



Article Beyond Pixel-Wise Unmixing: Spatial–Spectral Attention Fully Convolutional Networks for Abundance Estimation

Jiaxiang Huang¹ and Puzhao Zhang^{2,3,*}

- Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China; jxhuang@stu.xidian.edu.cn
- ² Key Laboratory of Collaborative Intelligence Systems of Ministry of Education, Xidian University, Xi'an 710071, China
- ³ Division of Geoinformatics, KTH Royal Institute of Technology, 10044 Stockholm, Sweden
- * Correspondence: puzhao@kth.se

Abstract: Spectral unmixing poses a significant challenge within hyperspectral image processing, traditionally addressed by supervised convolutional neural network (CNN)-based approaches employing patch-to-pixel (pixel-wise) methods. However, such pixel-wise methodologies often necessitate image splitting into overlapping patches, resulting in redundant computations and potential information leakage between training and test samples, consequently yielding overoptimistic outcomes. To overcome these challenges, this paper introduces a novel patch-to-patch (patch-wise) framework with nonoverlapping splitting, mitigating the need for repetitive calculations and preventing information leakage. The proposed framework incorporates a novel neural network structure inspired by the fully convolutional network (FCN), tailored for patch-wise unmixing. A highly efficient band reduction layer is incorporated to reduce the spectral dimension, and a specialized abundance constraint module is crafted to enforce both the Abundance Nonnegativity Constraint and the Abundance Sum-to-One Constraint for unmixing tasks. Furthermore, to enhance the performance of abundance estimation, a spatial-spectral attention module is introduced to activate the most informative spatial areas and feature maps. Extensive quantitative experiments and visual assessments conducted on two synthetic datasets and three real datasets substantiate the superior performance of the proposed algorithm. Significantly, the method achieves an impressive RMSE loss of 0.007, which is at least 4.5 times lower than that of other baselines on Urban hyperspectral images. This outcome demonstrates the effectiveness of our approach in addressing the challenges of spectral unmixing.

Keywords: hyperspectral unmixing; abundance estimation; patch-wise unmixing; fully convolutional networks; spatial–spectral attention

1. Introduction

With the significant advancements in hyperspectral camera technology, hyperspectral images can capture a broader spectrum compared to red–green–blue (RGB) images, especially in the nonvisible range of light. The wealth of spectral information in hyperspectral images allows for the identification of materials based on their unique spectral signatures, particularly in scenarios where visual discrimination is challenging. These advantages have led to numerous applications in diverse fields such as mineral exploration, crop health monitoring, urban planning, medical diagnosis, and more [1–12].

Due to the low spatial resolution of imaging instruments and the intricate natural blending of materials in observed scenes, individual pixels in hyperspectral images often contain contributions from multiple materials, each characterized by distinct spectral signatures. This phenomenon is referred to as 'spectral mixture', posing limitations on the broader utilization and hindering the further advancement of hyperspectral imaging [13–16]. Consequently, a crucial and urgent task before the application of hyperspectral images is to address these



Citation: Huang, J.; Zhang, P. Beyond Pixel-Wise Unmixing: Spatial–Spectral Attention Fully Convolutional Networks for Abundance Estimation. *Remote Sens.* 2023, *15*, 5694. https://doi.org/ 10.3390/rs15245694

Academic Editor: Danfeng Hong

Received: 22 October 2023 Revised: 5 December 2023 Accepted: 8 December 2023 Published: 12 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). spectral mixtures in each pixel by separating them into pure spectral signatures (endmembers) and their respective fractional percentages (abundances). This process is known as 'spectral unmixing'.

In the context of hyperspectral unmixing, the primary objectives include estimating three key quantities: the number of endmembers, the spectral signatures of materials (endmembers), and their corresponding abundances. The abundances must adhere to both the Abundance Nonnegativity Constraint (ANC) and the Abundance Sum-to-One Constraint (ASC). The ill-posed nature of spectral unmixing, framed as an inverse problem, has led researchers to explore the application of deep neural networks (DNNs) to address these challenges [17–24]. Unsupervised deep autoencoder (AE) networks, capable of revealing latent structures by reconstructing the original data, have gained popularity in this field [21,22,24–31]. For instance, Ref. [27] utilizes an AE framework to investigate the functions of different blocks of AEs for unmixing, and [28] employs a stacked nonnegative sparse autoencoder to address outliers. However, the low dimensionality of the latent space in AEs, constrained by the number of endmembers, makes it challenging to capture the complete original information of hyperspectral images. To overcome this limitation, Ref. [29] integrates a transformer into a convolutional autoencoder to capture long-range and nonlocal contextual information. Typically, the encoder part of AEs yields abundances, while the weights connecting the last hidden layer and the output layer in the decoder are interpreted as endmembers. Nevertheless, the simple linear encoder that is often adopted struggles to accurately represent the true mixture model, and it is challenging to craft a decoder structure with specific physical meaning that can adapt to various hyperspectral images effectively.

In recent years, various supervised hyperspectral unmixing methods have been proposed, such as pixelCNN, cubeCNN [32], 2D CNN, 3D CNN, and CNNs combining 2D and 3D structures [33], leveraging labeled abundances and demonstrating remarkable performance. While pixelCNN is an exception, most CNN-based unmixing methods take an image patch as input and output a single abundance value for the center pixel of the patch, following a pixel-wise approach. In pixel-wise unmixing, other pixels in the patch are used to assist the unmixing process. However, the presence of heterogeneous areas within the patch may negatively impact performance. Moreover, these methods often split the hyperspectral image into patches with overlap, leading to potential information leakage between the training and test sets [34,35]. This can result in an overestimation of accuracy in predicting abundances, leading to an unfair evaluation.

To address the above concerns, we propose a patch-wise unmixing method. The differences between pixel-wise (i.e., patch-to-pixel) and patch-wise (i.e., patch-to-patch) unmixing are illustrated in Figure 1. Subfigures (a) and (c) in Figure 1 illustrate pixel-wise unmixing on a single patch and a hyperspectral image, respectively, while Subfigures (b) and (d) represent patch-wise unmixing. In Subfigure (a), the pixel-wise method exclusively unmixes the center pixel **K** with the assistance of other pixels in the patch to derive a pixel's abundance. Subsequently, in Subfigure (b), after systematically unmixing all pixels in the image, these pixel abundances are arranged row by row to construct the final abundance map. It is noteworthy that pixel-wise approaches commonly adopt overlapping splitting when decomposing the image into patches. For instance, to unmix pixel **A**, a 3×3 yellow patch is required, while a 3×3 blue patch is necessary for unmixing pixel **N**. However, there are two overlapping pixels in the violet \triangle between the two patches. If one patch is used for training and the other for testing, information leakage may occur from the training sample to the test sample through these overlapping pixels. Furthermore, the overlapping pixels are recomputed multiple times, resulting in an increase in computational burden.

For the patch-wise method, as illustrated in Subfigures (b) and (d), each image patch is unmixed to generate the corresponding abundance patch in a single run. These abundance patches are then combined to create the target abundance. The patch-wise method enables the unmixing of all pixels in a hyperspectral image using nonoverlapping splitting, effectively saving the computation time and also avoiding the risk of information leakage.





(d) Patch-wise unmixing of the image.

Figure 1. Pixel-wise vs. patch-wise unmixing.

To realize the concept of patch-wise unmixing, the key lies in designing an image patch-to-abundance patch network structure. Similar to the semantic segmentation task, hyperspectral unmixing is a per-pixel task of making dense predictions. Both tasks take an image as input and output the classification or abundance of each pixel in the input image. While semantic segmentation is a classification problem, hyperspectral unmixing is typically treated as a regression problem. Semantic segmentation has been a hot topic in computer vision, with various successful methods proposed [36–38]. One of the most popular and impactful methods is the FCN [39]. This raises the question of whether a structure similar to an FCN can effectively address the regression problem of hyperspectral unmixing.

