remote sensing

MDPI

Article

CCC-SSA-UNet: U-Shaped Pansharpening Network with Channel Cross-Concatenation and Spatial–Spectral Attention Mechanism for Hyperspectral Image Super-Resolution

Zhichao Liu [1,2], Guangliang Han [1,*], Hang Yang [1], Peixun Liu [1], Dianbing Chen [1], Dongxu Liu [3] and Anping Deng [1,2]

[1] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; liuzhichao201@mails.ucas.ac.cn (Z.L.); yanghang@ciomp.ac.cn (H.Y.); liupx@ciomp.ac.cn (P.L.); chendb@ciomp.ac.cn (D.C.); denganping20@mails.ucas.ac.cn (A.D.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610041, China; liudongxu18@mails.ucas.ac.cn
* Correspondence: hangl@ciomp.ac.cn

Abstract: A hyperspectral image (HSI) has a very high spectral resolution, which can reflect the target's material properties well. However, the limited spatial resolution poses a constraint on its applicability. In recent years, some hyperspectral pansharpening studies have attempted to integrate HSI with PAN to improve the spatial resolution of HSI. Although some achievements have been made, there are still shortcomings, such as insufficient utilization of multi-scale spatial and spectral information, high computational complexity, and long network model inference time. To address the above issues, we propose a novel U-shaped hyperspectral pansharpening network with channel cross-concatenation and spatial–spectral attention mechanism (CCC-SSA-UNet). A novel channel cross-concatenation (CCC) method was designed to effectively enhance the fusion ability of different input source images and the fusion ability between feature maps at different levels. Regarding network design, integrating a UNet based on an encoder–decoder architecture with a spatial–spectral attention network (SSA-Net) based on residual spatial–spectral attention (Res-SSA) blocks further enhances the ability to extract spatial and spectral features. The experiment shows that our proposed CCC-SSA-UNet exhibits state-of-the-art performance and has a shorter inference runtime and lower GPU memory consumption than most of the existing hyperspectral pansharpening methods.

Keywords: U-shaped fusion network; hyperspectral image super-resolution; hyperspectral pansharpening; channel cross-concatenation; spatial–spectral attention mechanism

Citation: Liu, Z.; Han, G.; Yang, H.; Liu, P.; Chen, D.; Liu, D.; Deng, A. CCC-SSA-UNet: U-Shaped Pansharpening Network with Channel Cross-Concatenation and Spatial–Spectral Attention Mechanism for Hyperspectral Image Super-Resolution. *Remote Sens.* **2023**, *15*, 4328. https://doi.org/10.3390/rs15174328

Academic Editor: Salah Bourennane

Received: 18 July 2023
Revised: 14 August 2023
Accepted: 16 August 2023
Published: 2 September 2023

1. Introduction

As a data cube, HSI often contains hundreds of spectral bands. HSI can reconstruct any point in space through its continuous and fine spectral curves, thus simultaneously obtaining the target's spatial and material properties. However, due to the limitations of sensors, HSI often has low spatial resolution. In many applications, such as fine classification of land cover [1], medical diagnosis [2], and anomaly detection [3], HSI is required to have the characteristics of high spatial resolution and hyperspectral resolution at the same time. For these reasons, HSI SR has become a research hotspot. At present, RGB image super-resolution technology is very mature. There are many methods for single RGB image super-resolution (SISR), including the interpolation method, reconstruction method, traditional machine-learning method, and deep learning method. However, because hyperspectral images have higher spectral dimensions than RGB images, HSI SR technology is more challenging than SISR. At present, HSI SR technology has achieved some research results but is still under development. How to improve the spatial resolution and maintain as much spectral information as possible is a pressing problem to be solved [4].

According to whether additional information is used (such as multispectral image (MSI), RGB image, or panchromatic image (PAN)), the existing HSI SR methods can be roughly classified into the following three categories: (1) single HSI SR method without any other auxiliary image; (2) RGB or MSI spectral super-resolution reconstruction method; and (3) the method based on fusion is to fuse HSI and its corresponding auxiliary image, and the method of fusing HSI and PAN is also called hyperspectral pansharpening.

Hyperspectral pansharpening has evolved from the research on pansharpening, which involves the fusion of MSI and PAN. This method achieves a balance between improving spatial resolution and maintaining spectral information. The study of MSI and PAN image fusion has been ongoing for several decades and has reached a performance bottleneck, with minimal differences in the performance of different fusion methods, making them visually indistinguishable. On the other hand, research on the fusion of HSI and PAN started relatively late and still holds significant potential for development.

Early pansharpening methods primarily relied on traditional techniques such as component substitution (CS) [5–8], multiresolution analysis (MRA) [9–11], and optimization-based methods [12–16]. In recent years, deep learning methods based on artificial neural networks have exhibited remarkable efficacy in diverse computer vision tasks. These tasks include high-level applications such as object detection [17,18], object tracking [19], and image classification [20], as well as low-level applications such as image denoising [21], image deblurring [22], and image super-resolution [23–27], achieving excellent results. Inspired by these studies, researchers have gradually started to introduce various deep learning methods into the field of hyperspectral pansharpening. Examples include HyperPNN [28] and Hyper-DSNet [29] based on convolutional neural networks (CNNs), DHP-DARN [30], and DIP-HyperKite [31] based on deep image prior network (DIP-Net) and CNNs, PS-GDANet [32], and HPGAN [33] based on generative adversarial networks (GANs), as well as HyperTransformer [34] based on Transformer.

After an extensive literature review, we identified the following issues with existing deep learning-based hyperspectral pansharpening methods. HyperPNN focuses solely on the fusion of single-scale information and fails to achieve satisfactory fusion quality. DIP-HyperKite employs an encoder–decoder network with layer-wise upsampling and downsampling for multiscale information fusion, which significantly increases computational complexity and GPU memory consumption. Both DHP-DAR and DIP-HyperKite utilize a two-step pansharpening approach, involving upsampling the LR-HSI using a deep prior network followed by image fusion using a convolutional neural network. This approach substantially increases the inference runtime of the network model, and the adopted deep prior network exhibits significant instability. While recent Transformer-based pansharpening networks such as HyperTransformer have achieved good fusion results, their large parameter and computational requirements pose high demands on computer performance, thereby reducing their practicality. To address the drawbacks of the aforementioned methods and effectively extract spatial and spectral features from the input HSI and PAN, we propose a U-shaped network with channel cross-connection and spatial–spectral attention mechanism.

The contributions of this paper are summarized as follows:

- We propose a novel framework for hyperspectral pansharpening named the CCC-SSA-UNet, which integrates the UNet architecture with the SSA-Net.
- We propose a novel channel cross-concatenation method called Input CCC at the network's entrance. This method effectively enhances the fusion capability of different input source images while introducing only a minimal number of additional parameters. Furthermore, we propose a Feature CCC approach within the decoder. This approach effectively strengthens the fusion capacity between different hierarchical feature maps without introducing any extra parameters or computational complexity.
- We propose an improved Res-SSA block to enhance the representation capacity of spatial and spectral features. Experimental results demonstrate the effectiveness

of our proposed hybrid attention module and its superiority over other attention module variants.

The remaining sections of this paper are organized as follows. Section 2 reviews related works about pansharpening methods, including classical pansharpening methods and deep learning-based methods. Section 3 provides a detailed exposition of the proposed methodology. Section 4 presents the experimental results and provides a comprehensive discussion. The conclusion of the paper is given in Section 5.

2. Related Work

HSI holds versatile applications in pansharpening [35], change detection [36], object detection [37], and classification [38] tasks, making it invaluable in various domains. This research specifically delves into the realm of pansharpening, where our focus lies. Within this section, we meticulously investigate both classical pansharpening methods and deep learning-based pansharpening methods, aiming to unravel their potential and advancements in this domain.

2.1. Classical Pansharpening Methods

Classical pansharpening methods can be roughly classified into the following three categories: CS, MRA, and optimization-based methods.

The CS approach first transforms the HSI into a new projection space, decomposes it into spectral components and spatial components, and then replaces its spatial components with a panchromatic image. Subsequently, the inverse transformation is applied to generate the reconstructed image. Typical CS approaches include IHS color space transformation [39], Gram–Schmidt (GS) transformation [5], Gram–Schmidt transformation with adaptive weights (GSA) [6], principal component analysis (PCA) [8], and guided filter principal component analysis (GFPCA) [7]. This category of methods is easy to implement and exhibits fast processing speeds while effectively preserving spatial information. However, these methods might introduce certain degrees of distortion to spectral information.

The MRA approach involves initially downsampling the HR-PAN image at multiple scales and decomposing it into high-frequency and low-frequency components. Subsequently, these components are then fused with the upsampled HSI according to various fusion rules, and finally, an inverse transformation is applied to obtain the reconstructed image. Typical methods within the MRA category can be classified as follows: the Laplacian pyramid method [40], the approach [41] based on undecimated discrete wavelet transform (UDWT) and the generalized Laplacian pyramid (GLP), the method using modulation transfer function and GLP (MTF-GLP) [9], and MTF-GLP with High Pass Modulation (MTF-GLP-HPM) [11], as well as the integration of MRA with CNNs, as seen in LPPNet [42]. The advantages of these methods lie in their ability to incorporate high-frequency spatial details into the HSI while preserving spectral information. However, they may result in the loss of some spatial information and introduce ringing artifacts.

CS-based and MRA-based methods are primarily employed in the field of MSI pansharpening. Due to the relatively low spatial resolution of HSI, pixel-level ambiguity often arises, rendering CS and MRA less suitable for addressing the fusion of HSI and PAN. Instead, optimization-based approaches are suitable to tackle this problem.

The core idea of the optimization-based approach lies in treating the fusion problem as an inverse reconstruction problem. By establishing a relationship model between the original image and the reference ground truth image, the model is mathematically optimized to obtain a solution. Bayesian estimation methods [12–14,43,44] and matrix factorization methods [15,16,45] are commonly used optimization-based methods. In contrast to the CS-based and MRA-based methods, these methods perform well in preserving both spatial and spectral information. However, due to their high computational demands, these methods also require massive computational resources.

2.2. Deep Learning-Based Pansharpening Methods

In the earlier research on deep learning-based pansharpening, the Pansharping Neural Network (PNN) [46] treated HR-PAN as an additional spectral band of LR-MSI, and employed three convolutional layers to learn the mapping relationship between the composited image of HR-PAN and LR-MSI and the reference ground truth MSI. However, this fusion method only combined the two input images at a basic level. Moreover, the utilized convolutional network was overly simplistic, which hindered the extraction of intricate spectral and spatial information. Consequently, the fusion performance was compromised. Yuan et al. [47] introduced a multi-scale and multi-depth CNN (MSDCNN) that improved upon the PNN by employing parallel multi-scale convolutional blocks to enhance the representational capacity.

Certain approaches employ a strategy of separately extracting features from the two input images before fusion. Liu et al. [48] proposed a TFNet, which employs two subnetworks to extract spectral and spatial features from the upsampled LR-MSI and the corresponding HR-PAN images, respectively. These features are then fused using a fusion network comprising multiple convolutional layers. Finally, an image reconstruction process is carried out using a reconstruction network.

In addition to fusing the features extracted from the two inputs, some methods propose the fusion of an original HR-PAN with deep features extracted from LR-HSI. He et al. [28] introduced two spectral prediction CNNs, called HyperPNN1 and HyperPNN2, which initially extract spectral features from the upsampled LR-HSI using two convolutional layers. These spectral features are then concatenated with the HR-PAN image and subjected to fusion reconstruction through multiple convolutional layers. While both HyperPNN1 and HyperPNN2 share a fundamental network structure, HyperPNN2 incorporates an additional residual structure with skip connections. These methods effectively extract features from the LR-HSI input. However, the process of feature fusion does not adequately account for the correlation between LR-HSI and HR-PAN. Additionally, some methods adopt the fusion of the upsampled LR-HSI with high-frequency detail features extracted from HR-PAN. Zhuo et al. [29] devised an HSI pansharpening network named Hyper-DSNet. This network employs five spatial domain high-pass filter templates to extract high-frequency detail characteristics from the HR-PAN. Subsequently, these extracted details are concatenated with the upsampled LR-HSI in the spectral dimension. The network architecture incorporates multi-scale convolutional modules, shallow-to-deep fusion structures, and a spectral attention mechanism. This method retains inherent spatial details and spectral fidelity.

In recent years, attention mechanisms have been widely applied in various computer vision tasks such as image super-resolution, object detection, and object recognition. The principle underlying attention mechanism is to automatically highlight the most informative components while suppressing less relevant ones, thereby enhancing computational efficiency. Hu et al. [49] initially introduced the channel attention mechanism, where a Squeeze-and-Excitation (SE) module, constructed using global average pooling along the spatial dimensions and two 1×1 convolutions, was employed to improve the object recognition performance of networks. Building upon the SE module, Roy et al. [50] proposed a concurrent spatial and channel Squeeze-and-Excitation (scSE) module, which utilized convolutional layers with $1 \times 1 \times C$ kernels and sigmoid activation functions to generate spatial attention maps. The scSE module then combined with the channel attention mechanism, yielding promising results in medical image segmentation. Motivated by this, Zheng et al. [30] proposed a hyperspectral pansharpening approach based on Deep Hyperspectral Prior (DHP) and Dual Attention Residual Network (DARN) that combines spatial–spectral attention mechanisms. In this approach, the DHP process solely employs spectral constraints, overlooking spatial constraints. Moreover, the fusion network only employs single-scale feature maps for fusion, neglecting multi-scale feature information. To overcome these limitations, Bandara et al. [31] introduced a novel spatial constraint in the Deep Image Prior (DIP) upsampling process and proposed the HyperKite network for

residual reconstruction. HyperKite employs an encoder–decoder network that sequentially performs upsampling and downsampling layers for multi-scale feature fusion. However, the simplistic encoder–decoder architecture in HyperKite hinders the extraction of fine spectral and spatial information. Additionally, the layered upsampling and downsampling design imposes a significant computational burden.

In addition to the aforementioned CNN-based pansharpening methods, in recent years, pansharpening approaches based on Generative Adversarial Networks (GANs) have also emerged. Dong et al. [32] developed a specific pansharpening framework using a Paired-Shared Generative Dual Adversarial Network (PS-GDANet), featuring two discriminators. The spatial discriminator enforces the similarity between the intensity component of the pansharpened image and the panchromatic (PAN) image, while the spectral discriminator aids in preserving the spectral characteristics of the original HSI image. This configuration enables the network to generate high-resolution pansharpened images. Xie et al. [33] introduced a high-dimensional pansharpening framework called HPGAN based on a 3D Generative Adversarial Network (3D-GAN) and devised a loss function that comprehensively considers global, spectral, and spatial constraints. Despite the favorable perceptual quality of images generated by GANs, the instability in generating images was not well received in the remote sensing field.

The Transformer architecture initially emerged in the field of natural language processing and was later introduced to computer vision. In recent years, with the continuous development and expansion of the Transformer, it has also made its mark in the domain of hyperspectral pansharpening. Bandara et al. [34] introduced a novel Transformer-based pansharpening network known as HyperTransformer. This network comprises three core modules: two separate PAN and HSI feature extractors, a multi-head feature attention module, and a spatial–spectral feature fusion module. Despite its enhancement of the spatial and spectral quality of the pansharpened HSI, the network's large parameter count and high computational load pose challenges to its practical applicability.

3. Proposed Method

This section will provide a detailed introduction to the proposed method from three aspects: problem statement and formulation, network architecture design, and loss function design.

3.1. Problem Statement and Formulation

Original hyperspectral image (LR-HSI) possesses high spectral resolution but suffers from low spatial resolution, whereas panchromatic image (PAN) exhibits high spatial resolution but lacks spectral information. Therefore, employing a fusion approach to combine these two types of image data is an effective means to obtain high-spatial-resolution hyperspectral image (HR-HSI). The main objective of this paper is to design a deep neural network model that can fuse LR-HSI and PAN to generate high-quality HR-HSI.

Let $\mathbf{X} \in \mathbb{R}^{h \times w \times C}$ represent LR-HSI, with a spatial resolution of $h \times w$ pixels and C spectral bands. Let $\mathbf{P} \in \mathbb{R}^{H \times W \times 1}$ represent PAN, with a spatial resolution of $H \times W$ pixels and a single spectral band. Let $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times C}$ denote the reconstructed hyperspectral image (HR-HSI), with a spatial resolution of $H \times W$ pixels and C spectral bands. Let $\mathbf{Y} \in \mathbb{R}^{H \times W \times C}$ represent the reference ground truth HR-HSI (Ref-HR-HSI), with a spatial resolution of $H \times W$ pixels and C spectral bands. Additionally, it is assumed that conditions $H > h$, $W > w$ and $C >> 1$ hold. Then, the training process of the HSI-PAN fusion network can be described as follows: The training dataset $\{[\mathbf{X}_1, \mathbf{P}_1, \mathbf{Y}_1], \cdots, [\mathbf{X}_D, \mathbf{P}_D, \mathbf{Y}_D]\}$ consists of D pairs of images. These images are processed by the neural network model $\Phi(\cdot, \cdot; \Theta)$, resulting in the output image set $\left[\hat{\mathbf{Y}}_1, \cdots, \hat{\mathbf{Y}}_D\right]$. The parameters Θ of the neural network are continuously optimized and adjusted using an optimization algorithm, aiming to minimize

the difference between $\hat{\mathbf{Y}}_d (1 \leq d \leq D)$ and $\mathbf{Y}_d$ until it converges to a certain value. The training process of the network can be represented by the following equation:

$$\hat{\Theta} = \text{argmin}_{\Theta} \frac{1}{D} \sum_{d=1}^{D} \text{Loss}(\hat{\mathbf{Y}}_d, \mathbf{Y}_d) \;\; s.t. \hat{\mathbf{Y}}_d = \Phi(\mathbf{X}_d, \mathbf{P}_d; \Theta) \tag{1}$$

where $\hat{\Theta}$ represents the optimized network parameters and $\text{Loss}(\cdot, \cdot)$ refers to the loss function employed by the network. The loss function quantifies the dissimilarity between the predicted output $\hat{\mathbf{Y}}_d$ and the desired target $\mathbf{Y}_d$, facilitating the training and optimization of the network.

