PAN/LRHS/GT format, with sizes of 64×64, $16\times16\times102$, and $64\times64\times102$, respectively. The testing samples comprise PAN/LRHS/GT image triplets sized 400×400, $100\times100\times102$, and $400\times400\times102$, respectively. The Botswana dataset contains 967 training samples, divided into 83% for training and 17% for validation, alongside four testing samples. Each training sample consists of a PAN/LRHS/GT image triplet with dimensions of 64×64, $16\times16\times145$, and $64\times64\times145$, respectively. The testing samples comprise PAN/LRHS/GT image triplets of sizes 128×128, $32\times32\times145$, and $128\times128\times145$. In the WDC dataset, there are 1024 training samples, divided into 90% for training and 10% for validation. Additionally, this dataset includes four testing samples. Each training sample contains a PAN/LRHS/GT image triplet of sizes 64×64, $16\times16\times191$, and $64\times64\times191$, respectively. The testing samples consist of PAN/LRHS/GT image triplets sized 128×128, $32\times32\times191$, and $128\times128\times191$.

2) Benchmarks: The proposed method is compared with several representative techniques, including four traditional approaches: GLP [6], GSA [60], CNMF [61], and Hysure [62]; as well as five DL-based methods: HyperPNN [19], HSpeNet series [63], FusionNet [23], Hyper-DSNet [59], and FPFNet [64]. For a fair comparison, all DL-based methods are trained using the same Nvidia GPU 3090 and PyTorch environment.

3) Quality Indices: In line with the research standards of the hyper-spectral pansharpening task, we select five widely used quality indices for evaluation, namely PSNR, cross-correlation (CC), SSIM [54], SAM, and ERGAS. The ideal values for these indices are $+\infty$, 1, 1, 0, and 0, respectively.

[2]https://github.com/liangjiandeng/HyperPanCollection

4) Settings: For hyper-spectral pansharpening, we set C to 48 and N to 4. Additionally, we employ the bicubic interpolation for up-sampling. Furthermore, in the U-shaped network branches, the number of channels in the feature maps remains constant across different stages, which slightly deviates from the depiction in Fig. 3. During the training of our networks on the Pavia, Botswana, and WDC datasets, the number of epochs is set to 1600, 3500, and 4000, respectively. Additionally, the batch size and initial learning rate are uniformly set to 32 and 2×10^{-4}. Furthermore, we use the Adam optimizer, with the learning rate halving every 1000 epochs. As for other DL-based methods, we follow the default settings specified in the corresponding papers or source codes.

5) Results: The quantitative evaluation results on three distinct datasets are presented in Table III. Clearly, our method significantly outperforms other techniques across all quality indices. Additionally, the qualitative evaluation outcomes, shown in Fig. 8, illustrate that FusionMamba produces fusion results that most closely resemble the GT images. Furthermore, Fig. 9 displays spectral vectors from various spatial locations of a WDC testing sample, highlighting the minimal spectral distortion achieved by our method. These results indicate that FusionMamba excels in the hyper-spectral pansharpening task.

C. Ablation Studies

1) Network Architecture: To validate the effectiveness of the proposed network architecture, we develop six variants of the FusionMamba and evaluate their performance using the reduced-resolution samples from the WV3 dataset, as detailed in Table IV. For fairness, all compared methods are designed with an identical number of network parameters. Specifically, we maintain a consistent spatial resolution of feature maps across different stages (w/o U-shape) to assess the efficacy of hierarchical information learning. Additionally, we remove the four-directional Mamba blocks from either the spatial branch (w/o Spatial Branch) or the spectral branch (w/o Spectral Branch) to determine the effectiveness of separate