To implement the concept of patch-wise unmixing, inspired from the FCN, we tailored a new patch-wise neural network structure for hyperspectral unmixing. The main contributions of our approach are summarized as follows.

- Beyond the conventional pixel-wise framework commonly employed in CNN unmixing, we introduce a patch-wise unmixing method, facilitating the mapping of image patches to abundance patches. This approach allows for nonoverlapping splitting, eliminating the need to recompute overlapping pixels and mitigating information leakage between the training and test sets, ensuring a fair evaluation.
- A novel convolutional-transposed convolutional structure is meticulously designed. The inclusion of a band reduction convolutional layer effectively reduces the dimensionality of bands, facilitating the extraction of spectral features crucial for accurate unmixing. The fusion of spatial and spectral attention networks enables the model to selectively emphasize informative spatial areas and spectral features, thereby enhancing the performance of abundance estimation. Additionally, a weighted regression loss, combining *RMSE* and *AAD_r*, is proposed to guide the optimization process in hyperspectral unmixing.
- The comparative quantitative experimental results and visual assessments of abundance on two synthetic datasets and three real hyperspectral images validate the superiority of the designed network. Notably, the proposed algorithm significantly outperforms other baselines on synthetic data and Samson data, achieving at least a 4.5-fold improvement in *RMSE* over other baselines on the Urban image.

The remainder of this paper is organized as follows. Section 2 reviews related work on hyperspectral unmixing. Section 3 first defines the hyperspectral unmixing problems including the linear mixture model and the nonlinear mixture model, and then describes the proposed patch-wise unmixing framework in detail. The experiments are given in Section 4. Finally, some concluding remarks are presented in Section 5.

2. Related Work

The hyperspectral mixing problem is commonly addressed using AEs, where the hidden layer of the encoder output signifies abundances, and the decoder's weights connecting this hidden layer to the output represent endmembers. Various AE network structures have been employed for hyperspectral unmixing [23,24,26,30,31,40–54]. Specifically, [25,27,40,55] utilize fully connected layers to construct the autoencoder, while [24,42] leverage CNN structures in both the encoder and decoder to capture spatial information. In [22], an abundance prior and adversarial procedure are integrated into the method to enhance performance and robustness in unmixing. For capturing spectral correlation information in the image, a Long Short-Term Memory network (LSTM) is employed in [31]. To achieve faster and interpretable unmixing, Refs. [48,56] unfold the Iterative Shrinkage-Thresholding Algorithm (ISTA) and Alternating Direction Method of Multipliers (ADMMs) optimization algorithms within the AE framework. However, the single fully connected layer in the decoder of AE methods can only capture linear models, and designing a decoder network based on a specific physics-based mixture model that can be widely applied to various hyperspectral images is often challenging.

Recently, several methods employing deep neural networks were proposed to address the hyperspectral mixing problem in a supervised scenario [32,33,57–65]. Most of these supervised unmixing algorithms utilize CNN structures. For instance, Ref. [32] introduces two versions of CNN: a pixel-based CNN and a cube-based CNN (referred to as CubeCNN) to address unmixing problems. The pixel-based CNN divides hyperspectral images into pixels and performs convolution solely on the spectral dimension using 1×5 and 1×4 convolutional kernels. The cubeCNN, on the other hand, splits the image into 3×3 patches (or treats them as $3 \times 3 \times B$ blocks) and applies $1 \times 1 \times 5$ and $1 \times 1 \times 4$ convolutional operations. Ref. [57] incorporates deep convolutional autoencoders (DCAE) into a supervised scenario by leveraging given endmembers to predict abundances. Two versions of DCAE are proposed: a pixel-based DCAE with a 1 \times 3 kernel and a cube-based DCAE with 3 \times 3 \times 3 and 1 \times 1 \times 3 kernels to determine abundances. Ref. [33] introduces three versions: 2DCUN, 3DCUN, and CrossCUN, using a 2D CNN, a 3D CNN, and a network with a combination of a 3D CNN and a 2D CNN to address hyperspectral unmixing. CrossCUN processes 9 \times 9 patches and applies a 7 \times 7 \times 7, 5 \times 5 \times 3 3D CNN, followed by a 3 \times 3 2D CNN to estimate abundances.

In general, due to the limited availability of hyperspectral image data and the demand for large datasets in deep neural networks, supervised CNN-based unmixing methods resort to splitting the image into overlapping patches to augment the volume of data samples. Subsequently, these methods predict the abundances of the central pixel within the obtained patches by leveraging information from the entire patch. However, this approach involves redundant computations of overlapping pixels, resulting in increased computational overheads. Moreover, the use of overlapping patches may leak information from training samples to test samples, potentially leading to an overestimation of the true accuracy of unmixing. To address these challenges associated with supervised CNNbased unmixers, we propose a patch-wise framework. This framework performs unmixing at the patch level, determining the abundances for all pixels within a patch in a single run to produce an abundance patch, as opposed to only predicting the abundance of the central pixel. In the following section, we introduce our patch-wise framework and present an FCN-inspired network structure that incorporates spatial–spectral attention for hyperspectral unmixing.

3. Methodology

3.1. Preliminary

Consider a hyperspectral image *Y* with *B* bands and *N* pixels (= N_w width × N_h height); each pixel spectrum $y = Y_i$, i = 1, 2, ..., N is mixed in a way as follows:

$$y = \Phi(M, \alpha) + \epsilon, \tag{1}$$

where Φ are the real mixing schemes in natural including linear or nonlinear mixing functions. *M*, α denotes the endmembers and their respective abundances. ϵ represents modeling errors and additive noise. *M* is subject to the Endmember Nonnegativity Constraint (ENC):

$$l \ge 0,$$
 (2)

The abundances satisfy the Abundance Nonnegativity Constraint (ANC) and the Abundance Sum-to-one Constraint (ASC):

N

$$\sum_{k=1}^{P} \alpha_k = 1, \tag{3}$$

$$c_{c} \geq 0,$$
 (4)

where *P* means the number of endmembers.

Each band of pixel spectrum *y* can be represented as $y_j = Y_{ij}$, j = 1, 2, ..., B, where *B* is the number of bands of the spectrum.

 α_{1}

Given a pixel spectrum *y* of a hyperspectral image, the task of unmixing is to find a $\hat{\Phi}^{-1}$ to estimate the endmembers \hat{M} and their respective abundances $\hat{\alpha}$,

$$y \stackrel{\hat{\Phi}^{-1}}{\to} (\hat{M}, \hat{\alpha})$$
 (5)

The hyperspectral unmixing model can be the linear mixture model (LMM) or the nonlinear mixture model (NLMM). The LMM can be formulated as

$$y = M\alpha$$

= $\sum_{k=1}^{p} m_k \alpha_k + \epsilon$ (6)

LMMs are challenging in the case where photons interact with more than two materials. NLMMs take the nonlinearity into consideration to model this situation more precisely. They can be categorized into Additive Nonlinear Models and Post-nonlinear Models [18]. An Additive Nonlinear Model consists of a linear part and a nonlinear part

$$y = M\alpha + \Phi_{nonlinear}(M, \alpha) + \epsilon \tag{7}$$

Post-nonlinear models such as the Hapke model provide a nonlinear transformer on the results of the LMM, which can be written as

$$y = \Phi_{nonlinear}(M\alpha) + \epsilon \tag{8}$$

This means it is hard to find the explicit function to model the inherent mixing mechanism. Deep neural networks make it possible to search the more suitable ones in a wider nonlinear space by learning from data.

3.2. The Patch-Wise Unmixing Framework

We developed a patch-wise framework for hyperspectral unmixing. As depicted in Figure 2, the proposed framework unmixes a hyperspectral image to generate an abundance map through three main stages: Image Padding and Splitting, Patch Unmixing, and Abundance Joining and Cropping.