During the testing phase, test image pairs $[\mathbf{X}_t, \mathbf{P}_t]$ are processed using a neural network model $\Phi(\cdot, \cdot; \hat{\Theta})$ with pre-trained parameters $\hat{\Theta}$, resulting in the final output fused image $\hat{\mathbf{Y}}_t$. The testing process of the network can be represented by the following equation:

$$\hat{\mathbf{Y}}_t = \Phi(\mathbf{X}_t, \mathbf{P}_t; \hat{\Theta}) \tag{2}$$

where, $\mathbf{X}_t$ and $\mathbf{P}_t$ respectively represent the input LR-HSI and PAN images used for testing, while $\hat{\mathbf{Y}}_t$ denotes the fused image, which is the final output of the network.

The schematic diagram of the training and testing phases of the deep learning-based HSI-PAN fusion network is illustrated in Figure 1.

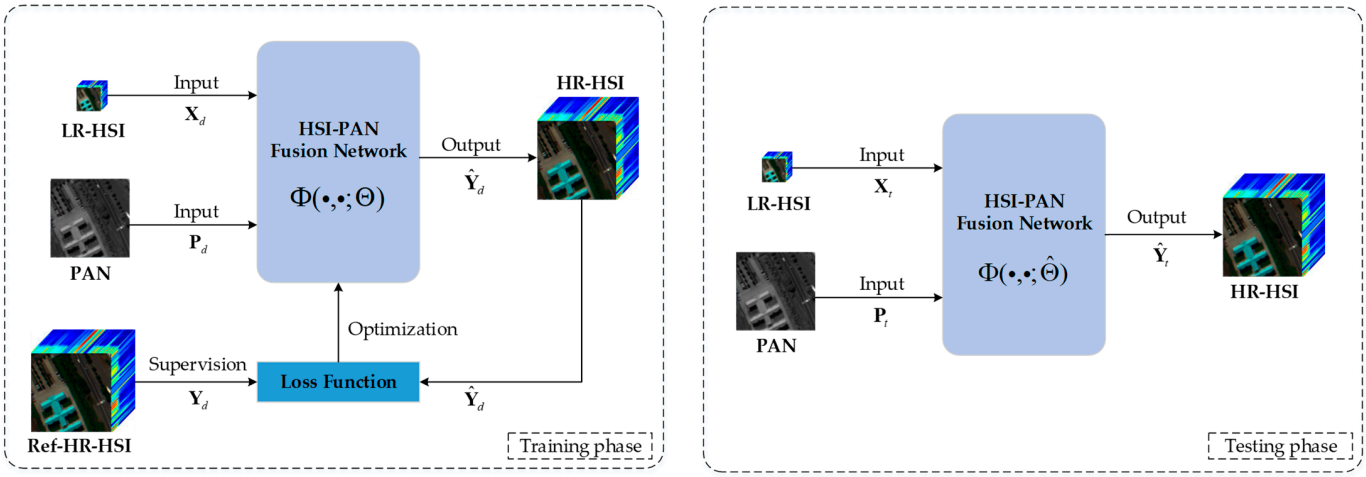


Figure 1. The schematic diagram of the training phase and testing phase of the deep learning-based HSI-PAN fusion network. (**a**) Training phase; (**b**) testing phase.

3.2. Network Design

Figure 2 illustrates the overall network architecture of the proposed CCC-SSA-UNet. CCC-SSA-UNet takes an LR-HSI (represented by $\mathbf{X} \in \mathbb{R}^{h \times w \times C}$) and a PAN (represented by $\mathbf{P} \in \mathbb{R}^{H \times W \times 1}$) as initial inputs and outputs an HR-HSI (represented by $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times C}$). Following the design of DHP-DARN [30] and DIP-HyperKite [31], our pansharpening network CCC-SSA-UNet adopts a residual learning-based framework. Firstly, LR-HSI $\mathbf{X}$ is upsampled using bilinear interpolation to obtain the image $\mathbf{U} \in \mathbb{R}^{H \times W \times C}$, which has the same spatial resolution as $\mathbf{P}$. Then, the proposed Input CCC method is applied to cross-concatenate $\mathbf{U}$ and $\mathbf{P}$ in the channel dimension, resulting in the image $\mathbf{O} \in \mathbb{R}^{H \times W \times (C+m)}$. Image $\mathbf{O}$ is fed into a U-shaped network to learn the residual image $\mathbf{X}_{res} \in \mathbb{R}^{H \times W \times C}$ for HR-HSI. Finally, the residual image $\mathbf{X}_{res}$ is pixel-wise added to the image $\mathbf{U}$ to obtain the

final fusion result $\hat{\mathbf{Y}}$. The aforementioned process can be described using the following equation:

$$\mathbf{U} = \uparrow(\mathbf{X}) \tag{3}$$

$$\mathbf{X}_{res} = f_{CCC-SSA-UNet}(\mathbf{U}, \mathbf{P}) \tag{4}$$

$$\hat{\mathbf{Y}} = \mathbf{U} + \mathbf{X}_{res} \tag{5}$$

where $\uparrow(\cdot)$ represents bilinear interpolation for upsampling, while $f_{CCC-SSA-UNet}(\cdot,\cdot)$ represents the proposed CCC-SSA-UNet network introduced in this paper.

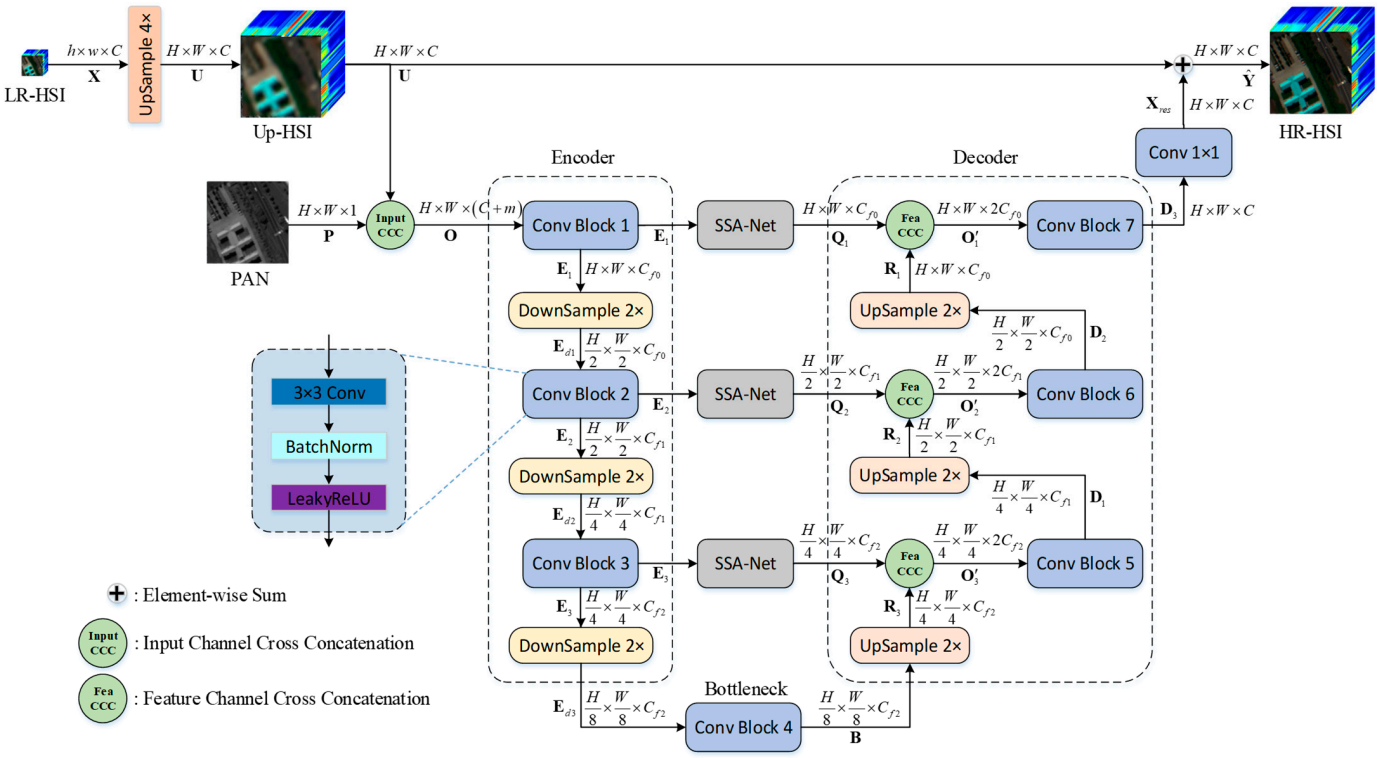


Figure 2. The architecture of the proposed CCC-SSA-UNet. CCC-SSA-UNet takes an LR-HSI and a PAN as input and takes an HR-HSI as output. It combines UNet and SSA-Net, exploits the "Conv Block" as the main building block of the UNet backbone, and adopts the Res-SSA block as the main building block of the SSA-Net. In each "Conv Block" of the UNet, the input is first passed through a 3 × 3 convolution and subsequently is passed through a batch normalization layer (BN) and a LeakyReLU layer. In this way, the feature map is extracted and passed to the next layer. Finally, the output of UNet is passed through a 1 × 1 convolution to produce the residual map of the Up-HSI and then the residual map and Up-HSI are element-wise summed to produce the final output HR-HSI. "DownSample 2×" denotes 2 × 2 maxpooling, and "UpSample 2×" and "UpSample 4×" denote bilinear interpolation with scale 2 and scale 4, respectively. "Input CCC" and "Fea CCC" denote channel cross-concatenation of input images and feature maps, respectively.

The main idea behind CCC-SSA-UNet is to integrate the U-Net [51], based on an encoder–decoder architecture, with the SSA-Net, which incorporates spatial–spectral residual attention modules. The encoder–decoder architecture is applicable in many areas such as medical science [52], HSI classification [53], and agriculture science [54]. Firstly, we construct a U-shaped encoder–decoder network similar to U-Net, named UNet. Within UNet, we introduce the SSA-Net, which utilizes spatial–spectral attention mechanisms, between the layers of the encoder and their corresponding decoder counterparts. This design aims to enhance the expression capability of both spatial and spectral features.

Additionally, we propose novel channel cross-concatenation methods, namely Input CCC and Feature CCC, at the network's entrance and within the decoder, respectively. These methods effectively enhance the fusion capability of different input source images and the fusion capability between different hierarchical feature maps while minimizing additional computational complexity.

3.2.1. UNet Backbone

The network design of CCC-SSA-UNet draws inspiration from the state-of-the-art RGB image denoising method DRUNet [55] and the hyperspectral pansharpening method DIP-HyperKite [31]. Similar to DRUNet, our UNet backbone network consists of four scales with skip connections between the encoder and decoder at each scale. The number of channels at each layer varies across scales, denoted as C_{f0}, C_{f1}, C_{f2}, and C_{f2} from the first to the fourth scale, respectively. These parameters are determined through experimentation, and the specific parameter settings will be described in Section 4.3.

The UNet backbone network we propose is composed of several key components. The encoder consists of three Conv Block modules and three downsampling modules, which are arranged in an alternating manner. Similarly, the decoder follows the same structure, with three upsampling modules and three Conv Block modules. The skip connections between each layer of the encoder and its corresponding decoder are equipped with the SSA-Net, enhancing the fusion and representation capabilities. Additionally, a Bottleneck layer, comprising one Conv Block module, resides between the last downsampling module and the first upsampling module, facilitating the information flow between the encoder and decoder pathways.

The Conv Block module consists of consecutive layers, including a 3×3 convolutional layer with a stride of 1, a batch normalization layer, and a LeakyReLU activation function layer. This module plays a crucial role in feature extraction in the encoder and feature reconstruction in the decoder. Mathematically, the Conv Block module can be represented as follows:

$$\mathbf{CB}_{out} = f_{CB}(\mathbf{CB}_{in}) = \delta(\mathrm{BN}(\mathrm{Conv}_{3\times3}(\mathbf{CB}_{in}))) \tag{6}$$

where $\mathbf{CB}_{in}$ represents the input and $\mathbf{CB}_{out}$ represents the output of the Conv Block module, $f_{CB}(\cdot)$ represents the function representation of the Conv Block module, $\delta(\cdot)$ represents the LeakyReLU activation function, $\mathrm{BN}(\cdot)$ represents the batch normalization layer, and $Conv_{3\times3}(\cdot)$ represents the 3×3 convolutional layer. It should be noted that the Conv Block module sequentially connects these layers to effectively capture and process the input features throughout the network.

In the encoder section, the output image $\mathbf{O} \in \mathbb{R}^{H\times W\times(C+m)}$ from Input CCC undergoes a series of operations. First, it passes through the first Conv Block module, resulting in the first-level feature map $\mathbf{E}_1 \in \mathbb{R}^{H\times W\times C_{f0}}$. Subsequently, $\mathbf{E}_1$ is processed by a downsampling layer, yielding a feature map $\mathbf{E}_{d1} \in \mathbb{R}^{H/2\times W/2\times C_{f0}}$ with half the spatial dimensions. $\mathbf{E}_{d1}$ then undergoes the second Conv Block module, generating the second-level feature map $\mathbf{E}_2 \in \mathbb{R}^{H/2\times W/2\times C_{f1}}$, which is further downsampled to obtain the feature map $\mathbf{E}_{d2} \in \mathbb{R}^{H/4\times W/4\times C_{f1}}$. Similarly, $\mathbf{E}_{d2}$ goes through the third Conv Block module, producing the third-level feature map $\mathbf{E}_3 \in \mathbb{R}^{H/4\times W/4\times C_{f2}}$, followed by downsampling to obtain the feature map $\mathbf{E}_{d3} \in \mathbb{R}^{H/8\times W/8\times C_{f2}}$. Finally, $\mathbf{E}_{d3}$ is processed by the fourth Conv Block module, generating the feature map $\mathbf{B} \in \mathbb{R}^{H/8\times W/8\times C_{f2}}$ in the Bottleneck layer. This process can be mathematically represented as follows:

$$\mathbf{E}_1 = f_{CB1}(\mathbf{O}) \tag{7}$$

$$\mathbf{E}_2 = f_{CB2}(\mathbf{E}_{d1}) = f_{CB2}(\downarrow(\mathbf{E}_1)) \tag{8}$$

$$\mathbf{E}_3 = f_{CB3}(\mathbf{E}_{d2}) = f_{CB3}(\downarrow(\mathbf{E}_2)) \tag{9}$$

$$\mathbf{B} = f_{CB4}(\mathbf{E}_{d3}) = f_{CB4}(\downarrow(\mathbf{E}_3)) \tag{10}$$

where $f_{CBi}(\cdot), i = 1, 2, 3, 4$ represents the functional notation for the i-th Conv Block module, which greatly contributes to feature extraction and reconstruction. On the other hand, $\downarrow(\cdot)$ signifies the 2×2 maxpooling operation, employed for downsampling the feature maps to reduce their spatial dimensions.

Subsequently, the feature maps from the first three levels denoted as $\mathbf{E}_1$, $\mathbf{E}_2$, and $\mathbf{E}_3$ undergo spatial and spectral feature enhancement using the respective SSA-Net modules at their corresponding levels. This enhancement process yields refined feature maps $\mathbf{Q}_1 \in \mathbb{R}^{H \times W \times C_{f0}}$, $\mathbf{Q}_2 \in \mathbb{R}^{H/2 \times W/2 \times C_{f1}}$, and $\mathbf{Q}_3 \in \mathbb{R}^{H/4 \times W/4 \times C_{f2}}$, which embody strengthened spatial and spectral characteristics. These transformations can be mathematically represented by the equations as follows:

$$\mathbf{Q}_1 = f_{SSA-Net1}(\mathbf{E}_1) \tag{11}$$

$$\mathbf{Q}_2 = f_{SSA-Net2}(\mathbf{E}_2) \tag{12}$$

$$\mathbf{Q}_3 = f_{SSA-Net3}(\mathbf{E}_3) \tag{13}$$

where $f_{SSA-Net1}(\cdot)$, $f_{SSA-Net2}(\cdot)$, and $f_{SSA-Net3}(\cdot)$ denote the functional representations of the SSA-Net modules at the three corresponding levels. The comprehensive design details of these modules will be presented in Section 3.2.3.