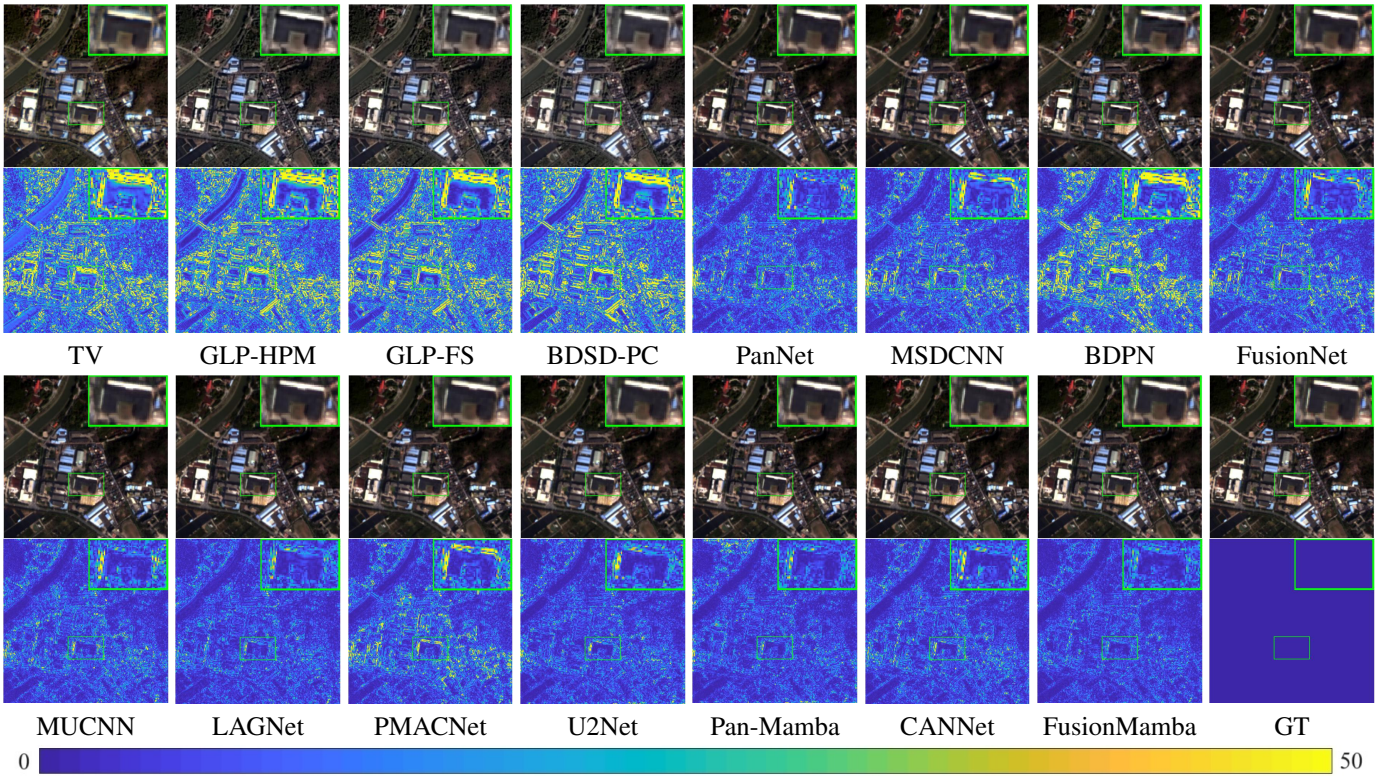


Fig. 7. Results on a reduced-resolution example from the GF2 dataset. Rows 1 and 3: Natural color images. Rows 2 and 4: AEMs for spectral band 3.

TABLE III
RESULTS ON TESTING SAMPLES OF THE WDC, BOTSWANA, AND PAVIA DATASETS, WHICH BELONG TO THE HYPER-SPECTRAL PANSHARPENING TASK.

Methods	Params	Pavia					Botswana					WDC				
		PSNR	CC	SSIM	SAM	ERGAS	PSNR	CC	SSIM	SAM	ERGAS	PSNR	CC	SSIM	SAM	ERGAS
GLP [6]	–	31.944	0.935	0.749	6.099	4.909	32.559	0.951	0.837	1.383	1.207	27.946	0.934	0.761	6.546	5.110
GSA [60]	–	31.501	0.937	0.722	6.282	4.978	31.739	0.939	0.828	1.389	1.386	24.462	0.906	0.671	7.846	6.079
CNMF [61]	–	31.184	0.894	0.659	6.953	6.263	30.220	0.917	0.788	1.934	1.718	24.604	0.890	0.678	8.441	6.682
Hysure [62]	–	32.208	0.921	0.730	6.240	5.474	30.610	0.928	0.796	1.747	1.595	25.598	0.913	0.718	7.254	5.834
HyperPNN [19]	0.13-0.14M	33.394	0.963	0.827	4.566	3.750	33.114	0.961	0.873	1.366	1.195	29.258	0.945	0.860	4.051	5.749
HSpeNet1 [63]	0.18-0.19M	33.612	0.964	0.824	4.690	3.721	31.746	0.942	0.844	1.456	1.663	29.634	0.960	0.870	4.039	4.266
HSpeNet2 [63]	0.11-0.13M	33.472	0.962	0.819	4.642	3.818	32.575	0.953	0.849	1.400	1.348	29.700	0.961	0.872	4.009	4.261
FusionNet [23]	0.21-0.26M	34.739	0.969	0.847	4.462	3.446	32.506	0.952	0.850	1.397	1.367	29.696	0.959	0.866	3.917	4.339
HyperDSNet [59]	0.18-0.31M	34.376	0.969	0.849	4.295	3.434	33.538	0.964	0.876	1.305	1.126	30.232	0.964	0.875	4.102	3.943
FPFNet [64]	3.00-3.06M	33.581	0.959	0.825	4.627	3.931	33.451	0.962	0.871	1.369	1.135	30.291	0.957	0.855	4.440	4.250
FusionMamba	0.44-0.51M	**35.628**	**0.973**	**0.872**	**3.963**	**3.171**	**33.943**	**0.966**	**0.881**	**1.277**	**1.076**	**31.860**	**0.965**	**0.881**	**3.755**	**3.882**
Ideal Values	–	**+∞**	**1**	**1**	**0**	**0**	**+∞**	**1**	**1**	**0**	**0**	**+∞**	**1**	**1**	**0**	**0**

feature extraction. Furthermore, we assess the validity of the combination branch by incorporating the FusionMamba blocks into the spectral branch (w/o Combination Branch). Finally, we remove the MCA module (w/o MCA) or replace it with the SENet [48] (w/ SENet) to evaluate the contribution of the MCA. The results strongly support the validity of our designs.

2) Structures of Mamba and FusionMamba Blocks: To validate the effectiveness of our structural designs for the Mamba and FusionMamba blocks, we develop five variants and evaluate them using the reduced-resolution samples from the WV3 dataset, as presented in Table V. Specifically, we assess the performance of the four-directional flattening technique [2] employed in our Mamba and FusionMamba blocks by comparing it against the one-directional (OD) flattening method, the bidirectional (BD) flattening approach [41], and the shuffle flattening technique [43]. Additionally, we remove either $\mathbf{F}^{\mathrm{a}}_{\mathrm{out}}$ (w/o $\mathbf{F}^{\mathrm{a}}_{\mathrm{out}}$) or $\mathbf{F}^{\mathrm{b}}_{\mathrm{out}}$ (w/o $\mathbf{F}^{\mathrm{b}}_{\mathrm{out}}$) from the FusionMamba block to verify the efficacy of our design. The quantitative results strongly affirm the validity of our structural designs for the Mamba and FusionMamba blocks.