In stage 1, the hyperspectral image undergoes padding and is split into image patches for unmixing by the Image Padding and Splitting. In stage 2, each patch passes through a Patch Unmixing to produce the corresponding abundance patch. A spatial–spectral attention unmixing network structure, similar to but more shallow than an FCN, is wellsuited for patch unmixing in the Patch Unmixing stage. A band reduction layer is employed to reduce the spectral dimension due to the high dimensionality of bands. To adhere to the ANC and the ASC for the unmixing problem, an abundance constraint module is designed, incorporating a Softplus layer and an ASC layer. Finally, in stage 3, the yielded abundance patches are joined together and cropped into the target abundance map, matching the size of the original hyperspectral image, through the Abundance Joining and Cropping.



Figure 2. The patch-wise unmixing framework, including three stages of Image Padding and Splitting, Patch Unmixing, and Abundance Joining and Cropping. I, B, and P represent the patch size, number of bands, and number of endmembers, respectively.

The main stages are introduced in the subsequent sections, with detailed descriptions of the Image Padding and Splitting and Patch Unmixing followed by the Abundance Joining and Cropping. Additionally, in this section, we introduce a proposed weighted loss aimed at guiding the model search.

3.2.1. Image Padding and Splitting

In the Image Padding and Splitting stage, considering the challenges associated with acquiring hyperspectral image data and the high cost of labeling unannotated data, a common practice is to divide the hyperspectral image into patches. This strategy helps gather sufficient samples for training neural networks in the spectral unmixing domain. In the context of pixel-wise unmixing, to ensure that each pixel of the hyperspectral image is at the center of the corresponding patch, the splitting process commonly results in overlapping patches, posing the risk of information leakage and incurring computational overhead.

To address these concerns, we initially pad the image to a size divisible by the patch size. Subsequently, we split the padded image into nonoverlapping patches. As depicted in Figure 3, a simplified model illustrates the padding and splitting procedures. Assuming a patch size of 2, we conduct padding by adding 1 row at the top and 1 column on the right, i.e., padding on the (left, top, right, bottom) = (0, 1, 1, 0), of the original 3×3 image. The padding values replicate the values of pixels at the edge of the original image, referred to as "edge replicate" padding value mode. Consequently, this process yields a 4×4 padded image with a width divisible by the patch size. Following this, the padded image is split into four 2×2 patches, providing nonoverlapping patches for training, validation, and testing.

Notably, there are three additional padding position modes for the image, specified as (left, top, right, bottom) equal to (0, 0, 1, 1), (1, 0, 0, 1), and (1, 1, 0, 0). Regarding the padding value model, other options are also available; for example, all the padding values

can be set as zeros, 0.5, or ones. In Section 4.3.3, experiments are conducted to assess the impact of these various padding modes on spectral unmixing.



Figure 3. A toy model to illustrate the padding and splitting of the hyperspectral image before the unmixing.

3.2.2. Patch Unmixing

The Patch Unmixing stage serves as the central component of our proposed algorithm. In order to realize our framework, we devised a convolutional-transposed convolution structure inspired by the FCN, incorporating specific layers and modules tailored for patch-wise unmixing. As depicted in Figure 2, our model structure comprises 12 layers due to the small patch size. The foundational unit of the Patch Unmixing includes a bands reduction layer, a convolutional operator, a ReLU activation function, max pooling, transpose convolution, skip connection, and two pivotal modules: the spatial–spectral attention module and the abundance constraint module.

In particular, several key features distinguish our approach: the structure of patch input and patch output facilitates the implementation of a patch-wise framework and enables the unmixing of the image patches in a single run; the inclusion of a bands reduction layer at the beginning of the network for spectral reduction; the integration of a spatial–spectral attention module to capture the most informative regions and channels in feature maps; and the introduction of an abundance constraint module to adhere to the ANC and the ASC for the produced abundances. Additionally, a weighted regression loss is proposed to guide the model optimization process.

A comprehensive listing of the detailed configurations of the proposed model is provided in Table 1. It is noted that only the output sizes are specified for each layer in Table 1. The input size of the current layer is equal to the output size of the previous layer. The first value of 32 represents the batch size. The layers constituting the Patch Unmixing are described in detail as follows:

The input of the Patch Unmixing is one of the image patches generated by the preceding Image Padding and Splitting. The output of the Patch Unmixing is an abundance patch encompassing all abundances for every pixel within the input patch. Considering a hyperspectral image Y with B bands, a width of N_w , and a height of N_h , we create 2D patches (or 3D blocks when considering the band depth) with a window size of $I \times I$ by partitioning the image Y through the Patch Unmixing. As illustrated in Figure 2, the size of the input patch is $B \times I \times I$, and the produced abundance has a size of $P \times I \times I$.

Due to the small input patch, we utilize only three convolutional layers to extract features from hyperspectral patches, mitigating the risk of overfitting. Each convolutional layer employs a (3, 3) kernel, a stride of (1, 1), and padding of (1, 1) to maintain the size identical to the input of the convolution. While Conv1 preserves the number of input feature maps, convolutional layers Conv2 and Conv3 double the number of feature maps compared to their input. The final convolutional layer, Conv4, ensures that the number of output feature maps matches the number of endmembers. Each pooling layer maintains the channel size and reduces the spatial scale by half through max pooling.

ConvTranspos1 and ConvTranspos2 perform the transpose convolution operator to upsample feature maps to twice the size of their input while reducing the number of channels by half. To enhance the fusion of information from shallow layers, the output feature from Conv1 is incorporated into ConvTranspos1 through element-wise addition before ConvTranspos2. Similarly, the feature extracted by Conv2 is integrated into ConvTranspos2 to achieve feature fusion, followed by Conv4 to generate unconstrained abundances.

	Configurations					
BIOCKS	Output Size	Kernel Size	Stride	Padding		
Input	(32, <i>B</i> ^a , I ^b , I)	(3, 3)	(1, 1)	(1, 1)		
Bands reduction	(32, 64, I, I)	(3, 3)	(1, 1)	(1, 1)		
Conv1	(32, 64, I, I)	(3, 3)	(1, 1)	(1, 1)		
Pool1 ^c	(32, 64, I/2, I/2)	(2, 2)	(2, 2)	(0, 0)		
Conv2	(32, 128, I/2, I/2)	(3, 3)	(1, 1)	(1, 1)		
Pool2	(32, 128, I/4, I/4)	(2, 2)	(2, 2)	(0, 0)		
Conv3	(32, 256, I/4, I/4)	(3, 3)	(1, 1)	(1, 1)		
ConvTransp1 ^d	(32, 128, I/2, I/2)	(2, 2)	(2, 2)	(0, 0)		
Replication1 ^e	(32, 128, I/2, I/2)	-	-	-		
ConvTransp2	(32, 64, I, I)	(2, 2)	(2, 2)	(0, 0)		
Spatial-spectral attention module	(32, 64, I, I)	-	-	-		
Replication2	(32, 64, I, I)	-	-	-		
Conv4	(32, P ^f , I, I)	(3, 3)	(1, 1)	(1, 1)		
Abundance constraint module	(32, P, I, I)	-	-	-		
Softplus	(32, P, I, I)	-	-	-		
ASC g	(32, P, I, I)	-	-	-		
MaxPool1 ^h	(32, 64, 1, 1)	-	-	-		
AvgPool1	(32, 64, 1, 1)	-	-	-		
Conv5	(32, 4, 1, 1)	(1, 1)	(0, 0)	(0, 0)		
Conv6	(32, 64, 1, 1)	(1, 1)	(0, 0)	(0, 0)		
Sigmoid1	(32, 64, 1, 1)	-	-	-		
MaxPool2	(32, 1, I, I)	-	-	-		
AvgPool2	(32, 1, I, I)	-	-	-		
Concat	(32, 2, I, I)	-	-	-		
Conv7	(32, 1, I, I)	-	-	-		
Sigmoid2	(32, 1, I, I)	-	-	-		

 Table 1. Configurations of proposed model.