Within the decoder section, the feature map $\mathbf{B}$ is initially subjected to an upsampling operation, yielding the feature map $\mathbf{R}_3 \in \mathbb{R}^{H/4 \times W/4 \times C_{f2}}$. Subsequently, the Feature CCC method is utilized to concatenate the feature map $\mathbf{R}_3$ with the output feature map $\mathbf{Q}_3$ from the third-level SSA-Net, leading to the formation of the feature map $\mathbf{O}'_3 \in \mathbb{R}^{H/4 \times W/4 \times 2C_{f2}}$. Following this, $\mathbf{O}'_3$ is processed through the fifth Conv Block module, resulting in the refined feature map $\mathbf{D}_1 \in \mathbb{R}^{H/4 \times W/4 \times C_{f2}}$, which is further upsampled to generate the feature map $\mathbf{R}_2 \in \mathbb{R}^{H/2 \times W/2 \times C_{f1}}$. Simultaneously, the feature map $\mathbf{R}_2$ is merged with the output feature map $\mathbf{Q}_2$ from the second-level SSA-Net via channel concatenation, resulting in the composite feature map $\mathbf{O}'_2 \in \mathbb{R}^{H/2 \times W/2 \times 2C_{f1}}$. $\mathbf{O}'_2$ is then passed through the sixth Conv Block module, generating the enhanced feature map $\mathbf{D}_2 \in \mathbb{R}^{H/2 \times W/2 \times C_{f0}}$. Further upsampling is performed on $\mathbf{D}_2$, producing the feature map $\mathbf{R}_1 \in \mathbb{R}^{H \times W \times C_{f0}}$. Lastly, the feature map $\mathbf{R}_1$ is merged with the output feature map $\mathbf{Q}_1$ from the first-level SSA-Net through channel concatenation, yielding the combined feature map $\mathbf{O}'_1 \in \mathbb{R}^{H \times W \times 2C_{f0}}$. $\mathbf{O}'_1$ subsequently undergoes processing through the seventh Conv Block module, culminating in the final feature map $\mathbf{D}_3 \in \mathbb{R}^{H \times W \times C}$. These transformations can be mathematically represented by the equations as follows:

$$\mathbf{O}'_3 = \text{FeaCCC}(\mathbf{Q}_3, \mathbf{R}_3) = \text{FeaCCC}(\mathbf{Q}_3, \uparrow(\mathbf{B})) \tag{14}$$

$$\mathbf{D}_1 = f_{CB5}(\mathrm{O}'_3) \tag{15}$$

$$\mathbf{O}'_2 = \text{FeaCCC}(\mathbf{Q}_2, \mathbf{R}_2) = \text{FeaCCC}(\mathbf{Q}_2, \uparrow(\mathbf{D}_1)) \tag{16}$$

$$\mathbf{D}_2 = f_{CB6}(\mathrm{O}'_2) \tag{17}$$

$$\mathbf{O}'_1 = \text{FeaCCC}(\mathbf{Q}_1, \mathbf{R}_1) = \text{FeaCCC}(\mathbf{Q}_1, \uparrow(\mathbf{D}_2)) \tag{18}$$

$$\mathbf{D}_3 = f_{CB7}(\mathrm{O}'_1) \tag{19}$$

where $f_{CBi}(\cdot), i = 5, 6, 7$ represents the functional notation for the i-th Conv Block module, while $\uparrow(\cdot)$ denotes the bilinear interpolation operation for upsampling, which increases

the spatial resolution of the feature maps by a factor of 2. Meanwhile, FeaCCC$(\cdot)$ denotes the specific procedure of the proposed Feature CCC, which will be elaborated on in Section 3.2.2.

Finally, following the encoder section, we introduce a 1×1 convolutional layer for the reconstruction of residual maps. This process can be mathematically represented as follows:

$$\mathbf{X}_{res} = \mathrm{Conv}_{1\times1}(\mathbf{D}_3) \tag{20}$$

where the symbol $\mathrm{Conv}_{1\times1}(\cdot)$ represents a 1×1 convolutional layer, and $\mathbf{X}_{res} \in \mathbb{R}^{H\times W\times C}$ represents the residual image obtained after reconstruction.

The aforementioned details elucidate the design specifics of our proposed CCC-SSA-UNet backbone network, which draws inspiration from the design principles of DRUNet [55] and DIP-HyperKite [31] while featuring notable differences. CCC-SSA-UNet distinguishes itself from DRUNe in three key aspects. Firstly, our backbone network employs 2×2 maxpooling downsampling and bilinear interpolation upsampling, in contrast to DRUNet's use of 2×2 stride convolution (SConv) and 2×2 transpose convolution (TConv). This design choice reduces the parameter count in the sampling layers. Secondly, CCC-SSA-UNet utilizes a single Conv Block in each encoder or decoder block, as opposed to DRUNet's employment of four residual convolution blocks. This design decision reduces the complexity of the encoder–decoder network. Lastly, CCC-SSA-UNet places the skip connections before each downsampling layer of the encoder and after the corresponding upsampling layer of the decoder, in contrast to DRUNet's placement after each downsampling layer and before each upsampling layer. This arrangement maximizes the preservation of extracted features by the encoder, mitigating spatial information loss resulting from immediate upsampling following downsampling.

The CCC-SSA-UNet and DIP-HyperKite [31] exhibit three notable differences. First, DIP-HyperKite employs an architecture that performs layer-wise upsampling followed by downsampling, while our CCC-SSA-UNet adheres to the U-Net [51] architecture, which performs downsampling followed by upsampling. This design choice results in reduced intermediate feature map size, decreased computational complexity, and minimized GPU memory consumption. Second, DIP-HyperKite incorporates a DIP-Net for upsampling preprocessing of LR-HSI, while our CCC-SSA-UNet directly employs bilinear interpolation for upsampling, significantly reducing the inference time of the network model. Last, DIP-HyperKite employs direct skip connections between each encoder and its corresponding decoder, whereas our CCC-SSA-UNet integrates the SSA-Net, leveraging spatial–spectral attention mechanisms, between each encoder and its corresponding decoder. This design choice further enhances the representation capability of spatial–spectral features.

3.2.2. CCC

To enhance the fusion capability of different information sources, we propose a novel channel cross-concatenation method, referred to as CCC, which is shown in Figure 3. Based on the nature of the input sources, CCC can be further divided into two categories: Input CCC and Feature CCC. Input CCC refers to the channel cross-concatenation method between the HSI and PAN inputs, aiming to enhance the fusion capability of different input source images. On the other hand, Feature CCC refers to the channel cross-concatenation method between two feature maps, aiming to enhance the fusion capability between feature maps at different levels.

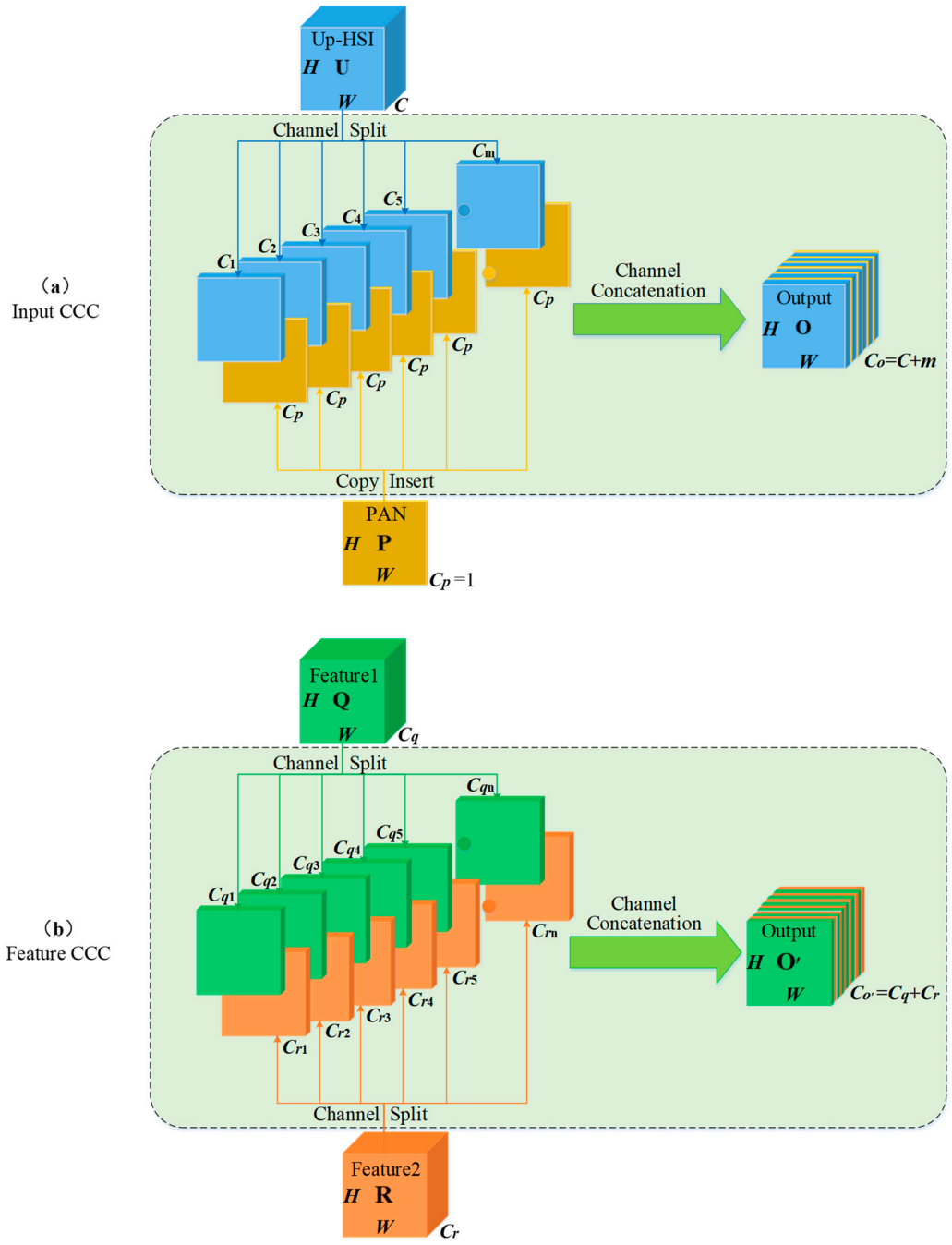


Figure 3. Schematic illustration of the proposed CCC method. (**a**) Schematic illustration of the proposed Input CCC operation. The tensors corresponding to Up-HSI and PAN are taken as inputs, and the result of channel cross-concatenation is used as the output. The spatial resolution of Up-HSI is $H \times W$ pixels, and it has a spectral bandwidth of C. The spatial resolution of PAN is $H \times W$ pixels, and it has a spectral bandwidth of C_p. The spatial resolution of output is $H \times W$ pixels, and it has a spectral bandwidth of C_o. (**b**) Schematic illustration of the proposed Feature CCC operation. Feature 1 denotes the first input feature map of Feature CCC, and Feature 2 denotes the second input feature map of Feature CCC. The tensors corresponding to Feature 1 and Feature 2 are taken as inputs, and the result of channel cross-concatenation is used as the output. The spatial resolution of Feature 1 is $H \times W$ pixels, and it has a spectral bandwidth of C_q. The spatial resolution of Feature 2 is $H \times W$ pixels, and it has a spectral bandwidth of C_r. The spatial resolution of output is $H \times W$ pixels, and it has a spectral bandwidth of $C_{o'}$.

- Input CCC

Figure 3a illustrates the working principle of Input CCC. The input consists of two tensors corresponding to different source images, namely Up-HSI and PAN. The output is the tensor obtained by performing channel cross-concatenation.

First, the Up-HSI is divided into m parts along the channel dimension, ensuring that the first $m-1$ parts have the same number of channels, and the number of channels in the m-th part should not exceed that of each previous part. Specifically, we set:

$$C = \sum_{i=1}^{m} C_i = C_1 + \cdots + C_{m-1} + C_m \tag{21}$$

where $C_1 = \cdots = C_{m-1} \geq C_m$ when $m \geq 2$. Particularly, $C_1 = C$ when $m = 1$, which signifies that no splitting is performed on Up-HSI.

The aforementioned splitting process can be represented by the following formula:

$$\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_{m-1}, \mathbf{U}_m = \text{Split}(\mathbf{U}) \tag{22}$$

where $\mathbf{U} \in \mathbb{R}^{H \times W \times C}$ denotes the tensor corresponding to Up-HSI, while $\text{Split}(\cdot)$ signifies the operation of tensor splitting. Notably, $\mathbf{U}_i \in \mathbb{R}^{H \times W \times C_i} (1 \leq i \leq m)$ denotes the individual sub-tensors obtained through the splitting of $\mathbf{U}$.

In the subsequent steps, the sub-tensor $\mathbf{U}_i$ obtained from the splitting, with a channel number of C_i, is sequentially inserted after the PAN tensor $\mathbf{P}$, which has a channel number of $C_p = 1$. By concatenating them along the channel dimension, the resulting tensor $\mathbf{O}$ is obtained, with the number of channels in the output tensor $\mathbf{O}$ being determined as follows:

$$\begin{aligned} C_{\mathrm{O}} &= \sum_{i=1}^{m} (C_i + C_p) \\ &= C_1 + C_p + \cdots + C_m + C_p \\ &= C + m \cdot C_p \\ &= C + m \end{aligned} \tag{23}$$

The process of channel cross-concatenation mentioned above can be expressed mathematically as follows:

$$\mathbf{O} = \text{Concat}(\mathbf{U}_1, \mathbf{P}, \mathbf{U}_2, \mathbf{P}, \cdots, \mathbf{U}_m, \mathbf{P}) \tag{24}$$

where $\mathbf{P} \in \mathbb{R}^{H \times W \times C_p}$ represents the tensor corresponding to PAN, while $\text{Concat}(\cdot)$ denotes the operation of channel concatenation. $\mathbf{O} \in \mathbb{R}^{H \times W \times C_{\mathrm{O}}}$ signifies the resulting tensor.

In summary, the entire process of channel-wise cross-concatenation between Up-HSI and PAN can be represented by the following equation:

$$\mathbf{O} = \text{InputCCC}(\mathbf{U}, \mathbf{P}) \tag{25}$$

where $\text{InputCCC}(\cdot)$ represents the operation process of Input CCC.

- Feature CCC

Figure 3b illustrates the working principle of Feature CCC. The input consists of two feature maps, Feature 1 and Feature 2, extracted from different levels of the UNet architecture. The output is a tensor obtained by cross-concatenating the two feature maps along the channel dimension.

First, Feature 1 and Feature 2 are split into $n = 2^k (0 \leq k \leq 5)$ equal parts along the channel dimension, ensuring each sub-tensor has the same number of channels. Mathematically, we can express this as:

$$\begin{aligned} C_{qi} &= \tfrac{C_q}{n}, 1 \leq i \leq n \\ C_{rj} &= \tfrac{C_r}{n}, 1 \leq j \leq n \end{aligned} \tag{26}$$

Specifically, when $n = 1$, indicating $k = 0$, it represents no split operation is performed on Feature 1 and Feature 2. The aforementioned split process can be represented by the following formula:

$$\begin{aligned} \mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_n &= \text{Split}(\mathbf{Q}) \\ \mathbf{R}_1, \mathbf{R}_2, \cdots, \mathbf{R}_n &= \text{Split}(\mathbf{R}) \end{aligned} \tag{27}$$

where $\mathbf{Q} \in \mathbb{R}^{H \times W \times C_q}$ represents the tensor corresponding to Feature 1, $\mathbf{R} \in \mathbb{R}^{H \times W \times C_r}$ represents the tensor corresponding to Feature 2, $\text{Split}(\cdot)$ represents the tensor split operation, $\mathbf{Q}_i \in \mathbb{R}^{H \times W \times C_{qi}} (1 \leq i \leq n)$ represents the sub-tensors obtained by splitting $\mathbf{Q}$, and $\mathbf{R}_j \in \mathbb{R}^{H \times W \times C_{rj}} (1 \leq j \leq n)$ represents the sub-tensors obtained by splitting $\mathbf{R}$.

The channel number of the output tensor $\mathbf{O}'$ can be produced by inserting a sub-tensor $\mathbf{R}_j$ with C_{rj} channels after a sub-tensor $\mathbf{Q}_i$ with C_{qi} channels obtained from splitting, and then sequentially concatenating them along the channel dimension. The channel number of the output tensor $\mathbf{O}'$ can be determined using the following formula:

$$\begin{aligned} C_{o'} &= \sum_{\substack{i=1 \\ j=1}}^{n} \left(C_{qi} + C_{rj}\right) \\ &= C_{q1} + C_{r1} + \cdots + C_{qn} + C_{rn} \\ &= C_q + C_r \end{aligned} \tag{28}$$

The process of channel cross-concatenation mentioned above can be expressed mathematically as follows:

$$\mathbf{O}' = \text{Concat}(\mathbf{Q}_1, \mathbf{R}_1, \mathbf{Q}_2, \mathbf{R}_2, \cdots, \mathbf{Q}_n, \mathbf{R}_n) \tag{29}$$

where $\text{Concat}(\cdot)$ denotes the operation of channel concatenation, while $\mathbf{O}' \in \mathbb{R}^{H \times W \times C_o}$ signifies the resulting tensor.

In summary, the entire process of channel-wise cross-concatenation between Feature 1 and Feature 2 can be represented by the following equation:

$$\mathbf{O}' = \text{FeaCCC}(\mathbf{Q}, \mathbf{R}) \tag{30}$$

where $\text{FeaCCC}(\cdot)$ represents the operation process of Feature CCC.

There are two main differences between Feature CCC and Input CCC. First, in Feature CCC, the channel numbers of input tensors Feature 1 and Feature 2 can be evenly divided by n, while in Input CCC, the channel number of the input tensor Up-HSI may not be evenly divisible by m. Second, in Feature CCC, the second input tensor Feature 2 needs to be split and inserted into each sub-tensor of Feature 1, while in Input CCC, the second input tensor PAN, having only one channel, cannot be split further. Instead, it is duplicated m times and then inserted into each sub-tensor of Up-HSI.

3.2.3. SSA-Net

In order to further enhance the expression capability of spatial–spectral features in the high-dimensional feature maps of CCC-SSA-UNet, we introduce SSA-Net, which is based on the spatial–spectral attention mechanism. Inspired by the design principles of the DARN network in DHP-DARN [30], SSA-Net incorporates N sequentially stacked Res-SSA blocks to adaptively highlight important spectral and spatial feature information. It is important to note that SSA-Net is positioned between the encoder and decoder in our network architecture, with its input being the feature maps extracted by the encoder and its output being the feature maps to be reconstructed by the decoder. Therefore, SSA-Net omits the front-end feature extraction module and the back-end feature reconstruction module of the DARN network. The schematic diagram of SSA-Net is shown in Figure 4, and it can be formulated as follows:

$$\mathbf{F}_N = f_{\text{Res}-SSAB_N}(\mathbf{F}_{N-1}) = f_{\text{Res}-SSAB_N}(f_{\text{Res}-SSAB_{N-1}}(\cdots f_{\text{Res}-SSAB_1}(\mathbf{F}_0)\cdots)) = f_{SSA-Net}(\mathbf{F}_0) \tag{31}$$

where $\mathbf{F}_0$ represents the input feature map of SSA-Net, corresponding to the encoder output feature maps $\mathbf{E}_1$, $\mathbf{E}_2$, and $\mathbf{E}_3$ of CCC-SSA-UNet in Figure 2. $\mathbf{F}_N$ represents the output feature map of SSA-Net, corresponding to the decoder input feature maps $\mathbf{Q}_1$, $\mathbf{Q}_2$, and $\mathbf{Q}_3$ of CCC-SSA-UNet in Figure 2. $\mathbf{F}_k(1 \leq k \leq N-1)$ represents the intermediate feature map of SSA-Net. $f_{\text{Res}-SSAB_k}(\cdot), 1 \leq k \leq N$ represents the function representation of the i-th Res-SSA block.