3) The Application of FusionMamba Block: We investigate the potential of the FusionMamba block by incorporating it into several representative pansharpening frameworks, including PanNet [17], FusionNet [23], and U2Net [30]. In this process, we substitute the concatenation operation in PanNet and FusionNet, as well as the S2Block in U2Net, with the FusionMamba block. Table VI showcases the evaluation results on the reduced-resolution samples from the WV3 dataset. The

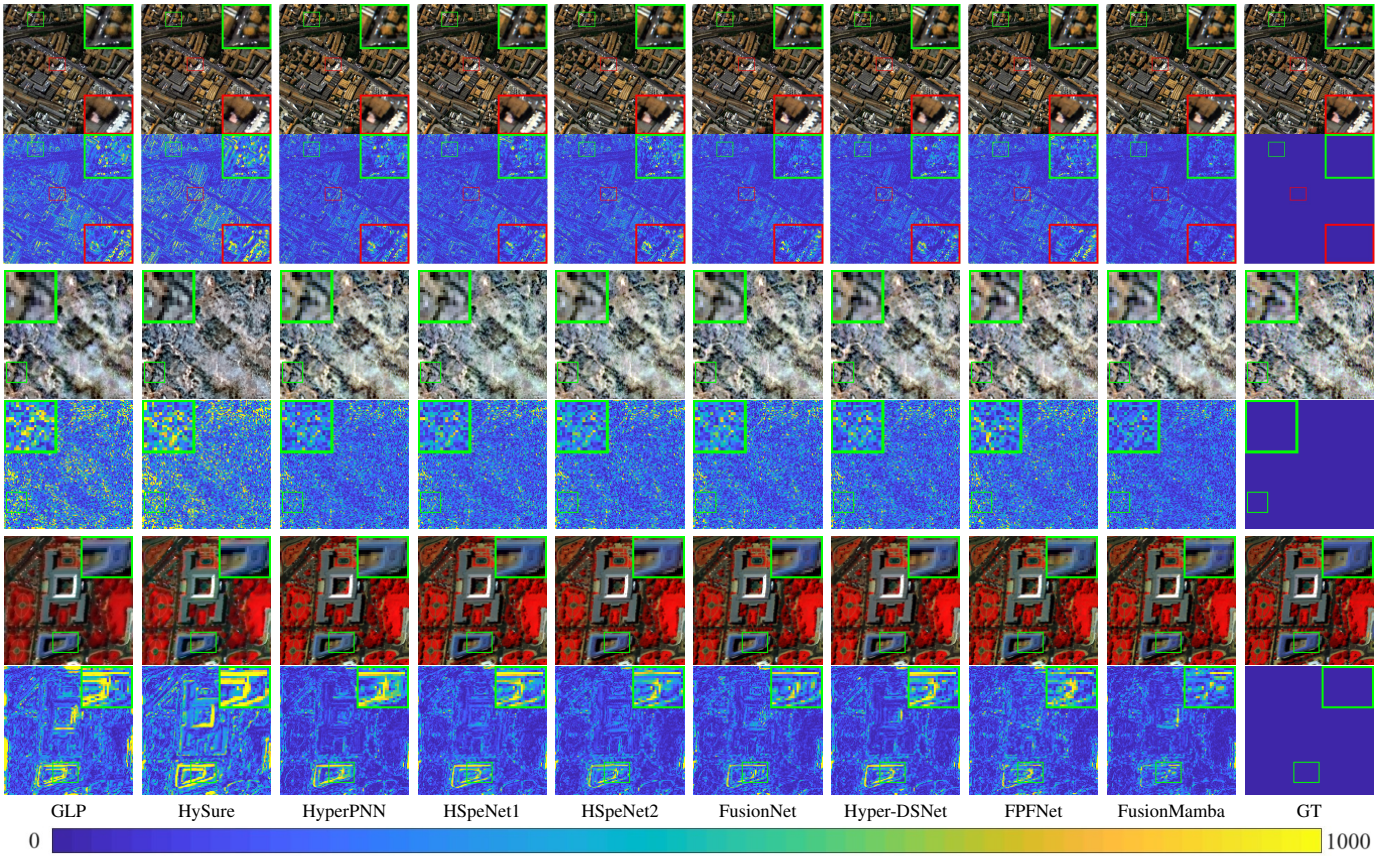
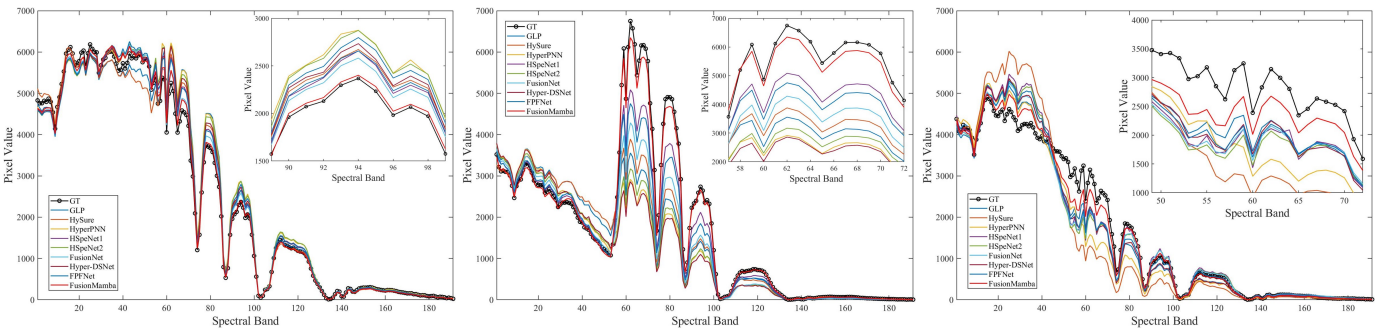