^a *B* denotes the number of spectral bands. ^b I is the patch size. ^c Pool1 and Pool2 represent MaxPooling2d. ^d ConvTransp1 and ConvTransp2 mean ConvTranspose2d. ^e Replication1 is the copy of Conv1's output, and Replication2 is that of Conv2. ^f *P* is the number of endmembers. ^g ASC normalizes the outputs subject to the Abundance Sum-to-one Constraint. ^h MaxPool and AvgPool represent AdaptiveMaxPool2d and AdaptiveAvgPool2d, respectively.

Band Reduction Layer. Hyperspectral images contain richer spectral information compared to natural images, which is a key factor contributing to their widespread application in material identification. However, the broad range of bands in hyperspectral images can pose challenges during processing, often demanding extensive computational resources to capture the spectral information. Therefore, in the preprocessing step of hyperspectral image unmixing, a common practice is to employ a classical dimension reduction method, such as PCA (Principal Component Analysis), to decrease the band dimension. To alleviate the computational load in the subsequent layers, we propose an alternative approach. Instead of PCA, we use a convolutional layer to achieve dimension reduction, transforming the original bands into 64 feature maps while maintaining the spatial size unchanged. Distinguished by its unique function compared to subsequent convolutional layers, this layer is specifically termed the band reduction layer, depicted in green in Figure 2. The experimental comparison between the band reduction layer and PCA is conducted in the Experiment section.

Abundance Constraint Module. The physical interpretation of abundance lies in the proportions of the decomposed endmembers. Therefore, the values representing these proportions should be greater than or equal to zero, and their sum is typically constrained to 1. It is crucial to design the layers of the model in a way that satisfies these conditions,

otherwise, the resulting abundances may lose their physical meanings. The abundance constraint module consists of the Softplus layer followed by the ASC layer. The Softplus layer enforces the ANC on the output of Conv4, while the ASC layer normalizes the output of the Softplus layer to ensure compliance with the ASC.

To impose the ANC on the abundances, the *Softplus* activation is utilized to filter out negative values, written as follows:

$$Softplus(\alpha) = \frac{1}{\beta} log(1 + e^{\beta \alpha})$$
(9)

where α denotes the abundance. β is the parameter that controls the reversion to the linear function, and the default value is 1.

Finally, the ASC layer normalizes each abundance to satisfy the ASC in the following way

$$ASC(\alpha_k) = \frac{\alpha_k}{\sum_{k=1}^{P} \alpha_k}$$
(10)

where α_k represents the abundance proportion of endmember *k*.

1

Spatial–Spectral Attention Module. The fundamental structure of the spatial–spectral attention module is depicted in Figure 4. In the spectral attention network segment, the spatial dimension is initially condensed into 1×1 to derive a channel descriptor through max pooling and average pooling along the spatial dimension [66–68]. Subsequently, an excitation operation is performed using two convolutional layers to obtain the excitation response for each channel. Regarding the spatial attention network, it undergoes max pooling and average pooling along the spectral dimension. The outputs of both operations are concatenated, and after passing through a convolutional and sigmoid layer, the most crucial areas are highlighted. Through the spatial–spectral attention module, informative areas and feature maps are activated and emphasized, while less significant ones are attenuated.



Figure 4. Spatial–spectral attention module. MaxPool1Out just duplicates Conv6's output without other operations, as does AvgPool1Out.

3.2.3. Abundance Joining and Cropping

Finally, to construct the target abundance map, the patch abundances are joined together without overlap and stitched in the inverse procedure of the hyperspectral image splitting. As illustrated in Figure 5, a toy model is plotted to illustrate the joining and cropping procedure after unmixing. Four 2×2 abundance patches are combined to form a 4×4 abundance map. After joining the patch abundances, the abundance map may be larger than the size of the original image or the abundance ground truth, which is in the size of 3×3 . Therefore, 1 row at the top and 1 column on the right of the joined map are

cropped using the inverse procedure of the initial padding, resulting in the production of the target abundance map.



Figure 5. A toy model to illustrate the joining and cropping of the abundance after the unmixing.

3.3. Weighted Loss

As hyperspectral unmixing is usually thought of as a regression problem, in general, the Root Mean Square (RMSE) is used to measure the dissimilarities between the estimated abundances and the abundance ground truth of hyperspectral unmixing.

The *RMSE* \in [0, $\sqrt{2}$] for hyperspectral unmixing is defined as

$$RMSE(\hat{\alpha}, \alpha) = \sqrt{\frac{\sum_{i=1}^{N} (\hat{\alpha}_i - \alpha_i)^2}{N}}$$
(11)

where $\hat{\alpha}$ represents the estimated abundances and α denotes the abundance ground truth. The Root Mean Square of each endmember $(RMSE_e) \in [0, \sqrt{2}]$ between the predicted

abundances and the abundance ground truth for each endmember. It can be written as

$$RMSE_e(\hat{\alpha}_k, \alpha_k) = \sqrt{\frac{\sum_{i=1}^N (\hat{\alpha}_{i,k} - \alpha_{i,k})^2}{N}}$$
(12)

where $\alpha_{i,k}$ represents the abundance ground truth for the *k*-th endmember of pixel *i*.

Abundance Angle Distance (AAD) is the other metric usually used in a hyperspectral unmixing field to measure the distance between the estimated abundance and the real one. Here, two forms of (AAD) are employed, which are defined as follows.

Abundance Angle Distance in RMSE form $(AAD_r) \in [0, \frac{\pi}{2}]$ between the output abundances and abundance ground truth is formulated as

$$AAD_{r}(\hat{\alpha}, \alpha) = \sqrt{\frac{\sum_{i=1}^{N} (\arccos(\frac{\hat{\alpha}_{i}^{T} \alpha_{i}}{\|\hat{\alpha}_{i}\|\|\|\alpha_{i}\|}))^{2}}{N}}$$
(13)

Abundance Angle Distance in average form $(AAD_a) \in [0, \frac{\pi}{2}]$ between the produced abundances and abundance ground truth is defined as

$$AAD_{a}(\hat{\alpha}, \alpha) = \frac{\sum_{i=1}^{N} \arccos(\frac{\hat{\alpha}_{i}^{T} \alpha_{i}}{\|\hat{\alpha}_{i}\| \|\alpha_{i}\|})}{N}$$
(14)

To combine the different merits of the above losses, we proposed a weighted loss *L* of *RMSE* and AAD_r , which is defined as follows

$$L(\hat{\alpha}, \alpha) = (1 - \lambda)RMSE(\hat{\alpha}, \alpha) + \lambda AAD_r(\hat{\alpha}, \alpha)$$
(15)

where λ is the weight used to balance the contribution of *RMSE* and *AAD*_r to loss *L*.

4. Experiments

This section presents the outcomes of extensive experiments, followed by a detailed analysis and discussion of the results. It firstly provides a data description, the adopted baselines, and presents implementation details for the proposed algorithm. Subsequently, a detailed discussion of the results analysis and visual evaluation is provided to compare the evaluation results of the abundances produced by the baseline algorithms with our proposed method. Furthermore, we conducted an ablation study on spatial–spectral attention to verify the effectiveness of the spatial and spectral network components for unmixing. Finally, a parameter sensitivity analysis was carried out to analyze the impact of the parameters in the designed algorithm.

4.1. Experimental setting

This subsection provides a comprehensive overview of the datasets employed in our study, including both synthetic and real datasets. Subsequently, the baseline algorithms utilized for comparison with our proposed method are introduced. Finally, the last part provides detailed information related to the proposed algorithm.