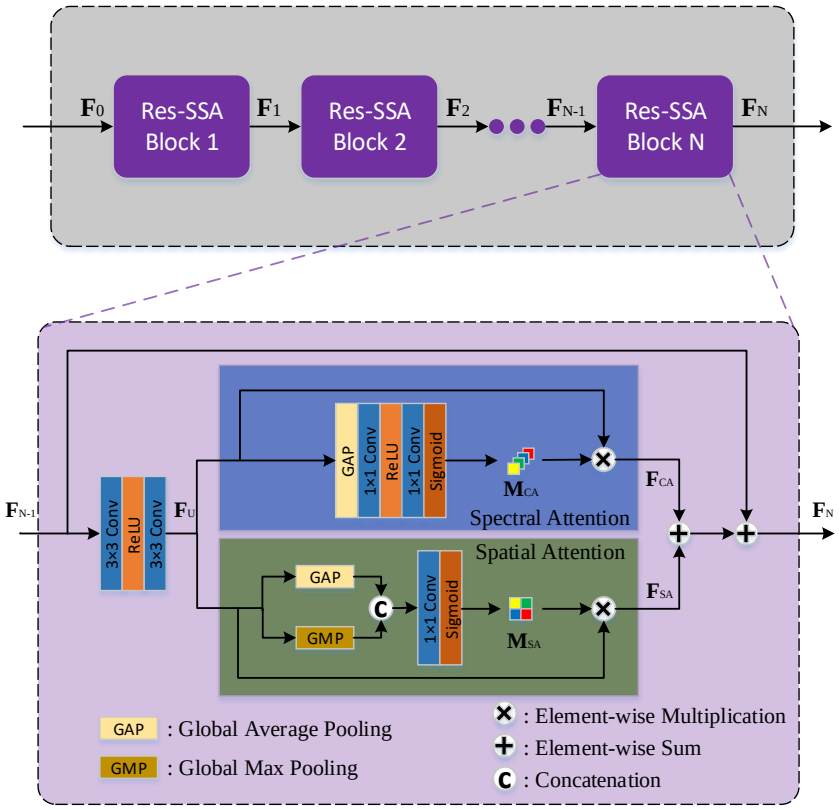


Figure 4. Schematic illustration of the proposed SSA-Net. The SSA-Net consists of N Res-SSA blocks connected sequentially. And "Res-SSA" denotes residual spatial–spectral attention. In the proposed Res-SSA block, spectral attention is parallel to spatial attention and is embedded in the basic residual module.

Similar to the CSA ResBlock in DHP-DARN [30] and the DAU in MIRNet [56], the design principle of our Res-SSA block also incorporates channel attention and spatial attention into the basic residual module. This integration aims to improve both spatial–spectral feature representation and the stability of network training, while accelerating convergence speed. In the field of hyperspectral image processing, channel attention is often referred to as spectral attention. To differentiate it from CSA, we name the entire residual spatial–spectral attention module as the Res-SSA block. The bottom half of Figure 4 illustrates the network structure of the Res-SSA block. For the Nth Res-SSA block, its input is the feature map $\mathbf{F}_{N-1}$. $\mathbf{F}_{N-1}$ goes through a 3×3 convolutional layer to reduce the number of channels to 64. It then undergoes a ReLU activation layer and another 3×3 convolutional layer to extract the feature map $\mathbf{F}_U$, which serves as the input to the attention modules. $\mathbf{F}_U$ is divided into two paths: one path obtains the spectral mask $\mathbf{M}_{CA}$ through the spectral attention module, which is multiplied element-wise with the feature map $\mathbf{F}_U$ to obtain $\mathbf{F}_{CA}$; the other path obtains the spatial mask $\mathbf{M}_{SA}$ through the spatial attention

module, which is multiplied element-wise with the feature map $\mathbf{F}_U$ to obtain $\mathbf{F}_{SA}$. Finally, $\mathbf{F}_{CA}$ and $\mathbf{F}_{SA}$ are added element-wise and combined with the input $\mathbf{F}_{N-1}$ to obtain the output $\mathbf{F}_N$ of the Res-SSA block. This can be expressed mathematically as:

$$\mathbf{F}_U = \mathrm{Conv}_{3\times3}(\delta'(\mathrm{Conv}_{3\times3}(\mathbf{F}_{N-1}))) \tag{32}$$

$$\mathbf{F}_{CA} = \mathbf{F}_U \otimes \mathbf{M}_{CA} \tag{33}$$

$$\mathbf{F}_{SA} = \mathbf{F}_U \otimes \mathbf{M}_{SA} \tag{34}$$

$$\mathbf{F}_N = \mathbf{F}_{CA} + \mathbf{F}_{SA} + \mathbf{F}_{N-1} \tag{35}$$

where $\delta'(\cdot)$ represents the ReLU activation layer, $\mathrm{Conv}_{3\times3}(\cdot)$ represents the 3×3 convolutional layer, $\mathbf{M}_{CA}$ and $\mathbf{M}_{SA}$ represent the spectral mask and spatial mask, respectively, and $\otimes$ represents the element-wise product operation.

Specifically, the backbone of the spectral attention module consists of a global average pooling layer along the spatial dimension, a 1×1 convolutional layer that is employed to reduce the number of channels from 64 to 64/r, a ReLU activation layer, a 1×1 convolutional layer that is employed to expand the number of channels from 64/r to 64, and a sigmoid activation layer in sequence. Here, r is referred to as the channel reduction ratio, which can be used to decrease the computational complexity of the network model. The backbone of the spatial attention module consists of a parallel arrangement of global average pooling and global maxpooling layers, a 1×1 convolutional layer, and a sigmoid activation layer in sequence. This can be expressed mathematically as:

$$\mathbf{M}_{CA} = \sigma(\mathrm{Conv}_{1\times1}(\delta'(\mathrm{Conv}_{1\times1}(\mathrm{GAP}(\mathbf{F}_U))))) \tag{36}$$

$$\mathbf{M}_{SA} = \sigma(\mathrm{Conv}_{1\times1}(\mathrm{Contact}(\mathrm{GAP}(\mathbf{F}_U), \mathrm{GMP}(\mathbf{F}_U)))) \tag{37}$$

where $\sigma(\cdot)$ represents the sigmoid activation layer, $\mathrm{Conv}_{1\times1}(\cdot)$ represents the 1×1 convolutional layer, $\mathrm{GAP}(\cdot)$ stands for Global Average Pooling operation, $\mathrm{GMP}(\cdot)$ stands for Global Maximum Pooling operation, and $\mathrm{Concat}(\cdot)$ represents the operation of channel concatenation.

The spectral attention module (CA) filters out spectral information in the feature tensor that is less important for the fusion results, allowing the network to adaptively select crucial spectral information. The spatial attention module (SA) enables the network to focus more on the features in regions closely related to enhancing spatial details in the hyperspectral image. By combining channel attention with spatial attention and embedding them into the basic residual module, both the spatial–spectral feature representation capability of the network and the stability of network training are enhanced.

3.3. Loss Function

In order to improve the fidelity of the fused reconstruction results to the ground truth HSIs, common loss functions used in the current literature include ℓ_1 loss [30,31,48,57], ℓ_2 loss [58,59], perceptual loss [34], and adversarial loss [60]. Perceptual and adversarial losses are capable of restoring details that do not exist in the original image, which may not be desirable in the field of remote sensing. Conversely, ℓ_1 and ℓ_2 losses are considered more reliable [57]. ℓ_2 loss tends to penalize larger errors while disregarding smaller errors, which can result in networks utilizing ℓ_2 loss producing slightly blurred reconstructions [57,60]. On the other hand, ℓ_1 loss effectively penalizes smaller errors and promotes better convergence during training. Therefore, we employ ℓ_1 loss to evaluate the fusion performance of our network. Specifically, the mean absolute error (MAE) between all reconstructed

images in a training batch and the reference HSIs is used to define the ℓ_1 loss, which can be mathematically expressed using the equation provided below:

$$\mathcal{L}_1(\Theta) = \frac{1}{D}\sum_{d=1}^{D} \|\hat{\mathbf{Y}}_d - \mathbf{Y}_d\|_1 = \frac{1}{D}\sum_{d=1}^{D} \|\Phi(\mathbf{X}_d, \mathbf{P}_d; \Theta) - \mathbf{Y}_d\|_1 \tag{38}$$

where $\hat{\mathbf{Y}}_d$ and $\mathbf{Y}_d$ correspond to the d-th reconstructed HR-HSI and the reference HSI (GT), respectively. D represents the batch size, indicating the number of images included in a training batch. Θ encompasses all the parameters within the network. $\Phi(\cdot, \cdot; \Theta)$ signifies the comprehensive hyperspectral pansharpening neural network model proposed in this paper. Finally, $\| \cdot \|_1$ denotes the utilization of the ℓ_1 norm as a mathematical measure.

4. Experiments and Discussion

4.1. Datasets

In order to assess the effectiveness of the proposed hyperspectral pansharpening algorithm in this study, several hyperspectral image datasets were utilized for the experiments. These datasets include:

- Pavia University Dataset [61]: The Pavia University dataset comprises aerial images acquired over Pavia University in Italy, utilizing the Reflective Optics System Imaging Spectrometer (ROSIS). The original image has a spatial resolution of 1.3 m and dimensions of 610 × 610 pixels. The ROSIS sensor captures 115 spectral bands, covering the spectral range of 430–860 nm. After excluding noisy bands, the image dataset contains 103 spectral bands. To remove uninformative regions, the right-side portion of the image was cropped, leaving a 610 × 340 pixel area for further analysis. Subsequently, a non-overlapping region of 576 × 288 pixels, situated at the top-left corner, was extracted and divided into 18 sub-images measuring 96 × 96 pixels each. These sub-images constitute the reference HR-HSI dataset, serving as the ground truth. To generate corresponding PAN and LR-HSI, the Wald protocol [62] was employed. Specifically, a Gaussian filter with an 8 × 8 kernel size was applied to blur the HR-HSI, followed by a downsampling process, reducing its spatial dimensions by a factor of four to obtain the LR-HSI. The PAN was created by computing the average of the first 100 spectral bands of the HR-HSI. Fourteen image pairs were randomly selected for the training set, while the remaining four pairs were reserved for the test set.
- Pavia Centre Dataset [61]: The Pavia Centre dataset consists of aerial images captured over the city center of Pavia, located in northern Italy, using the Reflective Optics System Imaging Spectrometer (ROSIS). The original image has dimensions of 1096 × 1096 pixels and a spatial resolution of 1.3 m, similar to the Pavia University dataset. After excluding 13 noisy bands, the dataset contains 102 spectral bands, covering the spectral range of 430–860 nm. Due to the lack of informative content in the central region of the image, this portion was cropped, and only the remaining 1096 × 715 pixel area containing the relevant information was used for analysis. Subsequently, a non-overlapping region of 960 × 640 pixels, situated at the top-left corner, was extracted and divided into 24 sub-images measuring 160 × 160 pixels each. These sub-images constitute the reference HR-HSI dataset, serving as the ground truth. Similar to the Pavia University Dataset, the PAN and LR-HSI corresponding to the HR-HSI were generated using the same methodology. Eighteen image pairs were randomly selected as the training set, while the remaining seven pairs were designated as the test set.
- Chikusei Dataset [63]: The Chikusei dataset comprises aerial images captured over the agricultural and urban areas of Chikusei, Japan, in 2014, using the Headwall Hyperspec-VNIR-C sensor. The original image has pixel dimensions of 2517 × 2355 and a spatial resolution of 2.5 m. It encompasses a total of 128 spectral bands, covering the spectral range of 363–1018 nm. For the experiments, a non-overlapping region of

2304 × 2304 pixels was selected from the top-left corner and divided into 81 sub-images of 256 × 256 pixels. These sub-images constitute the reference HR-HSI dataset, serving as the ground truth. Similar to the Pavia University dataset, LR-HSI corresponding to the HR-HSI were generated using the same method. The PAN image was obtained by averaging the spectral bands from 60 to 100 of the HR-HSI. Sixty-one image pairs were randomly selected as the training set, while the remaining 20 pairs were allocated to the test set.

Consistent with previous studies [30,31], the standard deviation (σ) of the Gaussian filter employed for LR-HSI generation is determined through the following formula:

$$\sigma = \sqrt{\frac{\beta^2}{2 \times 2.7725887}} \tag{39}$$

Here, β represents the downsampling scale factor used during the dataset generation process, indicating the linear spatial resolution ratio between the reference HR-HSI and the generated LR-HSI. In this study, the value of β is set to four.

4.2. Evaluation Metrics

To quantitatively assess the performance of our proposed method, we employed five widely used evaluation metrics: correlation coefficient (CC), spectral angle mapping (SAM) [64], root-mean-square error (RMSE), Erreur Relative Globale Adimensionnelle De Synthsès (ERGAS) [65], and peak signal-to-noise ratio (PSNR).

The CC metric provides insight into the geometric distortion present in the images, with values ranging from zero to one. RMSE measures the intensity differences between the super-resolved reconstruction and the ground truth. SAM evaluates the spectral fidelity of the reconstructed images. ERGAS assesses the overall quality of the generated images. PSNR serves as an important indicator of image reconstruction quality, and it is directly related to RMSE. For RMSE, SAM, and ERGAS, lower values indicate superior reconstruction quality. Conversely, higher values of CC and PSNR signify improved image quality, with an ideal value of one for CC and positive infinity for PSNR.

4.3. Implementation Details

The CCC-SSA-UNet proposed in this study comprises a four-level UNet architecture. Each level is characterized by the number of channels, represented as C_{f0}, C_{f1}, C_{f2}, and C_{f2}, ranging from the first to the fourth level, respectively. For the larger model, CCC-SSA-UNet-L, the channel values of C_{f0}, C_{f1}, and C_{f2} are assigned as 32, 64, and 128, respectively. Conversely, for the smaller model, CCC-SSA-UNet-S, the channel values of C_{f0}, C_{f1}, and C_{f2} are set as 32, 32, and 32, respectively. The SSA-Net module within CCC-SSA-UNet consists of ten sequential Res-SSA blocks. Each Res-SSA block employs a channel reduction factor of 16 within its spectral attention module. In the Input CCC module, the input tensor, Up-HSI, is partitioned into eight segments, while in the Feature CCC module, the input tensors, Feature 1 and Feature 2, are equally divided into eight partitions.

We set the batch size to four and employed the Adam [66] optimizer for training, with $\beta 1 = 0.9$ and $\beta 2 = 0.999$ as hyperparameters. The initial learning rate was set to 0.001. Specifically, for the Chikusei dataset, the learning rate was halved every 1000 epochs, whereas for the Pavia University dataset and Pavia Centre dataset, the learning rate was reduced by half every 2000 epochs. To optimize the model, the $\mathcal{L}_1$ loss function was utilized, and a total of 10,500 epochs were conducted. Our model was implemented using the PyTorch framework, and the training process was executed on a single GeForce RTX 3090 GPU. The training duration was approximately 40 h for the Chikusei dataset, 1.5 h for the Pavia University dataset, and 4 h for the Pavia Centre dataset.

4.4. Comparison with State-of-the-Art Methods

In order to demonstrate the effectiveness, efficiency, and state-of-the-art performance of the proposed CCC-SSA-UNet, comparative experiments were conducted on three datasets: Pavia University dataset, Pavia Centre dataset, and Chikusei dataset. Our method was compared against ten traditional pansharpening methods and five state-of-the-art deep learning-based methods. The traditional methods included in the comparison were GS [5], GSA [6], PCA [8], GFPCA [7], BayesNaive [14], BayesSparse [44], MTF-GLP [9], MTF-GLP-HPM [11], CNMF [16], and HySure [12]. The deep learning-based methods consisted of HyperPNN1 [28], HyperPNN2 [28], DHP-DARN [30], DIP-HyperKite [31], and Hyper-DSNet [29]. The traditional methods were implemented using the open-source MATLAB toolbox provided by Loncan et al. [67]. For the deep learning methods, we reproduced the experiments on our computer following the original papers' descriptions and parameter settings, presenting the best results obtained. Notably, all evaluation metrics for the test datasets were recalculated using MATLAB to ensure a fair comparison between traditional and deep learning methods. During the calculations, the reconstructed images were normalized concerning the reference images. The following sections present the detailed comparative experimental results of the different pansharpening methods on the three datasets.