Fig. 8. Qualitative evaluation results on the Pavia, Botswana, and WDC datasets. Row 1: Pseudo-color images for spectral bands 20, 40, and 60 from a testing sample in the Pavia dataset. Row 2: AEMs for spectral band 68 from the testing sample in row 1. Row 3: Pseudo-color images for spectral bands 30, 50, and 70 from a testing sample in the Botswana dataset. Row 4: AEMs for spectral band 34 from the testing sample in row 3. Row 5: Pseudo-color images for spectral bands 20, 50, and 80 from a testing sample in the WDC dataset. Row 6: AEMs for spectral band 25 from the testing sample in row 5.



Spectral Vectors at $(44, 30)$ Spectral Vectors at $(80, 47)$ Spectral Vectors at $(115, 59)$

Fig. 9. Comparison of spectral vectors at three randomly selected spatial locations from a testing sample in the WDC dataset.

FusionMamba block demonstrates significant performance improvements, particularly when the baseline metrics are low. Moreover, even with high baseline performance, our method is still capable of exceeding the performance threshold. Consequently, the FusionMamba block proves to be an effective plug-and-play module for information integration.

4) FSSM Block: The main contribution of the FSSM block lies in its information interaction mechanism. Consequently, we investigate different interactive combinations of the projection parameters $\mathbf{B}$ and $\mathbf{C}$, along with the timescale parameter $\mathbf{\Delta}$. The quantitative evaluation results on 20 reduced-resolution samples from the WV3 dataset, as illustrated in Table VII, demonstrate the correctness of our strategy: one input generates the projection and timescale parameters, while the other input serves as the 1D sequence to be processed.

5) Feature Extraction and Information Integration Combinations: To emphasize the superiority of the Mamba and FusionMamba blocks, we systematically examine various combinations of feature extraction methods and information integration approaches. Specifically, the candidate feature extraction methods include the convolution (Conv) layer, self-attention (SA) module, and four-directional Mamba (Mamba) block. For information integration, we consider the concatenation (Concat) operation, cross-attention (CA) module, and

TABLE IV
ABLATION STUDY ON OUR NETWORK ARCHITECTURE USING 20 REDUCED-RESOLUTION SAMPLES FROM THE WV3 DATASET. ALL COMPARED METHODS HAVE AN IDENTICAL NUMBER OF PARAMETERS.

Methods	GFLOPs	PSNR	Q2n	SAM	ERGAS
w/o U-shape	124	39.219	0.921	2.855	2.134
w/o Spatial Branch	31	39.255	0.920	2.869	2.121
w/o Spectral Branch	31	39.333	0.921	2.848	2.097
w/o Combination Branch	31	39.316	0.921	2.853	2.101
w/o MCA	31	39.324	0.921	2.861	2.099
w/ SENet	31	39.294	0.920	2.850	2.100
FusionMamba	31	**39.374**	**0.922**	**2.843**	**2.092**
Ideal Values	–	**+∞**	**1**	**0**	**0**

TABLE V
ABLATION STUDY ON STRUCTURES OF MAMBA AND FUSIONMAMBA BLOCKS USING THE REDUCED-RESOLUTION SAMPLES FROM THE WV3 DATASET. ALL METHODS HAVE THE SAME NUMBER OF PARAMETERS.

Methods	PSNR	Q2n	SAM	ERGAS
OD Flattening	39.014	0.917	2.951	2.178
BD Flattening	39.163	0.920	2.886	2.144
Shuffle Flattening	39.117	0.919	2.898	2.152
w/o $\mathbf{F}_{\text{out}}^{\text{a}}$	39.309	0.921	2.855	2.106
w/o $\mathbf{F}_{\text{out}}^{\text{b}}$	39.293	0.921	2.857	2.110
FusionMamba	**39.374**	**0.922**	**2.843**	**2.092**
Ideal Values	**+∞**	**1**	**0**	**0**

TABLE VI
THE APPLICATION OF THE FUSIONMAMBA (FMAMBA) BLOCK IN VARIOUS PANSHARPENING FRAMEWORKS. ALL METHODS ARE ASSESSED USING THE REDUCED-RESOLUTION SAMPLES FROM THE WV3 DATASET.

Methods	Params	PSNR	Q2n	SAM	ERGAS
PanNet [17]	0.08M	37.346	0.891	3.613	2.664
PanNet + FMamba	0.09M	**38.178**	**0.904**	**3.236**	**2.418**
FusionNet [23]	0.08M	38.047	0.904	3.324	2.465
FusionNet + FMamba	0.09M	**38.604**	**0.914**	**3.092**	**2.294**
U2Net [30]	0.66M	39.117	**0.920**	2.888	2.149
U2Net + FMamba	0.83M	**39.181**	**0.920**	**2.885**	**2.132**
Ideal Values	–	**+∞**	**1**	**0**	**0**

TABLE VII
ABLATION STUDY ON INTERACTIVE COMBINATIONS OF THE PARAMETERS $\mathbf{B}$, $\mathbf{C}$, AND $\mathbf{\Delta}$ IN THE FSSM BLOCK. ALL METHODS ARE EVALUATED USING 20 REDUCED-RESOLUTION SAMPLES FROM THE WV3 DATASET.