4.1.1. Data Description

To assess the generalization capability of the proposed algorithm, five hyperspectral images are employed, comprising two synthetic datasets, namely Synthetic-noise-free and Synthetic-SNR20dB, and three real hyperspectral images, i.e., Samson, Jasper Ridge, and Urban. The synthetic datasets are generated using the Hyperspectral Imagery Synthesis tools available at http://www.ehu.es/ccwintco/index.php/Hyperspectral_Imagery_Synthesis_tools_for_MATLAB (accessed on 13 May 2023), while the real datasets can be accessed from the Remote Sensing Laboratory at https://rslab.ut.ac.ir/data (accessed on 14 May 2023) or https://github.com/savasozkan/endnet (accessed on 16 May 2023).

(1) Synthetic data

Figure 6 displays hyperspectral images of Synthetic-noise-free and Synthetic-SNR20dB. Both synthetic images are created using five randomly selected endmembers from the USGS spectral library. Each image comprises 128×128 pixels with 431 bands. Notably, Synthetic-noise-free contains no noise, while Synthetic-SNR20dB is generated by introducing additive noise to the Synthetic-noise-free image, achieving a Signal-to-Noise Ratio (SNR) of 20 dB.



Synthetic-noise-free



Synthetic-SNR20dB

Figure 6. Hyperspectral synthetic data of Synthetic-noise-free and Synthetic-SNR20dB.

(2) Real Data

Figure 7 shows the hyperspectral image cubes of Samson, Jasper Ridge, and Urban. The three real hyperspectral image datasets are described in detail as follows.



Samson

Jasper Ridge

Urban

Figure 7. Hyperspectral real images of Samson, Jasper Ridge, and Urban.

Samson is one of the smallest and simplest of the hyperspectral image datasets for spectral unmixing. There are 952×952 pixels in the original Samson hyperspectral image dataset. For each pixel spectrum, there are 156 channels with wavelengths ranging from [401 nm to 889 nm] resulting in a high spectral resolution of 3.13 nm. Since the original Samson is too large to process, in general, the Samson image is cropped to a region of 95×95 , and 9025 pixels are used. The beginning pixel is the 252,332th one. There are three materials latent in the Samson image: #1 *Soil*, #2 *Tree*, and #3 *Water*.

Jasper Ridge is a fashionable hyperspectral dataset for a spectral unmixing study. There are 512×614 pixels in the original Jasper Ridge hyperspectral image dataset. As it is too complex to label the real abundances and endmembers, and it is computationally expensive to analyze the large original image, Jasper Ridge is cropped to a subimage with a size of 100×100 . A total of 10,000 pixels are used, and it begins from the 105, 269-th pixel of the initial image. In regard to each pixel spectrum, there are 224 channels covering wavelengths from 380 nm to 2500 nm with the spectral resolution of 9.46 nm. Owing to atmospheric effects and dense water vapor, the 1–3, 108–112, 154–166, and 220–224th channels are removed, and 198 channels are reserved for hyperspectral unmixing. There are four materials mixed in Jasper Ridge data: #1 *Road*, #2 *Soil*, #3 *Water*, and #4 *Tree*.

Urban is a popular hyperspectral dataset for spectral unmixing analyses. There are 307×307 pixels in the original Urban image. Each of the pixels covers a 2×2 m² region. There are 210 channels with the wavelengths starting from 400 nm to 2500 nm in Urban hyperspectral image data. The spectral resolution is up to 10 nm. Since there are atmospheric effects and dense water vapor in Urban data, the channels 1–4, 76, 87, 101–111, 136–153, and 198–210 are gotten rid of and 162 channels are kept. Three versions of ground truth with, respectively, four, five, and six endmembers are given. Case 1: There are four endmembers latent in the image, i.e., #1 *Asphalt*, #2 *Grass*, #3 *Tree*, #4 *Roof*, and #5 *Dirt*. Case 3: There are six endmembers combined in the image, i.e., #1 *Asphalt*, #2 *Grass*, #3 *Tree*, #4 *Roof*, and #5 *Dirt*. Case 1: There are six endmembers combined in the image, i.e., #1 Asphalt, #2 *Grass*, #3 *Tree*, #4 *Roof*, and #5 *Dirt*. Case 3: There are six endmembers combined in the image, i.e., #1 Asphalt, #2 *Grass*, #3 *Tree*, #4 *Roof*, and #5 *Dirt*. Case 3: There are six endmembers combined in the image, i.e., #1 Asphalt, #2 *Grass*, #3 *Tree*, #4 *Roof*, and #5 *Dirt*. Case 1: There are six endmembers combined in the image, i.e., #1 Asphalt, #2 *Grass*, #3 *Tree*, #4 *Roof*, and #5 *Dirt*. Case 3: There are six endmembers combined in the image, i.e., #1 Asphalt, #2 *Grass*, #3 *Tree*, #4 *Roof*, #5 *Metal*, and #6 *Dirt*. In our experiments, Case 1 with four endmembers is used to verify the effectiveness of the proposed algorithm.

To preprocess these data for hyperspectral unmixing analyses in the same way as [55], the data are normalized to be in the range [0, 1] as follows

$$\tilde{Y} = \frac{Y - \min(Y)}{\max(Y) - \min(Y)}$$
(16)

4.1.2. Baselines

Five hyperspectral unmixing methods are leveraged to be compared with our proposed algorithm to verify its effectiveness, including three unsupervised learning algorithms VAEUN, DAEU, and DeepTrans, and two supervised learning approaches CubeCNN and CrossCUN. A brief introduction of the parameter settings for them are given as follows: VAEUN [25] The original implementation is available at https://github.com/yuanchaosu/ TGRS-daen (accessed on 10 September 2023). However, it encountered issues when applied to our data. Therefore, the variational autoencoder version (VAEUN) is used to evaluate abundance estimation based on the author's recommendation.

DAEU [27] The code is accessible at https://github.com/burknipalsson/hu_autoencoders (accessed on 10 September 2023). The only modification is an increase in the number of epochs from the original 40 to 500 to ensure a fair evaluation.

DeepTrans [29] The code is found at https://github.com/preetam22n/DeepTrans-HSU (accessed on 11 September 2023). Due to the image having to be divided by DeepTrans's patch size 5, the Urban image is cropped to the size of 305×305 . The dim parameter is set as 400 for Urban owing to its large size. The number of epochs is kept the same as ours, which is 500. Other parameters remain unchanged.

CubeCNN [32] The original TensorFlow code for CubeCNN is available at https://web. xidian.edu.cn/xrzhang/paper.html (accessed on 12 September 2023). As it does not work in our environment, we reproduce CubeCNN in PyTorch. For a fair evaluation, the epochs, training set ratio, validation set ratio, and test set ratio are set the same as the proposed method, which are 500, 0.2, 0.1, and 0.7, respectively. Other parameters are configured the same as in the author's original code.

CrossCUN [33] The code is reproduced in PyTorch. To ensure fair evaluation, the epochs, training set ratio, validation set ratio, and test set ratio are set the same as the proposed method, which are 500, 0.2, 0.1, and 0.7, respectively. Other parameters are set the same as in the author's original paper. No batch size is mentioned, and we set it as 32, which is consistent with ours.

4.1.3. Implementation Details

The hyperparameters employed for the proposed algorithm in the experiments are detailed in Table 2. Given that the algorithm is a supervised learning method, all patch samples obtained from hyperspectral image data are split into training, validation, and test sets with respective proportions of 0.2, 0.1, and 0.7.

Hyperparameters	Values	Hyperparameters	Values
Epochs	500	Softplus threshold	1.0
Batch size	32	Patch size	4
Optimizer	Adam	Training set ratio	0.2
Learning rate	0.01	Validation set ratio	0.1
Learning rate scheduler	StepLR	Test set ratio	0.7
Scheduler step size	50	Augment times	5
Scheduler gamma	0.8	Loss weight λ	0.2

Table 2. Hyperparameters for experiments.