4.4.1. Experiments on Pavia University Dataset

We compared our proposed CCC-SSA-UNet with 15 other methods on the test set of the Pavia University dataset. The quantitative evaluation results are presented in Table 1, with the best values highlighted in red, the second-best values in blue, and the third-best values in green. It is evident that deep learning-based methods outperform traditional methods across various objective metrics. Among the traditional methods, HySure achieves the best performance, with a low SAM of 5.673 and a high PSNR of 32.663. However, there still exists a considerable gap compared to deep learning methods. Among the other five deep learning methods used for comparison, DHP-DARN delivers the best results, possibly due to its utilization of the spatial–spectral dual attention mechanism in the residual blocks. Our proposed CCC-SSA-UNet-S and CCC-SSA-UNet-L benefit from the fusion capability of the CCC operation and the feature extraction capability of the SSA-Net. They significantly outperform all other comparative methods across the objective metrics. Specifically, CCC-SSA-UNet-S improves upon the state-of-the-art comparative methods by 0.002 in CC, 1.675 in RSNR, and 0.786 in PSNR, while reducing SAM by 0.276, RMSE by 0.0014, and ERGAS by 0.176. CCC-SSA-UNet-L achieves improvements of 0.003 in CC, 2.002 in RSNR, and 0.928 in PSNR, along with reductions of 0.321 in SAM, 0.0016 in RMSE, and 0.204 in ERGAS, compared to the state-of-the-art comparative methods.

In addition to the aforementioned quantitative comparison results, we also present the visual results of various pansharpening methods on a randomly selected image patch (10th patch) from the test subset of the Pavia University dataset in Figure 5. To better showcase the reconstruction results, the regions of interest (ROI) in each image are magnified and highlighted with yellow rectangular boxes. Furthermore, in Figure 6, we display the mean absolute error (MAE) maps, which illustrate the differences between the reconstructed images and the reference images for the 10th patch, generated by each method. It can be observed that the images reconstructed by traditional methods are relatively blurry, with larger average absolute errors, resulting in poorer visual quality. In contrast, deep learning-based methods benefit from the powerful learning capabilities of deep neural networks, resulting in less blurriness in the reconstructed images and smaller average absolute errors, indicating better visual quality. Among them, our proposed CCC-SSA-UNet-S and CCC-SSA-UNet-L demonstrate the closest resemblance to the reference images, demonstrating their outstanding ability to maintain spectral fidelity and restore precise spatial details.

Table 1. Quantitative results of different methods on the Pavia University dataset. The best value is marked in red, the second-best value is marked in blue, and the third-best value is marked in green. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better.

Type	Method	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑
Traditional	GS [5]	0.941	6.273	0.0329	37.933	4.755	30.572
	GSA [6]	0.932	6.975	0.0326	38.687	4.745	30.709
	PCA [8]	0.807	9.417	0.0498	29.156	6.977	27.059
	GFPCA [7]	0.855	9.100	0.0516	28.738	7.247	26.754
	BayesNaive [14]	0.869	5.940	0.0443	31.833	6.598	27.662
	BayesSparse [44]	0.892	8.541	0.0428	32.220	6.211	28.210
	MTF-GLP [9]	0.941	6.170	0.0303	39.498	4.273	31.570
	MTF-GLP-HPM [11]	0.917	6.448	0.0348	36.459	5.569	30.401
	CNMF [16]	0.919	6.252	0.0369	35.905	5.356	29.617
	HySure [12]	0.953	5.673	0.0261	42.633	3.809	32.663
Deep learning	HyperPNN1 [28]	0.976	4.117	0.0179	49.903	2.700	35.771
	HyperPNN2 [28]	0.976	4.045	0.0176	50.270	2.663	35.900
	DHP-DARN [30]	0.980	3.793	0.0161	52.015	2.444	36.667
	DIP-HyperKite [31]	0.980	4.127	0.0168	51.126	2.545	36.270
	Hyper-DSNet [29]	0.977	4.038	0.0173	50.618	2.591	36.097
	CCC-SSA-UNet-S (Ours)	0.982	3.517	0.0147	53.690	2.268	37.453
	CCC-SSA-UNet-L (Ours)	0.983	3.472	0.0145	54.017	2.240	37.595

Figure 5. Visual results generated by different pansharpening algorithms for the 10th patch of the Pavia University dataset. (**a**) GS [5], (**b**) GSA [6], (**c**) PCA [8], (**d**) GFPCA [7], (**e**) BayesNaive [14], (**f**) BayesSparse [44], (**g**) MTF-GLP [9], (**h**) MTF-GLP-HPM [11], (**i**) CNMF [16], (**j**) HySure [12], (**k**) HyperPNN1 [28], (**l**) HyperPNN2 [28], (**m**) DHP-DARN [30], (**n**) DIP-HyperKite [31], (**o**) Hyper-DSNet [29], (**p**) CCC-SSA-UNet-S (Ours), (**q**) CCC-SSA-UNet-L (Ours), and (**r**) reference ground truth. The RGB images are generated using the HSI's 60th, 30th, and 10th bands as red, green, and blue bands, respectively. The yellow box indicates the magnified region of interest (ROI).

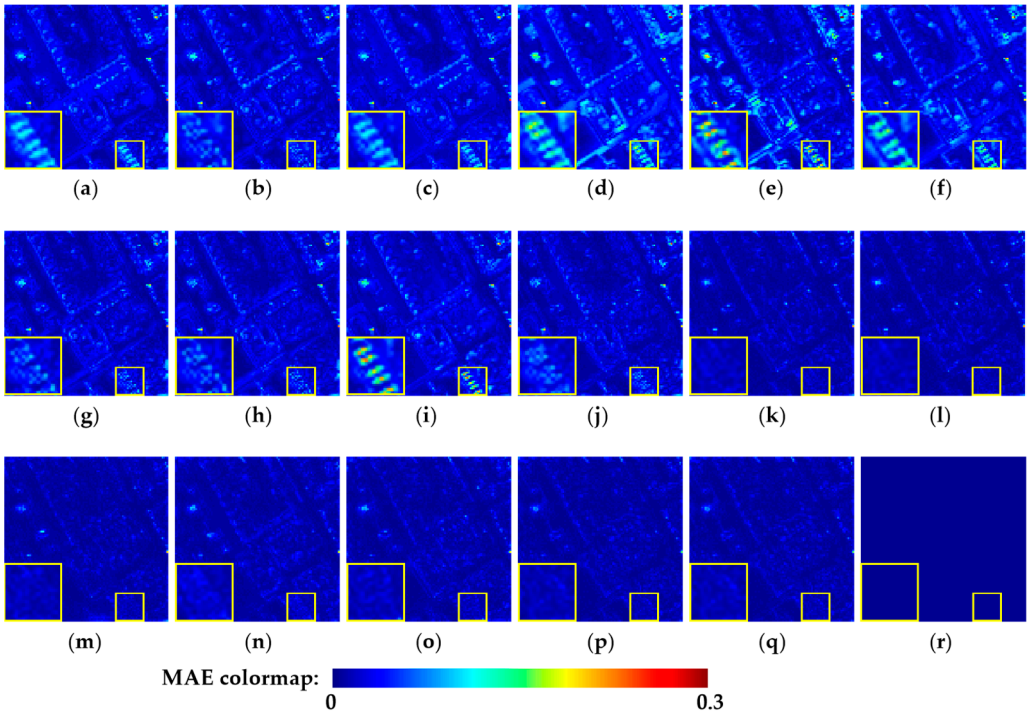


Figure 6. Mean absolute error maps of different pansharpening algorithms for the 10th patch of the Pavia University dataset. (**a**) GS [5], (**b**) GSA [6], (**c**) PCA [8], (**d**) GFPCA [7], (**e**) BayesNaive [14], (**f**) BayesSparse [44], (**g**) MTF-GLP [9], (**h**) MTF-GLP-HPM [11], (**i**) CNMF [16], (**j**) HySure [12], (**k**) HyperPNN1 [28], (**l**) HyperPNN2 [28], (**m**) DHP-DARN [30], (**n**) DIP-HyperKite [31], (**o**) Hyper-DSNet [29], (**p**) CCC-SSA-UNet-S (Ours), (**q**) CCC-SSA-UNet-L (Ours), and (**r**) reference ground truth. MAE colormap denotes the colormap of normalized mean absolute error across all spectral bands; the minimum value is set to 0 and the maximum value is set to 0.3 for better visual comparison. The yellow box indicates the magnified region of interest (ROI).

4.4.2. Experiments on Pavia Centre Dataset

We also compared our proposed CCC-SSA-UNet with 15 other methods on the test set of the Pavia Centre dataset, and the quantitative evaluation results are presented in Table 2. Among the traditional methods, HySure still achieved the best performance, with a SAM as low as 6.723 and a PSNR as high as 34.444, but there is still a significant gap compared to the deep learning methods. Consistent with the results on the Pavia University dataset, the deep learning-based methods outperformed the traditional methods across all objective metrics. Among the other five deep learning methods used for comparison, Hyper-DSNe achieved the best results. This may be attributed to the fact that the Pavia Centre dataset contains more high-frequency information than the Pavia University dataset, and Hyper-DSNet, with its five types of high-pass filter templates serving as multi-detail extractors, is more effective in recovering high-frequency details.

Our proposed CCC-SSA-UNet-S and CCC-SSA-UNet-L benefit from the fusion ability of the CCC and the feature extraction capability of the SSA-Net, demonstrating significant improvements over all other comparison methods in terms of various objective metrics. Specifically, CCC-SSA-UNet-S outperformed the state-of-the-art methods by 0.003 in CC, 1.905 in RSNR, and 0.868 in PSNR, while reducing SAM, RMSE, and ERGAS by 0.284, 0.0013, and 0.190, respectively. Similarly, CCC-SSA-UNet-L achieved improvements of 0.005 in CC, 1.895 in RSNR, and 0.873 in PSNR, along with reductions of 0.295, 0.0026, and 0.194 in SAM, RMSE, and ERGAS, respectively, compared to the state-of-the-art methods.

In addition to quantitative comparison results, Figure 7 presents the visual results of various pan-sharpening methods on randomly selected image patches (the 15th patch) from the test set of the Pavia Centre dataset. Figure 8 displays the mean absolute error

(MAE) maps between the reconstructed images by different methods and the reference images for the 15th patch. Clearly, the images reconstructed by traditional methods appear blurrier, with larger mean absolute errors and lower visual quality. In contrast, the images reconstructed by deep learning-based methods exhibit less blurriness, smaller mean absolute errors, and higher visual quality. Notably, our proposed CCC-SSA-UNet-S and CCC-SSA-UNet-L demonstrate the closest resemblance to the reference images, providing evidence for the effectiveness and advancement of our proposed approach.

Table 2. Quantitative results of different methods on the Pavia Centre dataset. The best value is marked in red, the second-best value is marked in blue, and the third-best value is marked in green. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better.

Type	Method	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑
Traditional	GS [5]	0.964	7.527	0.0281	37.003	4.956	31.694
	GSA [6]	0.955	7.915	0.0263	38.891	4.732	32.236
	PCA [8]	0.946	7.978	0.0324	34.560	5.555	30.917
	GFPCA [7]	0.903	9.463	0.0453	26.940	7.777	27.526
	BayesNaive [14]	0.885	6.964	0.0431	28.292	7.593	27.760
	BayesSparse [44]	0.929	8.908	0.0352	31.999	6.471	29.507
	MTF-GLP [9]	0.960	7.134	0.0248	39.962	4.429	32.852
	MTF-GLP-HPM [11]	0.952	7.585	0.0265	39.033	5.174	32.468
	CNMF [16]	0.948	7.402	0.0293	36.385	5.200	31.287
	HySure [12]	0.971	6.723	0.0208	43.624	3.792	34.444
Deep learning	HyperPNN1 [28]	0.981	5.365	0.0159	49.148	2.990	36.910
	HyperPNN2 [28]	0.981	5.415	0.0161	48.911	3.016	36.814
	DHP-DARN [30]	0.981	6.175	0.0158	49.185	3.038	36.678
	DIP-HyperKite [31]	0.981	6.162	0.0154	49.671	2.975	36.869
	Hyper-DSNet [29]	0.984	4.940	0.0141	51.547	2.680	37.971
	CCC-SSA-UNet-S (Ours)	0.986	4.656	0.0128	53.452	2.490	38.839
	CCC-SSA-UNet-L (Ours)	0.986	4.645	0.0128	53.442	2.486	38.844

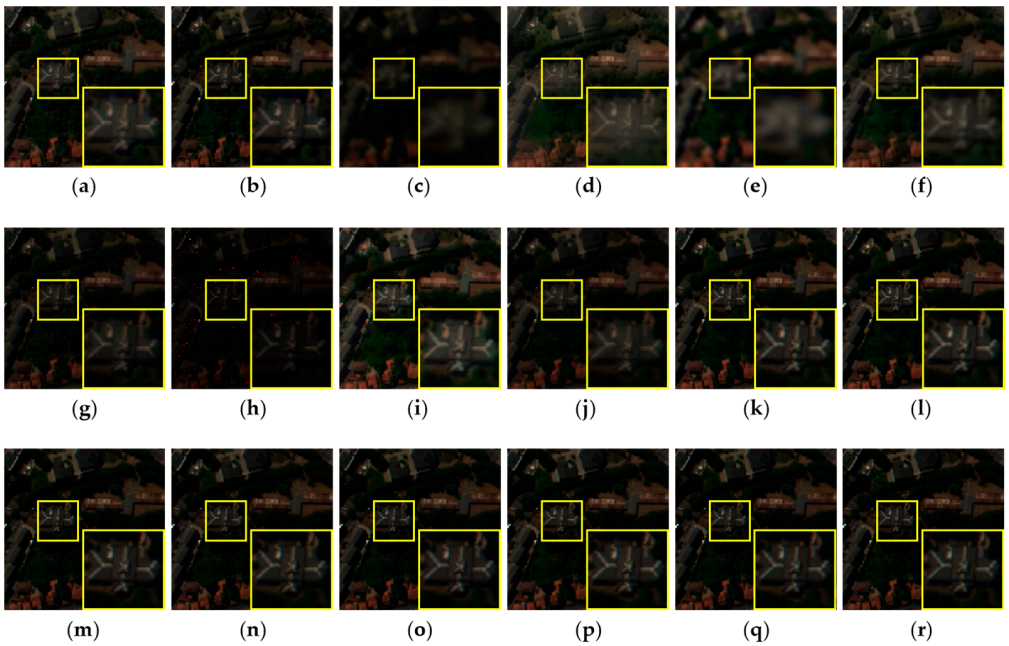


Figure 7. Visual results generated by different pansharpening algorithms for the 15th patch of the Pavia Center dataset. (**a**) GS [5], (**b**) GSA [6], (**c**) PCA [8], (**d**) GFPCA [7], (**e**) BayesNaive [14], (**f**) BayesSparse [44], (**g**) MTF-GLP [9], (**h**) MTF-GLP-HPM [11], (**i**) CNMF [16], (**j**) HySure [12], (**k**) HyperPNN1 [28], (**l**) HyperPNN2 [28], (**m**) DHP-DARN [30], (**n**) DIP-HyperKite [31], (**o**) Hyper-DSNet [29], (**p**) CCC-SSA-UNet-S (Ours), (**q**) CCC-SSA-UNet-L (Ours), and (**r**) reference ground truth. The RGB images are generated using the HSI's 60th, 30th, and 10th bands as red, green, and blue bands, respectively. The yellow box indicates the magnified region of interest (ROI).

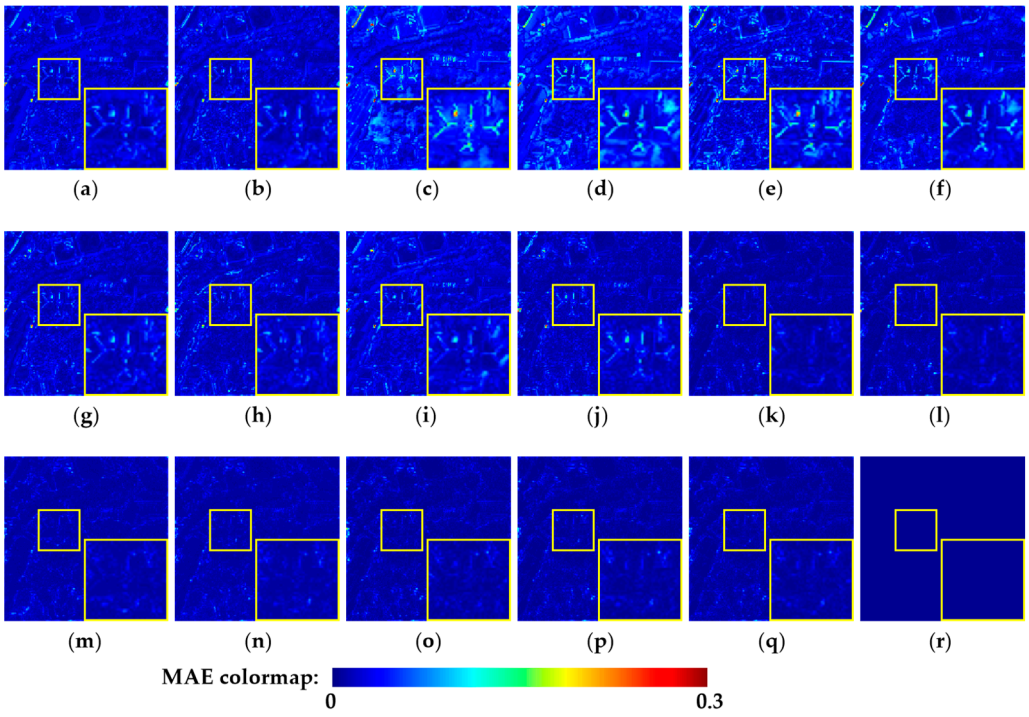


Figure 8. Mean absolute error maps of different pansharpening algorithms for the 15th patch of the Pavia Center dataset. (**a**) GS [5], (**b**) GSA [6], (**c**) PCA [8], (**d**) GFPCA [7], (**e**) BayesNaive [14], (**f**) BayesSparse [44], (**g**) MTF-GLP [9], (**h**) MTF-GLP-HPM [11], (**i**) CNMF [16], (**j**) HySure [12], (**k**) HyperPNN1 [28], (**l**) HyperPNN2 [28], (**m**) DHP-DARN [30], (**n**) DIP-HyperKite [31], (**o**) Hyper-DSNet [29], (**p**) CCC-SSA-UNet-S (Ours), (**q**) CCC-SSA-UNet-L (Ours), and (**r**) reference ground truth. MAE colormap denotes the colormap of normalized mean absolute error across all spectral bands, the minimum value is set to 0 and the maximum value is set to 0.3 for better visual comparison. The yellow box indicates the magnified region of interest (ROI).