Interactiveness			PSNR	Q2n	SAM	ERGAS
$\mathbf{B}$	$\mathbf{C}$	$\mathbf{\Delta}$				
×	×	×	39.124	0.919	2.903	2.154
✓	×	×	39.182	0.920	2.886	2.145
×	✓	×	39.157	0.919	2.893	2.150
×	×	✓	39.116	0.919	2.907	2.159
✓	✓	×	39.343	**0.922**	2.846	2.093
✓	×	✓	39.224	0.921	2.864	2.122
×	✓	✓	39.265	0.921	2.852	2.116
✓	✓	✓	**39.374**	**0.922**	**2.843**	**2.092**
Ideal Values			**+∞**	**1**	**0**	**0**

FusionMamba (FMamba) block. Notably, a convolution layer is applied in Concat to adjust the number of channels. Table VIII presents quantitative evaluation results on 20 reduced-resolution samples from the WV3 dataset. To ensure a fair comparison, all combinations are designed to have the same number of network parameters. The combination of Mamba + FMamba achieves the best results across all quality indices while maintaining a relatively low FLOP consumption, underscoring both the efficacy and efficiency of our method. Additionally, combinations with Mamba outperform those without it, indicating the effectiveness of the Mamba block in feature extraction. Moreover, combinations incorporating FMamba far exceed those without it, demonstrating the superiority of the FusionMamba block in merging different types of information.

V. DISCUSSION

A. Experiments for HISR

1) Dataset: To validate the effectiveness of FusionMamba for image fusion tasks beyond remote sensing, we perform experiments using the CAVE dataset[3], which belongs to the HISR task [72]. Initially introduced in [73], the CAVE dataset contains 32 RGB/HRHS image pairs with dimensions of $512 \times 512 \times 3$ and $512 \times 512 \times 31$, which are not directly suitable for training and testing purposes. During the data generation phase, we select 20 samples for training and reserve the remaining samples for testing. From the training HRHS images, we extract 3920 overlapped patches of size $64 \times 64 \times 31$ to serve as the GT images. Following this, a 3×3 Gaussian blur kernel with a standard deviation of 0.5 is employed to down-sample the GT images, creating LRHS samples of size $16 \times 16 \times 31$. Additionally, we segment the training RGB images into 3920 overlapped patches, each sized $64 \times 64 \times 3$, to match the spatial resolution of the GT images. This process generates 3920 training samples, each comprising an RGB/LRHS/GT image triplet of sizes $64 \times 64 \times 3$, $16 \times 16 \times 31$, and $64 \times 64 \times 31$, respectively. We process the testing data using a similar strategy, resulting in testing samples containing RGB/LRHS/GT image triplets sized $512 \times 512 \times 3$, $128 \times 128 \times 31$, and $512 \times 512 \times 31$.

2) Benchmarks, Metrics, and Settings: We compare FusionMamba with several representative techniques for HISR, including two traditional approaches, namely LTMR [65] and UTV [66], alongside six DL-based methods: ResTFNet [67], SSRNet [68], Fusformer [69], 3DT-Net [70], PSRT [71], and U2Net [30]. In accordance with the research standards of HISR, we choose four quality indices for evaluation: PSNR, SSIM, SAM, and ERGAS. Their ideal values are $+\infty$, 1, 0, and 0, respectively. For the network configuration, we set C and N to 64 and 4, respectively. Additionally, we utilize the bicubic interpolation for up-sampling. During training, we set the batch size, number of epochs, and initial learning rate to 32, 1100, and 2×10^{-4}, respectively. Besides, we employ the Adam optimizer and halve the learning rate every 500 epochs.

3) Results: The quantitative evaluation results, as presented in Table IX, indicate that our method outperforms all others. Additionally, the qualitative evaluation outcomes, as depicted in Fig. 10, illustrate that the FusionMamba produces fusion outputs that most closely match the GT images. These findings

[3]https://www.cs.columbia.edu/CAVE/databases/multispectral/

TABLE VIII
DIFFERENT COMBINATIONS OF FEATURE EXTRACTION METHODS AND INFORMATION INTEGRATION APPROACHES. ALL COMBINATIONS HAVE THE SAME NUMBER OF NETWORK PARAMETERS AND ARE EVALUATED USING 20 REDUCED-RESOLUTION SAMPLES FROM THE WV3 DATASET. NOTABLY, COMBINATIONS INCORPORATING SA OR CA REQUIRE SIGNIFICANTLY HIGHER FLOPS DUE TO THE QUADRATIC COMPLEXITY WITH RESPECT TO THE NUMBER OF INPUT TOKENS.

Methods	GFLOPs	PSNR	Q2n	SAM	ERGAS
Conv + Concat	20	38.729	0.917	3.013	2.258
Conv + CA	2511	38.885	0.917	2.995	2.211
Conv + FMamba	26	39.111	0.918	2.919	2.152
SA + Concat	2511	38.653	0.914	3.070	2.285
SA + CA	5003	38.690	0.915	3.052	2.266
SA + FMamba	2517	39.206	0.920	2.867	2.126
Mamba + Concat	26	38.822	0.917	2.958	2.230
Mamba + CA	2517	38.925	0.919	2.917	2.203
Mamba + FMamba	31	**39.374**	**0.922**	**2.843**	**2.092**
Ideal Values	–	$+\infty$	**1**	**0**	**0**

TABLE IX
QUANTITATIVE EVALUATION RESULTS ON THE TESTING SAMPLES FROM THE CAVE DATASET, WHICH BELONGS TO THE HISR TASK.