In the realm of remote sensing, labeled data are often scarce and annotating images is an expensive endeavor. The limited data in unmixing studies make deep neural networks susceptible to overfitting. To mitigate this challenge, data augmentation is applied to the training set to increase the samples. As illustrated in Figure 8, traditional data augmentations such as flip-upside-down, flip-left-and-right, and rotation (at angles of 90°, 180°, and 270°) are performed on the image patches of training samples. A limited set of three rotation angles is chosen because other angles may lead to distortion of the original image. Increasing the number of augmentations would yield more training samples, potentially leading to better abundance estimation, but it comes at the cost of increased computational resources. In this case, the number of training set samples is augmented up to five times the original size.



Augmentation for Hyperspectral Imag Patch

Figure 8. Augmentation for hyperspectral image patch.

4.2. Results Analysis and Visual Evaluation

To validate the effectiveness of the proposed algorithm, we conducted experiments on two synthetic datasets, namely Synthetic-noise-free and Synthetic-SNR20dB, as well as three real hyperspectral images: Samson, Jasper Ridge, and Urban. Evaluation metrics, including $RMSE_e$, RMSE, AAD_r , and AAD_a , are introduced in Equations (11)–(14), respectively (refer to Section 3.3). The subsequent subsections present quantitative comparisons of abundance results using these metrics and provide visual evaluations of abundance maps for both synthetic and real datasets.

4.2.1. Results of Synthetic Data

Table 3 presents the quantitative abundance results, comparing VAEUN, DAEU, CubeCNN, and CrossCUN with the proposed PFSSA (patch-wise FCN framework with spatial-spectral attention) across metrics including RMSE_e, RMSE, AAD_r, and AAD_a. Here, $RMSE_e$ #1 denotes the evaluation of $RMSE_e$ for the first material. The experiments are conducted on two synthetic datasets: Synthetic-noise-free and Synthetic-SNR20dB with added noise. The loss weight is set as 0.1 due to the occurrence of 'nan' when using a weight of 0.2 on noise-free data. The bold values in the last column indicate that the proposed PFSSA consistently outperforms other algorithms across all eight metrics, affirming the efficacy of the proposed network structure. Especially on Synthetic-noise-free data, VAEUN, the second-best performer, exhibits RMSE, AAD_r, and AAD_a losses 2.2, 1.5, and 2.3 times higher than PFSSA, respectively. On Synthetic-SNR20dB data, CubeCNN, the second-best performer, shows RMSE, AAD_r, and AAD_a losses 1.6, 1.2, and 1.7 times higher than PFSSA, respectively. The losses of PFSSA are 7.4, 6.9, and 10.2, and 5.7, 5.7, and 5.4 times lower than those of the last-placed algorithm DAEU on noise-free data and noise data with SNR = 20 dB, respectively. These demonstrate that the proposed PFSSA exhibits significantly better generalization capabilities compared to the baseline methods. It is also observed that, with the exception of CubeCNN, most algorithms experience higher losses on noise data with SNR = 20 dB compared to noise-free data. This suggests that the addition of noise diminishes the abundance estimation capabilities of most methods.

Ma	teri an		Methods					
Me	trics	VAEUN	DAEU	DeepTrans	CubeCNN	CrossCUN	PFSSA	
RMSE _e #1	noise-free	0.0279	0.1327	0.1027	0.0677	0.1371	0.0184	
	20 dB	0.0433	0.1167	0.0947	0.0548	0.1466	0.0292	
RMSE _e #2	noise-free	0.0205	0.1341	0.0325	0.0667	0.1487	0.0119	
	20 dB	0.0240	0.1242	0.0346	0.0542	0.1483	0.0181	
RMSE _e #3	noise-free	0.0625	0.1966	0.2037	0.0400	0.1562	0.0256	
	20 dB	0.0769	0.2632	0.1977	0.0395	0.1758	0.0364	
RMSE _e #4	noise-free	0.0342	0.1137	0.0337	0.0458	0.1280	0.0135	
	20 dB	0.0276	0.1583	0.0389	0.0429	0.1443	0.0193	
RMSE _e #5	noise-free	0.0538	0.1922	0.1128	0.0470	0.1910	0.0197	
	20 dB	0.0519	0.2456	0.1189	0.0438	0.2007	0.0330	
RMSE	noise-free 20 dB	$0.0428 \\ 0.0486$	0.1576 0.1917	0.1157 0.1139	$0.0548 \\ 0.0476$	0.1561 0.1678	0.0187 0.0283	
AAD _r	noise-free	0.0924	0.4590	0.4159	0.1178	0.2881	0.0580	
	20 dB	0.1476	0.5759	0.4062	0.1070	0.3427	0.0847	
AAD _a	noise-free	0.0791	0.3769	0.2272	0.1127	0.2302	0.0335	
	20 dB	0.1195	0.3737	0.2289	0.1004	0.2532	0.0583	

Table 3. Quantitative comparisons of the abundance results of the Synthetic-noise-free and Synthetic-SNR20dB dataset, where $RMSE_e$ for each material, RMSE, AAD_r , and AAD_a are listed. The best results are marked in bold.

The abundance maps of all baselines and the proposed PFSSA on the Synthetic-noisefree and Synthetic-SNR20dB datasets are illustrated in Figure 9. Each row in the figure represents the abundance maps for each endmember (or material), while each column illustrates the abundance maps for each algorithm. The first column denotes the ground truth of abundance. It is noteworthy that both DAEU and DeepTrans struggle to effectively unmix endmembers # 3 and # 5, whether there is the presence or absence of noise in the data. This limitation may stem from the unsupervised nature of these algorithms, as they lack the guidance provided by the ground truth. Consequently, they may unmix hyperspectral images based solely on their hidden layer representations. As evident in Figure 9, a comparison of endmember # 4 of DAEU in Subfigures (a) and (b) reveals a clearer abundance map for Synthetic-noise-free than for Synthetic-SNR20dB. This observation underscores the negative impact of additive noise on the performance of the DAEU algorithm. For PFSSA, the abundances produced across all endmembers and datasets exhibit better agreement with the ground truth, demonstrating the high performance of the designed framework for unmixing.



Abundance Maps of Synthetic-noise-free Dataset

(a) Synthetic-noise-free dataset



(b) Synthetic-SNR20dB dataset

Figure 9. Abundance maps on Synthetic-noise-free and Synthetic-SNR20dB datasets of ground truth (GT) and methods VAEUN, DAEU, DeepTrans, CubeCNN, CrossCUN, and our proposed PFSSA.

4.2.2. Results of Samson Data

Table 4 presents the abundance quantitative results for the Samson dataset, comparing VAEUN, DAEU, CubeCNN, CrossCUN, and the proposed PFSSA in terms of the $RMSE_e$, RMSE, AAD_r , and AAD_a metrics. The first two methods are AE-based unsupervised learning algorithms, while the subsequent three are CNN-based supervised learning approaches. $RMSE_e$ evaluates the performance on each single endmember, while the other three metrics are computed for the overall abundances of all three endmembers (Soil, Tree, and Water).

Mat		Methods					
wiet	rics	VAEUN	DAEU	DeepTrans	CubeCNN	CrossCUN	PFSSA
	Soil	0.2615	0.0914	0.1773	0.0440	0.1792	0.0183
$RMSE_{e}$	Tree	0.2697	0.0786	0.2007	0.0381	0.1631	0.0148
	Water	0.4027	0.0386	0.3100	0.0236	0.0831	0.0108
RM	SE	0.3180	0.0731	0.2365	0.0363	0.1487	0.0149
AA	D _r	0.7054	0.1530	0.5332	0.0685	0.2722	0.0338
AA	D _a	0.5923	0.1014	0.3940	0.0544	0.1879	0.0197

Table 4. Quantitative comparisons of the abundance results for the Samson dataset, where $RMSE_e$ for each material, *RMSE*, *AAD_r*, and *AAD_a* are listed. The best results are marked in bold.