4.4.3. Experiments on Chikusei Dataset

We also compared our proposed CCC-SSA-UNet with 15 other methods on the Chikusei dataset. Table 3 presents the average quantitative evaluation results on the test set. Among the traditional methods, HySure achieved the best performance, with a SAM as low as 3.139 and a PSNR as high as 39.615, surpassing the other nine classical methods by a significant margin. Among the five other deep learning methods used for comparison, Hyper-DSNet achieved the best results, consistent with the findings on the Pavia Centre dataset. Our proposed CCC-SSA-UNet-S and CCC-SSA-UNet-L outperformed Hyper-DSNet in terms of SAM, RSNR, and PSNR metrics, while being comparable in terms of CC and RMSE metrics. Specifically, CCC-SSA-UNet-S improved RSNR and PSNR by 0.116 and 0.047, respectively, and reduced SAM by 0.012. CCC-SSA-UNet-L improved RSNR and PSNR by 0.176 and 0.076, respectively, and reduced SAM by 0.011.

In addition to the quantitative comparison results mentioned above, Figure 9 showcases the visual results of various pansharpening methods on the randomly selected 31st image patch from the Chikusei dataset's test set. Figure 10 presents the average absolute error map (MAE map) between the reconstructed images by different methods and the reference images for the 31st patch. It is evident that the images reconstructed by the first nine traditional methods appear blurry with larger average absolute errors, indicating poorer visual quality. Particularly, the reconstructed image by MTF-GLP-HP exhibits numerous red spots, indicating significant spectral distortion. In contrast, HySure and the deep learning-based methods produce less blurry reconstructed images with smaller average absolute errors, demonstrating better visual quality. Among them, our proposed CCC-SSA-UNet-S and CCC-SSA-UNet-L show the closest resemblance to the reference

images and exhibit the lightest colors on the MAE map, further confirming the effectiveness and superiority of our approach.

Table 3. Quantitative results of different methods on the Chikusei dataset. The best value is marked in red, the second-best value is marked in blue, and the third-best value is marked in green. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better.

Type	Method	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑
Traditional	GS [5]	0.942	3.865	0.0176	45.053	5.950	36.334
	GSA [6]	0.947	3.752	0.0152	48.373	5.728	37.467
	PCA [8]	0.793	6.214	0.0343	31.766	9.522	31.524
	GFPCA [7]	0.880	5.237	0.0263	36.843	8.502	32.937
	BayesNaive [14]	0.910	3.367	0.0237	39.235	6.522	34.449
	BayesSparse [44]	0.899	4.840	0.0219	40.396	7.963	34.145
	MTF-GLP [9]	0.938	4.051	0.0157	47.559	6.211	36.994
	MTF-GLP-HPM [11]	0.765	6.322	0.0432	28.782	24.001	31.610
	CNMF [16]	0.901	4.759	0.0208	42.251	7.229	35.224
	HySure [12]	0.962	3.139	0.0117	53.571	4.825	39.615
Deep learning	HyperPNN1 [28]	0.966	2.874	0.0105	55.709	4.458	40.404
	HyperPNN2 [28]	0.967	2.860	0.0105	55.820	4.410	40.464
	DHP-DARN [30]	0.956	3.631	0.0117	53.572	5.029	39.268
	DIP-HyperKite [31]	0.952	3.884	0.0121	52.817	5.324	38.894
	Hyper-DSNet [29]	0.980	2.274	0.0084	60.232	3.460	42.535
	CCC-SSA-UNet-S (Ours)	0.980	2.262	0.0084	60.348	3.492	42.582
	CCC-SSA-UNet-L (Ours)	0.980	2.263	0.0084	60.408	3.478	42.611

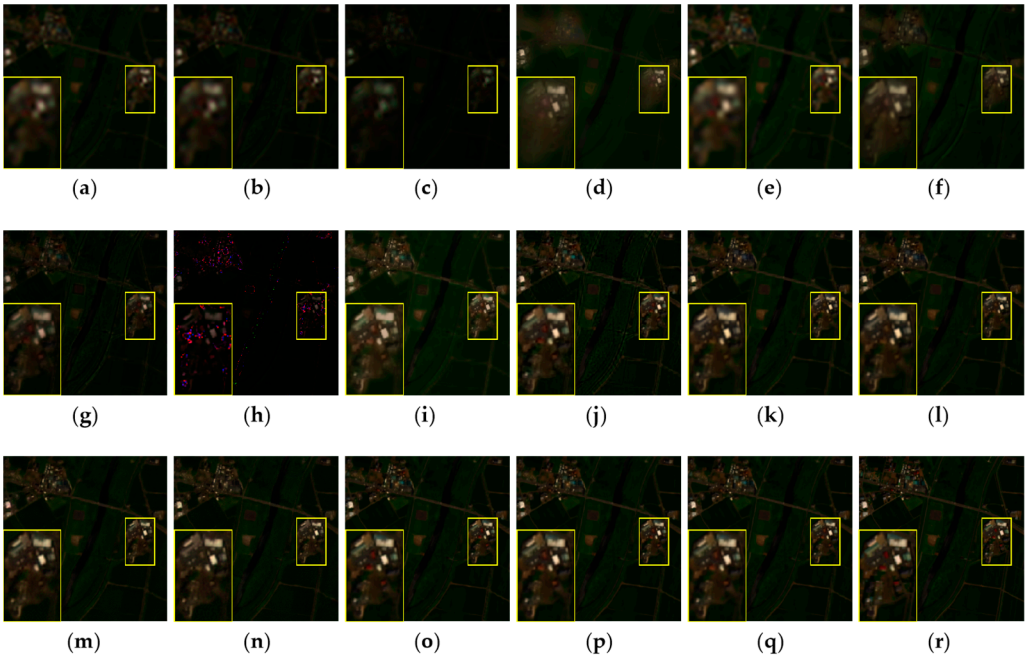


Figure 9. Visual results generated by different pansharpening algorithms for the 31st patch of the Chikusei dataset. (**a**) GS [5], (**b**) GSA [6], (**c**) PCA [8], (**d**) GFPCA [7], (**e**) BayesNaive [14], (**f**) BayesSparse [44], (**g**) MTF-GLP [9], (**h**) MTF-GLP-HPM [11], (**i**) CNMF [16], (**j**) HySure [12], (**k**) HyperPNN1 [28], (**l**) HyperPNN2 [28], (**m**) DHP-DARN [30], (**n**) DIP-HyperKite [31], (**o**) Hyper-DSNet [29], (**p**) CCC-SSA-UNet-S (Ours), (**q**) CCC-SSA-UNet-L (Ours), and (**r**) reference ground truth. The RGB images are generated using the HSI's 61st, 35th, and 10th bands as red, green, and blue bands, respectively. The yellow box indicates the magnified region of interest (ROI).

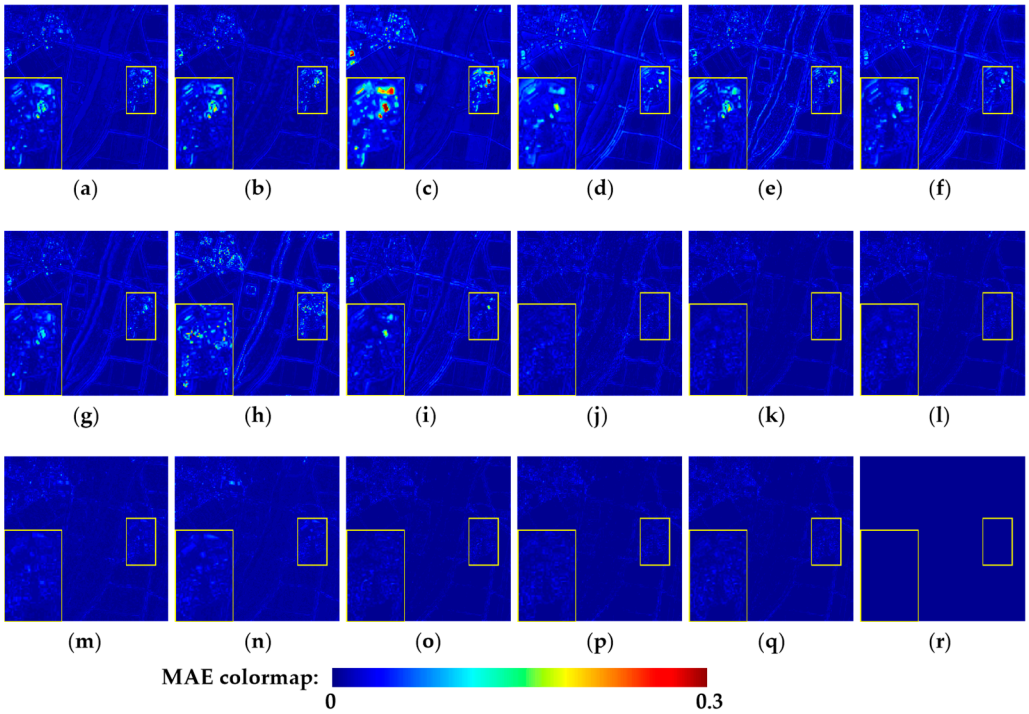


Figure 10. Mean absolute error maps of different pansharpening algorithms for the 31st patch of the Chikusei dataset. (**a**) GS [5], (**b**) GSA [6], (**c**) PCA [8], (**d**) GFPCA [7], (**e**) BayesNaive [14], (**f**) BayesSparse [44], (**g**) MTF-GLP [9], (**h**) MTF-GLP-HPM [11], (**i**) CNMF [16], (**j**) HySure [12], (**k**) HyperPNN1 [28], (**l**) HyperPNN2 [28], (**m**) DHP-DARN [30], (**n**) DIP-HyperKite [31], (**o**) Hyper-DSNet [29], (**p**) CCC-SSA-UNet-S (Ours), (**q**) CCC-SSA-UNet-L (Ours), and (**r**) reference ground truth. MAE colormap denotes the colormap of normalized mean absolute error across all spectral bands, the minimum value is set to 0 and the maximum value is set to 0.3 for better visual comparison. The yellow box indicates the magnified region of interest (ROI).

4.5. Analysis of the Computational Complexity

Table 4 presents a comparison of the computational complexities among different pansharpening methods on the Pavia University dataset. The metrics PSNR and SAM are representative indicators used to evaluate the quality of the network's reconstructed images, while the number of parameters (#Params), multiply accumulate operations (MACs), floating-point operations (FLOPs), GPU memory usage (GPU Memory), and average inference runtime (Runtime) are employed to assess the computational complexity of the neural network. From Table 4, it can be observed that our CCC-SSA-UNet-S achieves leading image reconstruction performance while maintaining a smaller #Params, MACs, FLOPs, and GPU memory usage. In comparison, CCC-SSA-UNet-L demonstrates a five-fold increase in #Params compared to CCC-SSA-UNet-S, while MACs and FLOPs only double. Moreover, GPU memory usage and runtime remain relatively unchanged. Remarkably, CCC-SSA-UNet-L achieves optimal performance in terms of image quality. CCC-SSA-UNet-S and CCC-SSA-UNet-L demonstrate superior performance compared to other methods, as evidenced by extensive experiments conducted on publicly available datasets. Our models effectively leverage multiscale image feature information for fusion reconstruction while maintaining lower memory usage. Furthermore, they achieve shorter inference runtime while ensuring fusion quality.

Table 4. Computational complexity comparison of different pansharpening methods on the Pavia University dataset. The best value is marked in red, and the second-best value is marked in blue. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better. # means the number of something.

Type	Method	PSNR ↑	SAM ↓	#Params (M)	MACs (G)	FLOPs (G)	Memory (G)	Runtime (ms)
Traditional	GS [5]	30.572	6.273	-	-	-	-	79.0
	GSA [6]	30.709	6.975	-	-	-	-	205.8
	PCA [8]	27.059	9.417	-	-	-	-	102.8
	GFPCA [7]	26.754	9.100	-	-	-	-	139.5
	BayesNaive [14]	27.662	5.940	-	-	-	-	136.5
	BayesSparse [44]	28.210	8.541	-	-	-	-	79.5
	MTF-GLP [9]	31.570	6.170	-	-	-	-	30.0
	MTF-GLP-HPM [11]	30.401	6.448	-	-	-	-	118.5
	CNMF [16]	29.617	6.252	-	-	-	-	8976.3
	HySure [12]	32.663	5.673	-	-	-	-	2409.0
Deep learning	HyperPNN1 [28]	35.771	4.117	0.133	1.222	2.444	0.898	24.3
	HyperPNN2 [28]	35.900	4.045	0.137	1.259	2.518	1.124	22.5
	DHP-DARN [30]	36.667	3.793	0.417	3.821	7.642	2.367	176,695.5
	DIP-HyperKite [31]	36.270	4.127	0.526	122.981	245.962	7.082	23,375.5
	Hyper-DSNet [29]	36.097	4.038	0.272	2.490	4.980	1.571	27.5
	CCC-SSA-UNet-S (Ours)	37.453	3.517	0.727	3.259	6.519	1.446	47.8
	CCC-SSA-UNet-L (Ours)	37.595	3.472	4.432	6.323	12.646	1.462	47.8

Figure 11 visually illustrates the balance and superiority of our network in terms of performance and computational complexity. Figure 11a illustrates a comparison of our method with the current state-of-the-art methods in terms of PSNR, FLOPs, and GPU memory usage on the Pavia University dataset. Figure 11b presents a comparison of SAM, FLOPs, and GPU memory usage. It can be observed that our method achieves superior fusion quality with lower GPU memory consumption than the three state-of-the-art deep learning-based pansharpening methods. Figures 11c and 11d, respectively demonstrate the comparisons of PSNR versus runtime and SAM versus runtime for different methods on the test set of the Pavia University dataset. It can be observed that our CCC-SSA-UNet achieves the highest image reconstruction performance while surpassing the majority of existing pansharpening methods in terms of inference runtime. This solidly demonstrates the effectiveness, advancement, and efficiency of our proposed approach.

4.6. Sensitivity Analysis of the Network Parameters

In order to select the optimal network parameters, we conducted extensive experiments and conducted detailed research on the number of Filter Channels, Input CCC groups, Feature CCC groups, SSA blocks, and the choice of initial learning rate. The following sections will provide a detailed description of each aspect.

4.6.1. Analysis of the Filter Channel Numbers

As described in Section 3.2.1, the proposed CCC-SSA-UNet adopts a four-layer UNet architecture with varying numbers of channels in each layer, denoted as C_{f0}, C_{f1}, C_{f2}, and C_{f2}, which can be referred to as Filter Channels. Clearly, the sizes of Filter Channels will have an impact on the model's computational complexity and performance. Therefore, we conducted comparative experiments on the Pavia University dataset to investigate the influence of channel numbers in each layer on the model. As shown in Table 5, while keeping other parameters consistent, we created different models by setting the channel numbers in each layer. Specifically, Model 1 has all channels set to 32, Model 2 has all channels set to 64, Model 3 has all channels set to 128, Model 4 has C_{f0}, C_{f1}, and C_{f2} set to 128, 64, and 32, respectively, and Model 5 has C_{f0}, C_{f1}, and C_{f2} set to 32, 64, and 128, respectively. The experimental results indicate that Model 5 achieved the best performance,

which corresponds to the CCC-SSA-UNet-L model mentioned in Section 4.3. Considering that Model 5 has slightly higher computational complexity, we also selected Model 1, which has comparable performance with the smallest #Params and MACs, as an alternative model. This corresponds to the CCC-SSA-UNet-S model mentioned in Section 4.3.