Methods	Params	PSNR	SSIM	SAM	ERGAS
LTMR [65]	–	36.543	0.963	6.711	5.387
UTV [66]	–	38.615	0.941	8.649	4.519
ResTFNet [67]	2.39M	45.584	0.994	2.764	2.313
SSRNet [68]	0.03M	48.620	0.995	2.542	1.636
Fusformer [69]	0.50M	49.983	0.994	2.203	2.534
3DT-Net [70]	3.16M	51.471	0.997	2.117	1.119
PSRT [71]	0.25M	50.595	0.997	2.146	2.001
U2Net [30]	2.65M	50.433	0.997	2.187	1.277
FusionMamba	2.58M	**51.658**	**0.998**	**2.021**	**1.081**
Ideal Values	–	$+\infty$	**1**	**0**	**0**

strongly demonstrate the effectiveness of FusionMamba in image fusion tasks beyond remote sensing applications.

B. Comparison between Transformers and Mamba

Both Transformers and Mamba demonstrate strong global perception capabilities, but Mamba consistently outperforms Transformers, as shown in Table VIII. This superiority can be attributed to two key factors. First, Mamba's approach to global perception aligns more intuitively with image processing principles, where the influence of neighboring pixels on the output is generally more significant. Second, Mamba dynamically learns projection and timescale parameters from the input, making it content-aware. In contrast, while Transformers offer some adaptability through their self/cross-attention mechanisms, they lack input-derived parameters, which limits their content-awareness compared to Mamba. Furthermore, as discussed in Section III-D4, Mamba exhibits significantly greater efficiency than Transformers. Consequently, Mamba emerges as a highly effective alternative to Transformers.

C. Comparison between Pan-Mamba and FusionMamba

As the pioneering application of the SSM in pansharpening, Pan-Mamba [44] employs stacked one-directional Mamba blocks to extract spatial and spectral features. For information integration, it merely adds the outputs of two Mamba blocks, resulting in limited interaction between spatial and spectral data. In contrast, the FusionMamba incorporates four-directional Mamba blocks into two U-shape network branches, allowing for the effective and hierarchical learning of spatial and spectral characteristics. Furthermore, the proposed FusionMamba block enhances information integration through data interaction at the SSM level. Consequently, the FusionMamba is more powerful and interpretable than Pan-Mamba.

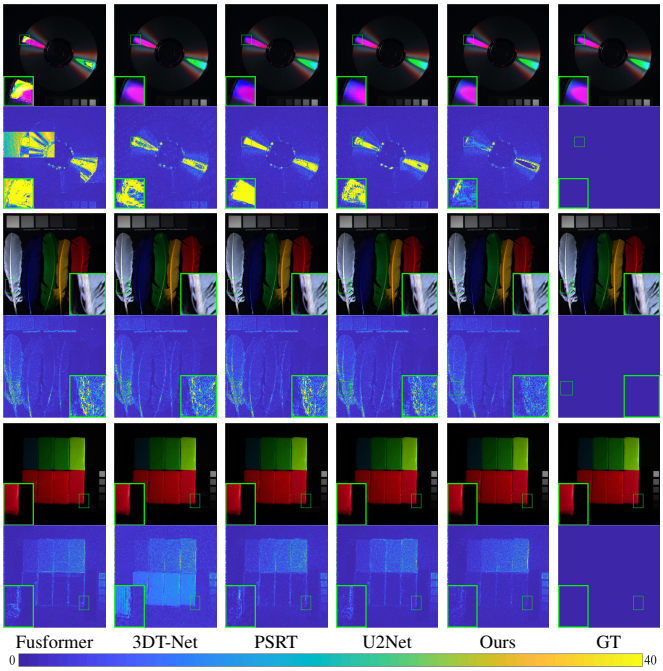


Fig. 10. Qualitative evaluation results of SOTA HISR methods on three testing samples from the CAVE dataset. Row 1: Pseudo-color images for spectral bands 6, 10, and 26 from the testing example *cd*. Row 2: AEMs for spectral band 13 from *cd*. Row 3: Pseudo-color images for spectral bands 6, 13, and 26 from the testing example *feathers*. Row 4: AEMs for spectral band 14 from *feathers*. Row 5: Pseudo-color images for spectral bands 6, 13, and 26 from the testing example *clay*. Row 6: AEMs for spectral band 8 from *clay*.

D. Strengths, Limitations, and Future Work

1) Strengths: First, the proposed FusionMamba block enables the effective combination of spatial and spectral features with only linear computational complexity, representing a significant advancement in the application of the SSM for combining different types of information. With an identical number of parameters, our method outperforms existing fusion techniques like concatenation and cross-attention. Furthermore, experimental results in Table VI demonstrate that the FusionMamba block can function as a plug-and-play module for information integration. Second, our interpretable network architecture allows for the separate and hierarchical learning of spatial and spectral features. This architecture also facilitates the progressive fusion of different information types and improves the representation of spectral characteristics. Therefore, the FusionMamba offers an optimal solution for image fusion.

2) Limitations: Our method exhibits two limitations. First, the FusionMamba block is designed to support only two

inputs, restricting its applicability in fusion tasks that involve more than two inputs. Second, the FusionMamba block requires both feature maps to have the same number of channels, which is less flexible than the concatenation operation.

3) Future Work: In the future, we plan to expand the dual-input FusionMamba block to accommodate an arbitrary number of inputs, each with a variable number of feature channels. Additionally, we will delve deeply into the theories of the SSM to foster further groundbreaking innovations.