As shown in Table 4, the proposed PFSSA achieves superior results in estimated abundances, not only for overall endmembers but also for individual materials. VAEUN does not yield satisfactory abundance estimates, likely due to its simple VAE structure. PFSSA and CubeCNN outperform other methods, and CrossCUN surpasses unsupervised VAEUN and DeepTrans, except for DAEU. The inclusion of label information contributes to improved performance. CrossCUN incurs higher losses compared to the other two supervised methods, PFSSA and CubeCNN. This could be attributed to the fact that CrossCUN employs a more complex network structure, involving a 3D CNN followed by a 2D CNN. The increased complexity may make it more susceptible to overfitting, especially on the relatively small patches of hyperspectral images.

Visually, as illustrated in Figure 10, most algorithms effectively decompose the image into endmembers. However, VAEUN and DeepTrans do not produce clear abundance maps compared to the ground truth. For instance, in Subfigures VAEUN, Water and DeepTrans, Water, the right area, which should be Water, is unmixed as Soil or Tree. This discrepancy may arise from the fact that, without the guidance of ground truth, AE-based VAEUN and DeepTrans decompose the image in their own way. In contrast, with regard to PFSSA, the abundances of edges, which may be the more challenging part to unmix, still align well with the GT. This demonstrates the effectiveness of the designed network structure for unmixing.





Figure 10. Abundance maps on Samson dataset of ground truth (GT) and methods VAEUN, DAEU, DeepTrans, CubeCNN, CrossCUN, and our proposed PFSSA.

4.2.3. Results of Jasper Ridge Data

As shown in Table 5, our proposed PFSSA outperforms all AE-based methods and surpasses all CNN-based approaches except CubeCNN. Despite the slightly higher $RMSE_e$ values of PFSSA in Water and Road compared to CubeCNN, PFSSA still outperforms CubeCNN in five out of seven evaluation values. For Jasper Ridge data, most losses of CubeCNN, CrossCUN, and our proposed PFSSA, which utilize label information, are lower than those of the other three unsupervised structures.

Mal	turi ag	Methods					
Ivier	trics	VAEUN	DAEU	DeepTrans	CubeCNN	CrossCUN	PFSSA
	Tree	0.1557	0.1106	0.0828	0.0394	0.2370	0.0245
DMCE	Water	0.2145	0.1638	0.1919	0.0250	0.0764	0.0304
KIVI5Le	Soil	0.1809	0.1632	0.2052	0.0476	0.2359	0.0348
	Road	0.0771	0.2658	0.2863	0.0377	0.1437	0.0391
RN	1SE	0.1650	0.1846	0.2048	0.0385	0.1880	0.0328
AA	ΔD_r	0.4164	0.4527	0.4971	0.0904	0.4275	0.0887
AA	ΔD_a	0.3290	0.3247	0.3370	0.0646	0.2813	0.0402

Table 5. Quantitative comparisons of the abundance results for the Jasper Ridge dataset, where $RMSE_e$ for each material, RMSE, AAD_r , and AAD_a are listed. The best results are marked in bold.

In terms of the visual results of abundances, as shown in Figure 11, for Tree, some detailed regions of CrossCUN do not have a better agreement with the ground truth, although most regions are decomposed well. The unsupervised methods VAEUN, DAEU, and DeepTrans do not yield satisfying visual abundance maps for Water in Jasper Ridge data, as they lack label information to guide the unmixing process. For Soil, VAEUN and DeepTrans have limited capability to unmix this material effectively. VAEUN does not produce a sharp abundance map for Soil, while DeepTrans confuses the material of Soil with Road. For Road, DAEU and DeepTrans do not decompose it well and mix it with the water material. In contrast, PFSSA produces abundance maps that are much more similar to the ground truth across all endmembers, demonstrating the effectiveness of the proposed approach in hyperspectral unmixing.



Abundance Maps of Jasper Ridge

Figure 11. Abundance maps of Jasper Ridge dataset for ground truth (GT) and methods VAEUN, DAEU, DeepTrans, CubeCNN, CrossCUN, and our proposed PFSSA.

4.2.4. Results of Urban Data

As indicated in Table 6, our proposed PFSSA consistently outperforms other algorithms across all evaluation metrics. The $RMSE_e$ values for Asphalt, Grass, Tree, and Roof, as well as the RMSE, AAD_r , and AAD_a metrics for PFSSA are 4.6, 4.7, 5.3, 3.4, 4.5, 4.6, and 6.4 times lower than the second-placed algorithm (CubeCNN), and 56, 40, 44, 28, 43, 48, and 71 times lower than the last-placed algorithm (VAEUN), respectively. This significant improvement may be attributed to the larger size of the Urban image, allowing for more patches and sufficient samples to train the network of PFSSA effectively.

Mal	ui ao	Methods					
Ivier	incs	VAEUN	DAEU	DeepTrans	CubeCNN	CrossCUN	PFSSA
	Asphalt	0.4087	0.2024	0.1583	0.0399	0.2053	0.0071
DMCT	Grass	0.3368	0.1405	0.1860	0.0464	0.2227	0.0081
KIVI5Le	Tree	0.2655	0.0866	0.1300	0.0375	0.1881	0.0059
	Roof	0.1857	0.1167	0.1524	0.0277	0.1295	0.0062
RN	1SE	0.3104	0.1430	0.1579	0.0386	0.1920	0.0070
AA	1D _r	0.8406	0.3234	0.3833	0.0970	0.4319	0.0171
AA	D_a	0.7399	0.2483	0.3610	0.0755	0.3288	0.0102

Table 6. Quantitative comparisons of the abundance results for the Urban dataset, where $RMSE_e$ for each material, RMSE, AAD_r , and AAD_a are listed. The best results are marked in bold.

In the visual results of abundances, shown in Figure 12, most algorithms perform well when unmixing the urban hyperspectral image, except for VAEUN. VAEUN fails to obtain a sharp abundance map for Asphalt, Grass, and Tree. Notably, the Grass region present in the ground truth is absent in VAEUN's Grass abundance map. In contrast, our proposed PFSSA demonstrates a much closer match to the abundance ground truth, confirming the high efficacy of our proposed network structure for hyperspectral unmixing in complex urban data.



Figure 12. Abundance maps of Urban dataset for ground truth (GT) and methods VAEUN, DAEU, DeepTrans, CubeCNN, CrossCUN, and our proposed PFSSA.

4.3. Ablation Study and Parameter Analysis

We start by comparing the band reduction layer with PCA and conducting an ablation study on PFSSA variants with different attention networks. Next, we analyze the influence of various padding models on the abundance evaluation to identify the most suitable one for our dataset. Following that, parameter sensitivity analyses on the training set ratio and the loss weight are performed to determine the optimal values for these parameters in our proposed algorithm. Finally, we evaluate the running time of PFSSA and the baseline algorithms. All experiments are conducted on the validation set over 500 epochs.

4.3.1. Comparing Band Reduction Layer with PCA

We use a band reduction layer to replace PCA, which is typically employed for dimension reduction in a band for hyperspectral unmixing. In this analysis, we examine the impact of the band reduction layer and PCA on abundance estimation. The experiments are conducted on Urban data with a patch size of 8 and 500 epochs. As depicted in Figure 13,

we compare the band reduction layer and PCA based on the RMSE, AADr, AADa, and running time. It is evident that the losses of RMSE, AADr, and AADa with the band reduction layer are significantly lower than those with PCA, with a shorter running time of 22.70 min versus PCA's 76.38 min. This demonstrates the effectiveness and efficiency of the proposed band reduction layer.



Figure 13. PFSSA with PCA vs. PFSSA with band reduction layer based on *RMSE*, *AAD_r*, *AAD_a*, and running time.