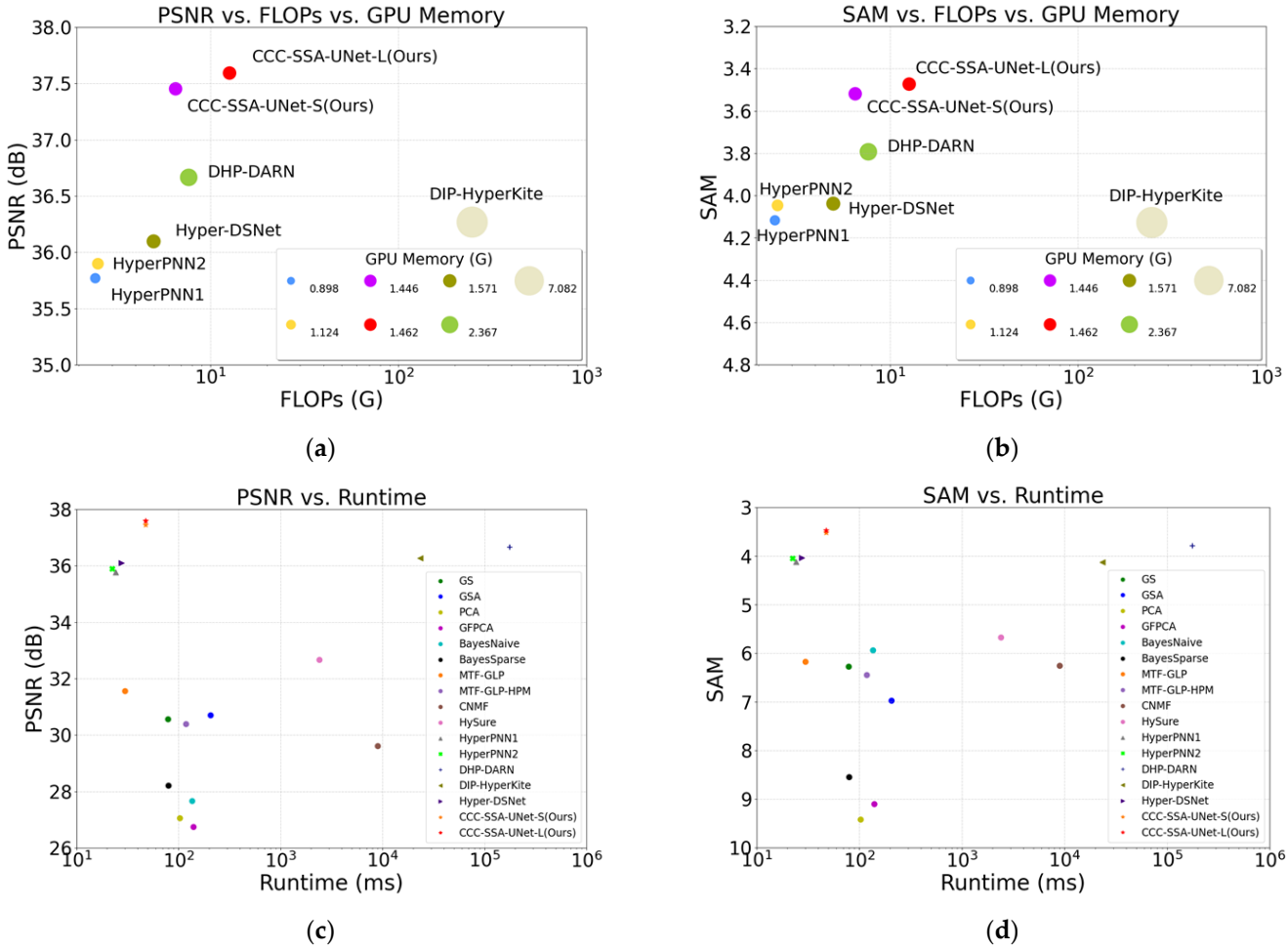


Figure 11. Visual comparison of the computational complexities among different pansharpening methods on the Pavia University dataset. (**a**) Comparison of PSNR, FLOPs, and GPU memory usage. (**b**) Comparison of SAM, FLOPs, and GPU memory usage. (**c**) Comparison of PSNR and Runtime. (**d**) Comparison of SAM and Runtime.

Table 5. Performance comparison of CCC-SSA-UNet with different numbers of filter channels on the Pavia University dataset. The best value is marked in red, the second-best value is marked in blue, and the third-best value is marked in green. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better. # means the number of something.

Model	C_{f0}	C_{f1}	C_{f2}	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑	#Params (M)	MACs (G)	Runtime (ms)
1	32	32	32	0.982	3.517	0.0147	53.690	2.268	37.453	0.727	3.259	47.8
2	64	64	64	0.982	3.528	0.0146	53.850	2.257	37.512	2.686	11.038	48.8
3	128	128	128	0.982	3.527	0.0147	53.709	2.282	37.432	10.331	40.458	54.3
4	128	64	32	0.982	3.495	0.0148	53.697	2.274	37.459	4.568	33.014	54.5
5	32	64	128	0.983	3.472	0.0145	54.017	2.240	37.595	4.432	6.323	47.8

4.6.2. Analysis of the Input CCC Group Numbers

The number of groups, denoted as m, in the Input CCC operation is another hyperparameter of the network. To determine the optimal number of groups, we conducted comparative experiments on the Pavia University dataset to evaluate the impact of the group number on the performance of CCC-SSA-UNet-L. By changing the value of m while keeping other parameters constant, the experimental results are shown in Table 6. It can be observed that when m is set to 8, the network achieves optimal performance in all evaluation metrics. Compared to the base model with m = 1, setting m to 8 only increases the #Params by 0.002 M, MACs by 0.018 G, and Runtime by 0.5 ms. This indicates that the Input CCC operation can effectively enhance the fusion capability of different input source images with minimal increases in parameters and computational complexity.

Table 6. Performance comparison of CCC-SSA-UNet-L with different numbers of Input CCC group on the Pavia University dataset. The best value is marked in **bold**. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better. # means the number of something.

m	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑	#Params (M)	MACs (G)	Runtime (ms)
1	**0.983**	3.492	**0.0145**	53.956	2.252	37.548	**4.430**	**6.305**	**47.3**
2	**0.983**	3.486	0.0146	53.877	2.270	37.490	**4.430**	6.307	**47.3**
4	0.982	3.538	0.0147	53.795	2.261	37.498	4.431	6.312	47.5
8	**0.983**	**3.472**	**0.0145**	**54.017**	**2.240**	**37.595**	4.432	6.323	47.8
12	**0.983**	3.496	0.0146	53.941	2.251	37.555	4.433	6.334	48.0
15	0.982	3.506	**0.0145**	53.967	2.258	37.533	4.434	6.342	48.5
26	0.982	3.577	0.0148	53.621	2.279	37.422	4.437	6.371	49.3
35	0.982	3.491	0.0146	53.885	2.258	37.502	4.440	6.395	50.0

4.6.3. Analysis of the Feature CCC Group Numbers

Similar to the Input CCC operation, the number of groups, denoted as n, in the Feature CCC operation is also one of the network's hyperparameters. We conducted comparative experiments on the Pavia University dataset to evaluate the impact of the group number in Feature CCC on the performance of CCC-SSA-UNet-L. By changing the value of n while keeping other parameters constant, the experimental results are shown in Table 7. It can be observed that when n is set to 8, the network achieves optimal performance in all evaluation metrics. Compared to the base model with n = 1, setting n to 8 does not increase the #Params or MACs. Moreover, the Runtime decreases by 0.2 ms. This indicates that the Feature CCC operation can effectively enhance the fusion capability between different levels of feature maps without adding any parameters or computational complexity.

Table 7. Performance comparison of CCC-SSA-UNet-L with different numbers of Feature CCC group on the Pavia University dataset. The best value is marked in **bold**. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better. # means the number of something.

n	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑	#Params (M)	MACs (G)	Runtime (ms)
1	**0.983**	3.508	0.0147	53.833	2.277	37.479	**4.432**	**6.323**	48.0
2	0.982	3.483	0.0146	53.858	2.268	37.475	**4.432**	**6.323**	48.8
4	0.982	3.497	0.0146	53.874	2.259	37.527	**4.432**	**6.323**	49.8
8	**0.983**	**3.472**	**0.0145**	**54.017**	**2.240**	**37.595**	**4.432**	**6.323**	**47.8**
16	0.982	3.488	**0.0145**	53.989	2.251	37.559	**4.432**	**6.323**	50.0
32	0.982	3.515	0.0146	53.868	2.258	37.516	**4.432**	**6.323**	50.0

4.6.4. Analysis of the SSA Block Numbers

The SSA Block enhances the spatial–spectral feature representation of the network by combining channel attention and spatial attention and embedding them into the basic residual module. Multiple SSA blocks are concatenated to form the SSA-Net. Therefore, the number of SSA blocks, denoted as N, affects the performance of the SSA-Net. To investigate

how the number of SSA blocks influences the network's performance, we constructed several variants of SSA-Net, each containing a different number of SSA blocks while keeping other settings the same. Table 8 presents the comparative experiments conducted on the Pavia University dataset with these 12 SSA-Net variants.

Table 8. Performance comparison of CCC-SSA-UNet-L with different numbers of SSA blocks on the Pavia University dataset. The best value is marked in **bold**. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better. # means the number of something.

N	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑	#Params (M)	MACs (G)	Runtime (ms)
0	0.978	3.964	0.0170	50.938	2.549	36.286	**0.528**	**1.222**	**24.5**
1	0.977	4.059	0.0176	50.349	2.608	36.039	0.918	1.732	26.5
2	0.980	3.779	0.0161	51.988	2.436	36.742	1.309	2.242	27.5
4	0.980	3.724	0.0158	52.386	2.400	36.892	2.089	3.262	32.5
6	0.981	3.679	0.0157	52.520	2.375	36.981	2.870	4.282	38.0
8	0.982	3.511	0.0147	53.807	2.264	37.491	3.651	5.303	45.3
10	**0.983**	3.472	**0.0145**	54.017	**2.240**	**37.595**	4.432	6.323	47.8
12	0.982	3.499	0.0147	53.751	2.264	37.470	5.213	7.343	61.5
14	0.982	3.502	0.0148	53.605	2.290	37.384	5.994	8.364	69.3
16	**0.983**	3.478	**0.0145**	**54.051**	2.246	37.572	6.775	9.384	71.5
18	**0.983**	**3.463**	**0.0145**	53.992	2.245	37.571	7.556	10.404	72.0
20	0.982	3.526	0.0146	53.857	2.270	37.482	8.337	11.425	76.5

The results show that, initially, as *N* increases, the network's performance improves. When *N* reaches 10, the network achieves its maximum performance. However, beyond *N* = 10, as the network deepens, the computational complexity increases significantly without a substantial performance improvement. In fact, the performance may even degrade. Considering both performance and computational complexity, we set the number of SSA blocks to 10 in CCC-SSA-UNet.

4.6.5. Analysis of the Learning Rate

The learning rate is one of the most important hyperparameters in neural networks. Table 9 presents the performance of the CCC-SSA-UNet-L on the Pavia University dataset with different initial learning rates. It can be observed that the network achieves the best results when the initial learning rate is set to 0.001 and undergoes a halving decay every 2000 epochs.

Table 9. Performance comparison of CCC-SSA-UNet-L with different initial learning rates on the Pavia University dataset. The best value is marked in **bold**. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better.

Learning Rate	Decay Rate	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑
0.004	0.5	0.982	3.540	0.0148	53.646	2.302	37.374
0.002	0.5	0.982	3.526	0.0147	53.793	2.263	37.496
0.001	**0.5**	**0.983**	**3.472**	**0.0145**	**54.017**	**2.240**	**37.595**
0.0005	0.5	0.982	3.582	0.0148	53.601	2.294	37.388
0.0001	0.5	0.975	4.441	0.0185	49.411	2.770	35.497

4.7. Ablation Study

In this section, we conducted detailed ablation experiments to validate the effectiveness of the proposed Input CCC, Feature CCC, SSA-Net, and Res-SSA block. We constructed several variants of the CCC-SSA-UNet-L network, labeled as model 1 to model 8, each variant incorporating different combinations of the Input CCC, Feature CCC, and SSA-Net modules. Specifically, model 1 did not use any of the modules and employed regular channel connections instead of Input CCC, and used skip connections instead of SSA-Net.

Models 2 to 4 utilized only one of the modules, while models 5 to 7 excluded one of the modules. Model 8 incorporated all three modules simultaneously. The quantitative results of the ablation experiments conducted on the Pavia University dataset are presented in Table 10. It can be observed that model 8, which incorporates Input CCC, Feature CCC, and SSA-Net, achieved the best performance. A detailed analysis of the effectiveness of each submodule will be discussed in the following subsections.

Table 10. Quantitative results of ablation study of the CCC-SSA-UNet-L on the Pavia University dataset. The best value is marked in **bold**. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better.

Model	Input CCC	Feature CCC	SSA-Net	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑
1	✘	✘	✘	0.977	4.046	0.0175	50.423	2.604	36.064
2	✔	✘	✘	0.978	4.009	0.0171	50.923	2.560	36.271
3	✘	✔	✘	0.978	3.999	0.0172	50.712	2.580	36.157
4	✘	✘	✔	0.982	3.506	0.0147	53.809	2.269	37.490
5	✘	✔	✔	**0.983**	3.492	**0.0145**	53.956	2.252	37.548
6	✔	✘	✔	**0.983**	3.508	0.0147	53.833	2.277	37.479
7	✔	✔	✘	0.978	3.964	0.0170	50.938	2.549	36.286
8	✔	✔	✔	**0.983**	**3.472**	**0.0145**	**54.017**	**2.240**	**37.595**

4.7.1. Effect of the Proposed Input CCC

From Table 10, it is evident that when comparing model 2 to model 1, the inclusion of the Input CCC module in the structure resulted in improvements in the CC, RSNR, and PSNR metrics by 0.001, 0.289, and 0.207, respectively. Additionally, the SAM, RMSE, and ERGAS metrics decreased by 0.037, 0.0004, and 0.044, respectively. Moreover, for model 8 compared to model 5, the addition of the Input CCC module to the structure led to an increase in RSNR and PSNR metrics by 0.061 and 0.047, respectively, while the SAM and ERGAS metrics decreased by 0.020 and 0.012, respectively. These findings strongly demonstrate the effectiveness of the proposed Input CCC method.

4.7.2. Effect of the Proposed Feature CCC

From Table 10, it can be observed that model 3, which incorporates the Feature CCC module into the structure of model 1, showed improvements in the CC, RSNR, and PSNR metrics by 0.001, 0.500, and 0.093, respectively. Moreover, the SAM, RMSE, and ERGAS metrics decreased by 0.047, 0.0003, and 0.024, respectively. On the other hand, model 8, which incorporates the Feature CCC module into the structure of model 6, demonstrated increases in the RSNR and PSNR metrics by 0.184 and 0.116, respectively, while the SAM, RMSE, and ERGAS metrics decreased by 0.036, 0.0002, and 0.037, respectively. These results provide strong evidence for the effectiveness of our proposed Feature CCC method.

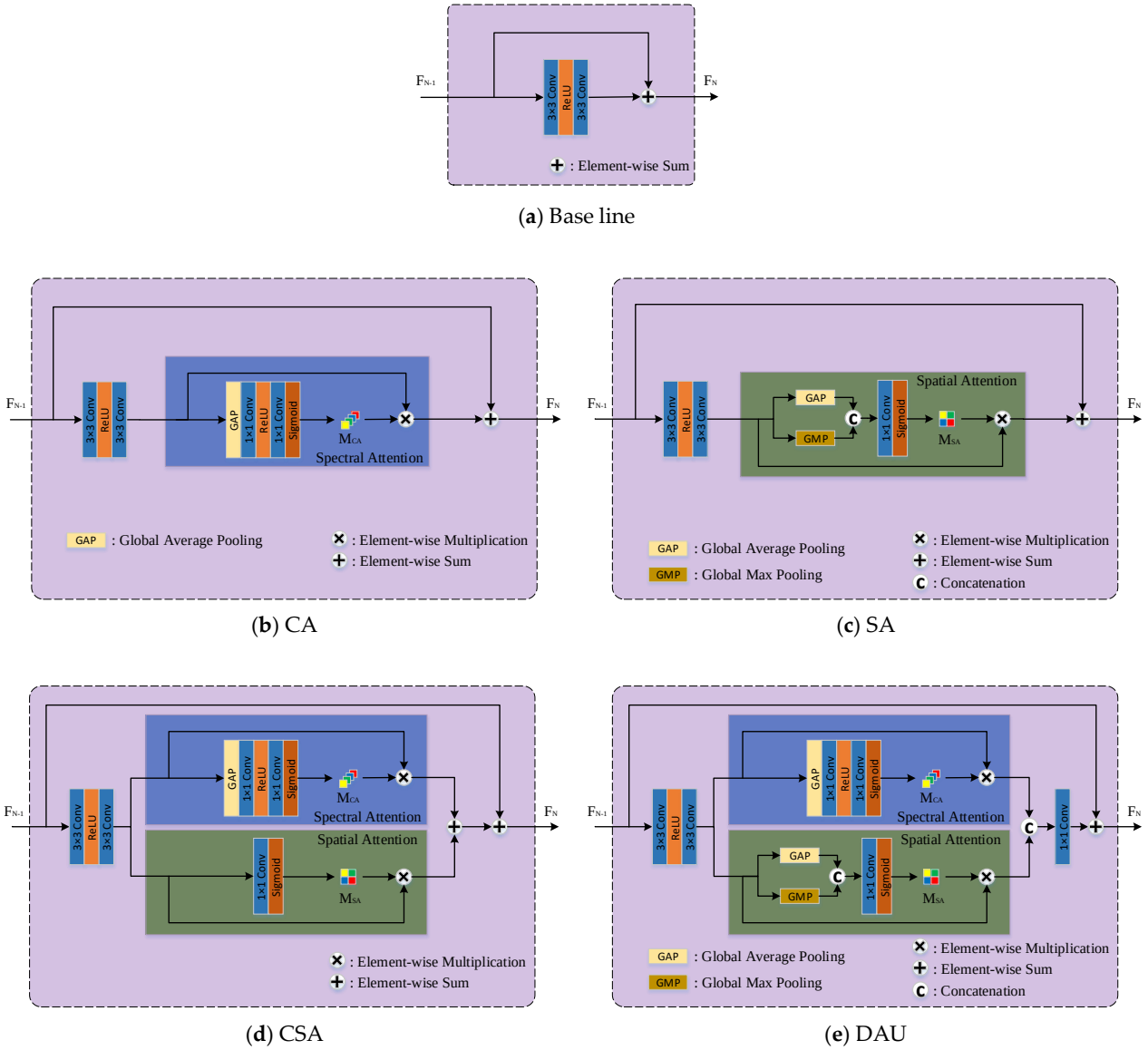
4.7.3. Effect of the Proposed SSA-Net

Table 10 also provides evidence for the effectiveness of our proposed SSA-Net. When SSA-Net was added to the structure of model 1, significant performance improvements were observed. Model 4, which incorporated SSA-Net, exhibited notable improvements in CC, RSNR, and PSNR metrics by 0.005, 3.386, and 1.426, respectively. Additionally, SAM, RMSE, and ERGAS metrics decreased by 0.540, 0.0028, and 0.335, respectively. Similarly, for model 8, which incorporated SSA-Net in the structure of model 7, improvements were observed in CC, RSNR, and PSNR metrics by 0.005, 3.079, and 1.309, respectively. Furthermore, SAM, RMSE, and ERGAS metrics decreased by 0.492, 0.0025, and 0.309, respectively.