VI. Conclusion

In this paper, we propose FusionMamba, an innovative method for efficient remote sensing image fusion. To sufficiently merge spatial and spectral features, we expand the single-input Mamba block to accommodate dual inputs, creating the FusionMamba block. This novel module surpasses existing fusion techniques such as concatenation and cross-attention, representing a successful application of the SSM for information integration. Besides, the FusionMamba block can serve as a plug-and-play module, effectively merging different types of information. Additionally, our interpretable network architecture supports the separate and hierarchical learning of spatial and spectral characteristics, facilitates the progressive combination of different feature maps, and enhances the representation of spectral information. We evaluate the performance of FusionMamba across six datasets covering three image fusion tasks: pansharpening, hyper-spectral pansharpening, and HISR. Our method yields exceptional results, demonstrating the superiority of FusionMamba in image fusion.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30. Curran Associates, Inc., 2017.

[2] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, and Y. Liu, "Vmamba: Visual state space model," *arXiv preprint arXiv:2401.10166*, 2024.

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

[4] G. Vivone, M. Dalla Mura, A. Garzelli, R. Restaino, G. Scarpa, M. O. Ulfarsson, L. Alparone, and J. Chanussot, "A new benchmark based on recent advances in multispectral pansharpening: Revisiting pansharpening with classical and emerging pansharpening methods," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 1, pp. 53–81, 2021.

[5] L. Loncan, L. B. de Almeida, J. M. Bioucas-Dias, X. Briottet, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. A. Licciardi, M. Simões, J.-Y. Tourneret, M. A. Veganzones, G. Vivone, Q. Wei, and N. Yokoya, "Hyperspectral pansharpening: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 27–46, 2015.

[6] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, and M. Selva, "Mtf-tailored multiscale fusion of high-resolution ms and pan imagery," *Photogramm. Eng. Remote Sens.*, vol. 72, no. 5, pp. 591–596, 2006.

[7] J. Choi, K. Yu, and Y. Kim, "A new adaptive component-substitution-based satellite image fusion by using partial replacement," *IEEE Trans. Geosci. Remote Sens.*, 2010.

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

[9] G. Vivone, R. Restaino, M. Dalla Mura, G. Licciardi, and J. Chanussot, "Contrast and error-based fusion schemes for multispectral image pansharpening," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 5, pp. 930–934, 2014.

[10] G. Vivone, R. Restaino, and J. Chanussot, "Full scale regression-based injection coefficients for panchromatic sharpening," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3418–3431, 2018.

[11] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "A new pansharpening algorithm based on total variation," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 318–322, 2013.

[12] J. Liu, C. Zhou, R. Fei, C. Zhang, and J. Zhang, "Pansharpening via neighbor embedding of spatial details," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4028–4042, 2021.

[13] Y. Yang, H. Lu, S. Huang, W. Wan, and L. Li, "Pansharpening based on variational fractional-order geometry model and optimized injection gains," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 2128–2141, 2022.

[14] K. Yan, M. Zhou, J. Huang, F. Zhao, C. Xie, C. Li, and D. Hong, "Panchromatic and multispectral image fusion via alternating reverse filtering network," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35. Curran Associates, Inc., 2022, pp. 21 988–22 002.

[15] H. Dai, Y. Yang, S. Huang, W. Wan, H. Lu, and X. Wang, "Pansharpening based on fuzzy logic and edge activity," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024.

[16] M. Giuseppe, C. Davide, V. Luisa, and S. Giuseppe, "Pansharpening by convolutional neural networks," *Remote Sens.*, vol. 8, no. 7, p. 594, 2016.

[17] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, and J. Paisley, "Pannet: A deep network architecture for pan-sharpening," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 1753–1761.

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

[19] L. He, J. Zhu, J. Li, A. Plaza, J. Chanussot, and B. Li, "Hyperpnn: Hyperspectral pansharpening via spectrally predictive convolutional neural networks," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 3092–3100, 2019.

[20] Y. Yang, W. Tu, S. Huang, and H. Lu, "Pcdrn: Progressive cascade deep residual network for pansharpening," *Remote Sens.*, vol. 12, no. 4, 2020.

[21] J. Liu, Y. Feng, C. Zhou, and C. Zhang, "Pwnet: An adaptive weight network for the fusion of panchromatic and multispectral images," *Remote Sens.*, vol. 12, no. 17, 2020.

[22] X. Fu, W. Wang, Y. Huang, X. Ding, and J. Paisley, "Deep multiscale detail networks for multiband spectral image sharpening," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2090–2104, 2021.

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

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

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

[26] W. Tu, Y. Yang, S. Huang, W. Wan, L. Gan, and H. Lu, "Mmdn: Multi-scale and multi-distillation dilated network for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[27] Y. Yan, J. Liu, S. Xu, Y. Wang, and X. Cao, "Md³net: Integrating model-driven and data-driven approaches for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.

[28] H. Lu, Y. Yang, S. Huang, X. Chen, B. Chi, A. Liu, and W. Tu, "Awfln: An adaptive weighted feature learning network for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, 2023.

[29] R. Dian, A. Guo, and S. Li, "Zero-shot hyperspectral sharpening," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12 650–12 666, 2023.

[30] S. Peng, C. Guo, X. Wu, and L.-J. Deng, "U2net: A general framework with spatial-spectral-integrated double u-net for image fusion," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*. New York, NY, USA: Association for Computing Machinery, 2023, p. 3219–3227.

[31] H. Gao, S. Li, J. Li, and R. Dian, "Multispectral image pan-sharpening guided by component substitution model," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–13, 2023.

[32] Y. Jia, Q. Hu, R. Dian, J. Ma, and X. Guo, "Paps: Progressive attention-based pan-sharpening," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 2, pp. 391–404, 2024.

[33] Y. Que, H. Xiong, X. Xia, J. You, and Y. Yang, "Integrating spectral and spatial bilateral pyramid networks for pansharpening," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 3985–3998, 2024.

[34] Y. Duan, X. Wu, H. Deng, and L.-J. Deng, "Content-adaptive non-local convolution for remote sensing pansharpening," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2024, pp. 27 738–27 747.

[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, and C. Ré, "Combining recurrent, convolutional, and continuous-time models with linear state space layers," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34. Curran Associates, Inc., 2021, pp. 572–585.

[37] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

[38] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Ré, "Hungry hungry hippos: Towards language modeling with state space models," *arXiv preprint arXiv:2212.14052*, 2022.

[39] J. T. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," *arXiv preprint arXiv:2208.04933*, 2022.

[40] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[41] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.

[42] C. Liu, K. Chen, B. Chen, H. Zhang, Z. Zou, and Z. Shi, "Rscama: Remote sensing image change captioning with state space model," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024.

[43] K. Chen, B. Chen, C. Liu, W. Li, Z. Zou, and Z. Shi, "Rsmamba: Remote sensing image classification with state space model," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024.

[44] X. He, K. Cao, K. Yan, R. Li, C. Xie, J. Zhang, and M. Zhou, "Pan-mamba: Effective pan-sharpening with state space model," *arXiv preprint arXiv:2402.12192*, 2024.

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

[46] K. Chen, Z. Zou, and Z. Shi, "Building extraction from remote sensing images with sparse token transformers," *Remote Sens.*, vol. 13, no. 21, 2021.

[47] W. Jiang, J. Zhang, D. Wang, Q. Zhang, Z. Wang, and B. Du, "LeMeViT: Efficient vision transformer with learnable meta tokens for remote sensing image interpretation," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2024.

[48] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2018, pp. 7132–7141.

[49] Y. Wei, Q. Yuan, X. Meng, H. Shen, L. Zhang, and M. Ng, "Multi-scale-and-depth convolutional neural network for remote sensed imagery pan-sharpening," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2017, pp. 3413–3416.

[50] Y. Wang, L.-J. Deng, T.-J. Zhang, and X. Wu, "Ssconv: Explicit spectral-to-spatial convolution for pansharpening," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*. New York, NY, USA: Association for Computing Machinery, 2021, p. 4472–4480.

[51] Z.-R. Jin, T.-J. Zhang, T.-X. Jiang, G. Vivone, and L.-J. Deng, "Lagconv: Local-context adaptive convolution kernels with global harmonic bias for pansharpening," *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 36, pp. 1113–1121, 2022.

[52] Y. Liang, P. Zhang, Y. Mei, and T. Wang, "Pmacnet: Parallel multiscale attention constraint network for pan-sharpening," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[53] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogramm. Eng. Remote Sens.*, vol. 63, pp. 691–699, 1997.

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

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

[56] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm," in *Proc. Summaries 3rd Annu. JPL Airborne Geosci. Workshop*, vol. 1, 1992, pp. 147–149.

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

[58] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016.

[59] Y.-W. Zhuo, T.-J. Zhang, J.-F. Hu, H.-X. Dou, T.-Z. Huang, and L.-J. Deng, "A deep-shallow fusion network with multidetail extractor and spectral attention for hyperspectral pansharpening," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 7539–7555, 2022.

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

[61] N. Yokoya, T. Yairi, and A. Iwasaki, "Coupled non-negative matrix factorization (cnmf) for hyperspectral and multispectral data fusion: Application to pasture classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2011, pp. 1779–1782.

[62] M. Simões, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image superresolution via subspace-based regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3373–3388, 2015.

[63] L. He, J. Zhu, J. Li, D. Meng, J. Chanussot, and A. Plaza, "Spectral-fidelity convolutional neural networks for hyperspectral pansharpening," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.g*, vol. 13, pp. 5898–5914, 2020.

[64] W. Dong, Y. Yang, J. Qu, Y. Li, Y. Yang, and X. Jia, "Feature pyramid fusion network for hyperspectral pansharpening," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, 2023.

[65] R. Dian and S. Li, "Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5135–5146, 2019.

[66] T. Xu, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, and J. Huang, "Hyperspectral image superresolution using unidirectional total variation with tucker decomposition," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4381–4398, 2020.

[67] X. Liu, Q. Liu, and Y. Wang, "Remote sensing image fusion based on two-stream fusion network," *Inf. Fusion*, vol. 55, pp. 1–15, 2020.

[68] X. Zhang, W. Huang, Q. Wang, and X. Li, "Ssr-net: Spatial–spectral reconstruction network for hyperspectral and multispectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5953–5965, 2021.

[69] J.-F. Hu, T.-Z. Huang, L.-J. Deng, H.-X. Dou, D. Hong, and G. Vivone, "Fusformer: A transformer-based fusion network for hyperspectral image super-resolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[70] Q. Ma, J. Jiang, X. Liu, and J. Ma, "Learning a 3d-cnn and transformer prior for hyperspectral image super-resolution," *Inf. Fusion*, vol. 100, p. 101907, 2023.

[71] S.-Q. Deng, L.-J. Deng, X. Wu, R. Ran, D. Hong, and G. Vivone, "Psrt: Pyramid shuffle-and-reshuffle transformer for multispectral and hyperspectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, 2023.

[72] X. Wang, Q. Hu, Y. Cheng, and J. Ma, "Hyperspectral image super-resolution meets deep learning: A survey and perspective," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 8, pp. 1668–1691, 2023.

[73] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2241–2253, 2010.