4.7.4. Effect of the Proposed Res-SSA Block

SSA-Net is composed of multiple Res-SSA blocks connected in series. To demonstrate the effectiveness of the proposed Res-SSA block, we designed several variants of attention modules for comparison in Figure 12. (a) Residual block baseline, which utilizes the

basic residual module only. (b) CA, which employs the spectral attention module from the Res-SSA block. (c) SA, which utilizes the spatial attention module from the Res-SSA block. (d) CSA, which adopts the channel-spatial attention module from DHP-DARN [30]. (e) DAU, which incorporates the dual attention module from MIRNet [56]. Table 11 presents the quantitative comparison results of CCC-SSA-UNet-L with different attention modules on the Pavia University dataset. It can be observed that using CA or SA alone had a negative impact on the network performance. However, CSA showed performance improvement compared to the baseline residual block, while DAU did not demonstrate significant improvement in network performance. The experimental results indicate that our Res-SSA block achieved the best performance while having fewer parameters and MACs compared to CSA and DAU, suggesting a good balance between image reconstruction performance and computational complexity.



(**a**) Base line

(**b**) CA

(**c**) SA

(**d**) CSA

(**e**) DAU

Figure 12. Schematic illustration of several attention block variants. (**a**) Residual block baseline, (**b**) CA, (**c**) SA, (**d**) CSA, and (**e**) DAU.

Table 11. Performance comparison of CCC-SSA-UNet-L with different attention blocks on the Pavia University dataset. The best value is marked in **bold**. ↑ means that the larger the value, the better, while ↓ means that the smaller the value, the better. # means the number of something.

Attention	CC ↑	SAM ↓	RMSE ↓	RSNR ↑	ERGAS ↓	PSNR ↑	#Params (M)	MACs (G)	Runtime (ms)
Baseline	0.982	3.581	0.0150	53.385	2.299	37.326	**4.403**	**6.318**	**32.8**
CA	0.981	3.652	0.0155	52.722	2.355	37.066	4.432	6.323	41.8
SA	0.980	3.714	0.0159	52.306	2.395	36.891	4.405	6.323	38.0
CSA	0.982	3.561	0.0147	53.710	2.288	37.414	4.434	6.328	44.0
DAU	0.982	3.568	0.0150	53.425	2.303	37.327	4.892	6.895	54.0
Res-SSA	**0.983**	**3.472**	**0.0145**	**54.017**	**2.240**	**37.595**	4.432	6.323	47.8

5. Conclusions

This paper has proposed a novel U-shaped hyperspectral pansharpening network CCC-SSA-UNet for hyperspectral image super-resolution. The channel cross-concatenation mechanism and the spatial–spectral attention mechanism have been incorporated in UNet, which effectively enhanced the network's ability to extract spatial and spectral features. In detail, a novel Input CCC method at the network entrance and a novel Feature CCC method within the decoder have been proposed, which effectively enhanced the fusion capability of different input source images and facilitated the fusion of features at different levels without introducing additional parameters, respectively. Furthermore, the effectiveness of SSA-Net, which is composed of Res-SSA blocks, has been demonstrated by comparing it with other attention module variants. Furthermore, an ablation study has been performed to verify the effectiveness of each module proposed in our framework. By conducting comparative experiments with ten traditional pansharpening methods and five state-of-the-art deep learning-based methods on three datasets, the effectiveness, efficiency, and advancement of our proposed CCC-SSA-UNet have been proven. The results show a satisfactory performance of CCC-SSA-UNet and its superiority over the reference methods.

Although our method has achieved a state-of-the-art performance, there are some limitations and room for improvement too. First, the number of spectral bands varies across different datasets, which affects the equal partitioning of the input tensor Up-HSI in the Input CCC module and limits the flexibility in choosing specific values. In the future, we will consider adding a convolutional layer before the Input CCC module to adjust the channel number of the input tensor. Second, both the encoder and decoder in our network are composed of simple Conv Block modules, which have limited feature extraction capability. In the future, we plan to incorporate lightweight Transformer modules with global self-attention into the encoder and decoder to enhance the network's ability to extract global features. The last problem is the scarcity of training data, which causes a limited generalization ability and poor performance on off-training test images. In the future, we will conduct in-depth research on unsupervised pansharpening methods to promote their application in practical scenarios.

Author Contributions: Conceptualization, Z.L.; validation, Z.L. and G.H.; formal analysis, Z.L.; investigation, Z.L., D.L. and G.H.; data curation, Z.L., A.D. and D.L.; original draft preparation, Z.L.; review and editing, Z.L., H.Y., P.L., D.C., D.L., A.D. and G.H.; Supervision, G.H., H.Y., P.L. and D.C.; funding acquisition, G.H. and D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Development Project of Jilin Province, Key R&D Programs No. 20210201132GX and No. 20210201078GX.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editors and reviewers for their insightful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Audebert, N.; Saux, B.L.; Lefevre, S. Deep Learning for Classification of Hyperspectral Data: A Comparative Review. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 159–173. [CrossRef]
2. Fabelo, H.; Ortega, S.; Ravi, D.; Kiran, B.R.; Sosa, C.; Bulters, D.; Callicó, G.M.; Bulstrode, H.; Szolna, A.; Piñeiro, J.F.; et al. Spatio-spectral classification of hyperspectral images for brain cancer detection during surgical operations. *PLoS ONE* **2018**, *13*, e0193721. [CrossRef] [PubMed]
3. Xie, W.; Lei, J.; Fang, S.; Li, Y.; Jia, X.; Li, M. Dual feature extraction network for hyperspectral image analysis. *Pattern Recognit.* **2021**, *118*, 107992. [CrossRef]
4. Zhang, M.; Sun, X.; Zhu, Q.; Zheng, G. A Survey of Hyperspectral Image Super-Resolution Technology. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 4476–4479.
5. Laben, C.A.; Brower, B.V. Process for Enhancing the Spatial Resolution of Multispectral Imagery Using Pan-Sharpening. U.S. Patent 6,011,875, 4 January 2000.
6. Aiazzi, B.; Baronti, S.; Selva, M. Improving Component Substitution Pansharpening through Multivariate Regression of MS + Pan Data. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 3230–3239. [CrossRef]
7. Liao, W.; Huang, X.; Van Coillie, F.; Gautama, S.; Pižurica, A.; Philips, W.; Liu, H.; Zhu, T.; Shimoni, M.; Moser, G. Processing of multiresolution thermal hyperspectral and digital color data: Outcome of the 2014 IEEE GRSS data fusion contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2984–2996. [CrossRef]
8. Shah, V.P.; Younan, N.H.; King, R.L. An Efficient Pan-Sharpening Method via a Combined Adaptive PCA Approach and Contourlets. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1323–1335. [CrossRef]
9. Aiazzi, B.; Alparone, L.; Baronti, S.; Garzelli, A.; Selva, M. MTF-tailored multiscale fusion of high-resolution MS and Pan imagery. *Photogramm. Eng. Remote Sens.* **2006**, *72*, 591–596. [CrossRef]
10. Liu, J. Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details. *Int. J. Remote Sens.* **2000**, *21*, 3461–3472. [CrossRef]
11. Vivone, G.; Restaino, R.; Mura, M.D.; Licciardi, G.; Chanussot, J. Contrast and Error-Based Fusion Schemes for Multispectral Image Pansharpening. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 930–934. [CrossRef]
12. Simões, M.; Bioucas-Dias, J.; Almeida, L.B.; Chanussot, J. A Convex Formulation for Hyperspectral Image Superresolution via Subspace-Based Regularization. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3373–3388. [CrossRef]
13. Wei, Q.; Bioucas-Dias, J.; Dobigeon, N.; Tourneret, J.Y. Hyperspectral and Multispectral Image Fusion Based on a Sparse Representation. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3658–3668. [CrossRef]
14. Wei, Q.; Dobigeon, N.; Tourneret, J.Y. Bayesian Fusion of Multi-Band Images. *IEEE J. Sel. Top. Signal Process.* **2015**, *9*, 1117–1127. [CrossRef]
15. Kawakami, R.; Matsushita, Y.; Wright, J.; Ben-Ezra, M.; Tai, Y.; Ikeuchi, K. High-resolution hyperspectral imaging via matrix factorization. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 2329–2336.
16. Yokoya, N.; Yairi, T.; Iwasaki, A. Coupled Nonnegative Matrix Factorization Unmixing for Hyperspectral and Multispectral Data Fusion. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 528–537. [CrossRef]
17. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
18. Zhang, Z.; Lu, X.; Cao, G.; Yang, Y.; Jiao, L.; Liu, F. ViT-YOLO: Transformer-Based YOLO for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Montreal, BC, Canada, 11–17 October 2021; pp. 2799–2808.
19. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. SiamAPN++: Siamese Attentional Aggregation Network for Real-Time UAV Tracking. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3086–3092.
20. Zhang, G.; Li, Z.; Li, J.; Hu, X. CFNet: Cascade Fusion Network for Dense Prediction. *arXiv* **2023**, arXiv:2302.06052.
21. Zhang, K.; Li, Y.; Liang, J.; Cao, J.; Zhang, Y.; Tang, H.; Timofte, R.; Van Gool, L. Practical Blind Denoising via Swin-Conv-UNet and Data Synthesis. *arXiv* **2022**, arXiv:2203.13278.
22. Cho, S.-J.; Ji, S.-W.; Hong, J.-P.; Jung, S.-W.; Ko, S.-J. Rethinking coarse-to-fine approach in single image deblurring. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 4641–4650.
23. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Gool, L.V.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 1833–1844.
24. Wang, X.; Xie, L.; Dong, C.; Shan, Y. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1905–1914.
25. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Change Loy, C. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.

26. Yoo, J.; Kim, T.; Lee, S.; Kim, S.H.; Lee, H.; Kim, T.H. Enriched CNN-Transformer Feature Aggregation Networks for Super-Resolution. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 4956–4965.
27. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 286–301.
28. He, L.; Zhu, J.; Li, J.; Plaza, A.; Chanussot, J.; Li, B. HyperPNN: Hyperspectral Pansharpening via Spectrally Predictive Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3092–3100. [CrossRef]
29. Zhuo, Y.W.; Zhang, T.J.; Hu, J.F.; Dou, H.X.; Huang, T.Z.; Deng, L.J. A Deep-Shallow Fusion Network With Multidetail Extractor and Spectral Attention for Hyperspectral Pansharpening. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 7539–7555. [CrossRef]
30. Zheng, Y.; Li, J.; Li, Y.; Guo, J.; Wu, X.; Chanussot, J. Hyperspectral Pansharpening Using Deep Prior and Dual Attention Residual Network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8059–8076. [CrossRef]
31. Bandara, W.G.C.; Valanarasu, J.M.J.; Patel, V.M. Hyperspectral Pansharpening Based on Improved Deep Image Prior and Residual Reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. [CrossRef]
32. Dong, W.; Hou, S.; Xiao, S.; Qu, J.; Du, Q.; Li, Y. Generative Dual-Adversarial Network With Spectral Fidelity and Spatial Enhancement for Hyperspectral Pansharpening. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 7303–7317. [CrossRef]
33. Xie, W.; Cui, Y.; Li, Y.; Lei, J.; Du, Q.; Li, J. HPGAN: Hyperspectral Pansharpening Using 3-D Generative Adversarial Networks. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 463–477. [CrossRef]
34. Bandara, W.G.C.; Patel, V.M. HyperTransformer: A Textural and Spectral Feature Fusion Transformer for Pansharpening. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1767–1777.
35. He, L.; Xi, D.; Li, J.; Lai, H.; Plaza, A.; Chanussot, J. Dynamic Hyperspectral Pansharpening CNNs. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–19. [CrossRef]
36. Luo, F.; Zhou, T.; Liu, J.; Guo, T.; Gong, X.; Ren, J. Multiscale Diff-Changed Feature Fusion Network for Hyperspectral Image Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5502713. [CrossRef]
37. He, X.; Tang, C.; Liu, X.; Zhang, W.; Sun, K.; Xu, J. Object Detection in Hyperspectral Image via Unified Spectral-Spatial Feature Aggregation. *arXiv* **2023**, arXiv:2306.08370.
38. Kordi Ghasrodashti, E. Hyperspectral image classification using a spectral–spatial random walker method. *Int. J. Remote Sens.* **2019**, *40*, 3948–3967. [CrossRef]
39. Chavez, P.; Sides, S.C.; Anderson, J.A. Comparison of three different methods to merge multiresolution and multispectral data-Landsat TM and SPOT panchromatic. *Photogramm. Eng. Remote Sens.* **1991**, *57*, 295–303.
40. Burt, P.J.; Adelson, E.H. The Laplacian pyramid as a compact image code. In *Readings in Computer Vision*; Fischler, M.A., Firschein, O., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1987; pp. 671–679.
41. Aiazzi, B.; Alparone, L.; Baronti, S.; Garzelli, A. Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution analysis. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 2300–2312. [CrossRef]
42. Dong, W.; Zhang, T.; Qu, J.; Xiao, S.; Liang, J.; Li, Y. Laplacian Pyramid Dense Network for Hyperspectral Pansharpening. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5507113. [CrossRef]
43. Fasbender, D.; Radoux, J.; Bogaert, P. Bayesian Data Fusion for Adaptable Image Pansharpening. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1847–1857. [CrossRef]
44. Wei, Q.; Dobigeon, N.; Tourneret, J. Fast Fusion of Multi-Band Images Based on Solving a Sylvester Equation. *IEEE Trans. Image Process.* **2015**, *24*, 4109–4121. [CrossRef]
45. Xue, J.; Zhao, Y.Q.; Bu, Y.; Liao, W.; Chan, J.C.W.; Philips, W. Spatial-Spectral Structured Sparse Low-Rank Representation for Hyperspectral Image Super-Resolution. *IEEE Trans. Image Process.* **2021**, *30*, 3084–3097. [CrossRef] [PubMed]
46. Masi, G.; Cozzolino, D.; Verdoliva, L.; Scarpa, G. Pansharpening by convolutional neural networks. *Remote Sens.* **2016**, *8*, 594. [CrossRef]
47. Yuan, Q.; Wei, Y.; Meng, X.; Shen, H.; Zhang, L. A Multiscale and Multidepth Convolutional Neural Network for Remote Sensing Imagery Pan-Sharpening. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 978–989. [CrossRef]
48. Liu, X.; Liu, Q.; Wang, Y. Remote sensing image fusion based on two-stream fusion network. *Inf. Fusion* **2020**, *55*, 1–15. [CrossRef]
49. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
50. Roy, A.G.; Navab, N.; Wachinger, C. Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Granada, Spain, 16–20 September 2018; pp. 421–429.
51. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.

52. Bass, C.; Silva, M.d.; Sudre, C.; Williams, L.Z.J.; Sousa, H.S.; Tudosiu, P.D.; Alfaro-Almagro, F.; Fitzgibbon, S.P.; Glasser, M.F.; Smith, S.M.; et al. ICAM-Reg: Interpretable Classification and Regression With Feature Attribution for Mapping Neurological Phenotypes in Individual Scans. *IEEE Trans. Med. Imaging* **2023**, *42*, 959–970. [CrossRef]
53. Kordi Ghasrodashti, E.; Sharma, N. Hyperspectral image classification using an extended Auto-Encoder method. *Signal Process. Image Commun.* **2021**, *92*, 116111. [CrossRef]
54. Adkisson, M.; Kimmell, J.C.; Gupta, M.; Abdelsalam, M. Autoencoder-based Anomaly Detection in Smart Farming Ecosystem. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 3390–3399.
55. Zhang, K.; Li, Y.; Zuo, W.; Zhang, L.; Gool, L.V.; Timofte, R. Plug-and-Play Image Restoration With Deep Denoiser Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 6360–6376. [CrossRef]
56. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H.; Shao, L. Learning enriched features for real image restoration and enhancement. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 492–511.
57. Jiang, J.; Sun, H.; Liu, X.; Ma, J. Learning Spatial-Spectral Prior for Super-Resolution of Hyperspectral Imagery. *IEEE Trans. Comput. Imaging* **2020**, *6*, 1082–1096. [CrossRef]
58. Hu, J.F.; Huang, T.Z.; Deng, L.J.; Jiang, T.X.; Vivone, G.; Chanussot, J. Hyperspectral Image Super-Resolution via Deep Spatiospectral Attention Convolutional Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 7251–7265. [CrossRef]
59. He, L.; Zhu, J.; Li, J.; Meng, D.; Chanussot, J.; Plaza, A. Spectral-Fidelity Convolutional Neural Networks for Hyperspectral Pansharpening. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 5898–5914. [CrossRef]
60. Li, J.; Cui, R.; Li, B.; Song, R.; Li, Y.; Dai, Y.; Du, Q. Hyperspectral Image Super-Resolution by Band Attention through Adversarial Learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4304–4318. [CrossRef]
61. Available online: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Pavia_University_scene (accessed on 1 August 2023).
62. Wald, L.; Ranchin, T.; Mangolini, M. Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images. *Photogramm. Eng. Remote Sens.* **1997**, *63*, 691–699.
63. Yokoya, N.; Iwasaki, A. *Airborne Hyperspectral Data over Chikusei*; Technical Report SAL-2016-05-27; The University of Tokyo: Tokyo, Japan, 2016.
64. Yuhas, R.H.; Goetz, A.F.; Boardman, J.W. Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm. In Proceedings of the JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop, Pasadena, CA, USA, 1–5 June 1992; Volume 1.
65. Wald, L. Quality of high resolution synthesised images: Is there a simple criterion? In Proceedings of the Third Conference Fusion of Earth Data: Merging Point Measurements, Raster Maps and Remotely Sensed Images, Sophia Antipolis, France, 26–28 January 2000; pp. 99–103.
66. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
67. Loncan, L.; de Almeida, L.B.; Bioucas-Dias, J.M.; Briottet, X.; Chanussot, J.; Dobigeon, N.; Fabre, S.; Liao, W.; Licciardi, G.A.; Simões, M.; et al. Hyperspectral Pansharpening: A Review. *IEEE Geosci. Remote Sens. Mag.* **2015**, *3*, 27–46. [CrossRef]