4.3.2. Ablation Study on Spatial–Spectral Attention

To assess the effectiveness of our proposed spatial–spectral attention module, we conducted an ablation study on the spatial–spectral attention network with the Samson dataset. As depicted in Figure 14, the proposed PFSSA with both spatial and spectral attention achieves the lowest losses in terms of *RMSE*, *AADr*, and *AADa* for abundance estimation. It is evident that the spatial–spectral attention module is effective and significantly enhances the performance of unmixing. The figure also illustrates that the losses of *RMSE*, *AADr*, and *AADa* for PFSSA with only spectral attention are higher than those of PFSSA without attention. PFSSA with only spatial attention shows a slight improvement compared to PFSSA without attention. However, when both attentions are combined, all losses decrease significantly. This indicates that adding only spectral attention may compromise the performance of abundance estimation in our proposed framework. When combined, the interaction between spatial and spectral attention boosts the abilities of both attentions and results in improved performance.



(c) AADa

Figure 14. Comparison of PFSSA variants based on RMSE, AADr, and AADa.

4.3.3. Padding modes

As listed in Table 7, the abundances of 16 different padding modes are evaluated based on *RMSE_e*, *RMSE*, *AADr*, and *AADa* for the Samson image before splitting it into patches. There are four padding value modes, including three modes using constant values of 0, 0.5, and 1, and one mode that replicates the pixel values at the edge of the image. Each value mode has four padding position models: padding on the (left, top, right, bottom) equal to (0, 0, 1, 1), (0, 1, 1, 0), (1, 0, 0, 1), or (1, 1, 0, 0). The abundance ground truth is padded accordingly. For the constant model, to adhere to the ASC, the padding values of the abundance ground truth are set as $\frac{1}{P}$, where *P* is the number of endmembers. In the case of the Samson data with three endmembers (Soil, Tree, and Water), the padding values of the abundance ground truth are set as $\frac{1}{3}$. For the edge replicate mode, similar to the method for padding the image, the edge abundance values are replicated for padding.

As shown in Table 7, the (0, 1, 1, 0) edge replicate padding model achieves the best abundance estimation compared to the other 15 padding models. Therefore, this model is chosen for padding our Samson hyperspectral image. It is also observed that most of the constant padding models cannot outperform the edge replicate model. This may be because the difference between the pixels with constant values and the padding abundances is larger than that between the pixels with edge replicate values and the padding abundances. The pixels with constant values in the constant model may not align with the distribution of the hyperspectral image, making it harder for the networks to learn compared to the edge replicate model.

		Metrics						
Paddin	Padding Mode		RMSE _e Tree	Water	RMSE	AAD _r	AAD _a	
Constant zeros	(0, 0, 1, 1) (0, 1, 1, 0) (1, 0, 0, 1) (1, 1, 0, 0)	0.0221 0.0206 0.0253 0.0188	0.0197 0.0165 0.0221 0.0163	0.0154 0.0123 0.0150 0.0110	0.0194 0.0168 0.0213 0.0157	0.0447 0.0376 0.0485 0.0349	0.0240 0.0222 0.0269 0.0208	
Constant 0.5	(0, 0, 1, 1) (0, 1, 1, 0) (1, 0, 0, 1) (1, 1, 0, 0)	0.0265 0.0235 0.0231 - ^a	0.0212 0.0178 0.0208	0.0257 0.0152 0.0147	0.0250 0.0194 0.0199 -	0.0557 0.0421 0.0442 -	0.0260 0.0227 0.0238	
Constant ones	(0, 0, 1, 1)(0, 1, 1, 0)(1, 0, 0, 1)(1, 1, 0, 0)	0.0277 0.0352 0.0280 0.0235	0.0286 0.0251 0.0250 0.0249	0.0267 0.0279 0.0187 0.0169	0.0280 0.0301 0.0243 0.0222	0.0631 0.0645 0.0555 0.0498	0.0315 0.0317 0.0293 0.0239	
Edge replicate	(0, 0, 1, 1)(0, 1, 1, 0)(1, 0, 0, 1)(1, 1, 0, 0)	0.0191 0.0183 0.0189 0.0202	0.0159 0.0148 0.0167 0.0167	0.0108 0.0108 0.0119 0.0127	0.0157 0.0149 0.0161 0.0169	0.0360 0.0338 0.0365 0.0381	0.0199 0.0197 0.0210 0.0213	

Table 7. Abundance results of different padding modes on Samson hyperspectral image before splitting, where $RMSE_e$ for each material, RMSE, AAD_r , and AAD_a for all materials are listed. (0, 0, 1, 1) represents padding (0 column on the left, 0 row on the top, 1 column on the right, 1 row on the bottom) of the original hyperspectral image before splitting. The best results are marked in bold.

^a "-" means that the results cannot be obtained owing to the 'nan' appearing during the training.

4.3.4. Training Set Ratio

We also investigated the impact of the training set ratio for the proposed method on the Samson image. In Figure 15, the losses of RMSE, AAD_r , and AAD_a generally decrease, except for the points at 0.3 and 0.6, while the time cost grows with the increasing training ratio. This behavior may be attributed to the small number of patches, leading to some oscillations during training. When choosing the training set ratio, a trade-off needs to be made to balance the estimation loss of abundances and the time cost. Additionally, since labeled data are often scarce in remote sensing, we adopted a train set ratio of 0.2 to train our PFSSA model.



Figure 15. Sensitivity analysis on training set ratio.

4.3.5. Weight of Loss

The evaluation of the abundance results for the Samson image with different weights of loss is depicted in Figure 16. The weight is employed to balance the contribution of *RMSE* and AAD_r to the total loss. Ten weights ranging from 0 to 1.0 with an interval of 0.1 are used for the experiments. A weight of 0 indicates that only *RMSE* loss is used

for backpropagation, while a weight of 1.0 represents that only AAD_r is utilized. It is observed that the point at 0.4 results in high loss values for all three metrics. Other weights achieve close values of abundance evaluation. However, at a weight of 0.2, the three metrics attain the lowest values, and thus the weight of loss is set as 0.2 in our experiments for hyperspectral unmixing.



Figure 16. Sensitivity analysis on loss weight.

4.3.6. Running Time

All the algorithms were executed on the Ubuntu 20.04 LTS platform with a CPU i7-13700k and GPU Nvidia RTX 4090. VAEUN was run on MATLAB R2023b, while the other methods were run on PyCharm 2022. As shown in Table 8, the running times of VAEUN, DAEU, DeepTrans, CubeCNN, CrossCUN, and our proposed PFSSA were compared on the Synthetic-noise-free dataset. DeepTrans is the fastest method, and CrossCUN is the slowest. CubeCNN and CrossCUN each take more than 1.5 h and 3 h, respectively, and this may be attributed to the time-consuming convolutional operation in the spectral dimension. Our proposed PFSSA ranks second and completes the unmixing task within 4 min, demonstrating the efficiency of our designed network structure.

Table 8. Comparison of the proposed PFSSA with other baselines in terms of running time (seconds).The best results are marked in bold.

Methods	VAEUN	DAEU	DeepTrans	CubeCNN	CrossCUN	PFSSA
Time (s)	363.23	484.20	33.21	5973.32	12573.74	213.95

5. Conclusions

In this study, we introduce a novel patch-wise framework that incorporates nonoverlapping splitting to address challenges related to repeated computation and information leakage in pixel-wise methods. Inspired from the FCN, We meticulously design an effective network structure incorporating key layers, including an abundance reduction layer and abundance constraint layers, tailored specifically for spectral unmixing. Furthermore, we integrate a spatial–spectral attention network to bolster the unmixing performance. Our proposed method outperforms other baseline algorithms in abundance evaluation across five out of six datasets with the exception of Jasper Ridge. Even in the Jasper Ridge image, our algorithm excels in five out of seven evaluation metrics, including $RMSE_e$ -Tree, $RMSE_e$ -Soil, RMSE, AAD_r , and AAD_a . In particular, our method achieves a minimum of 3.4 times lower $RMSE_e$ -Roof loss and a maximum of 71 times lower AAD_a loss compared to the baseline algorithms on the Urban image. The quantitative results and visual assessments strongly attest to the efficacy of our proposed algorithm. **Author Contributions:** Conceptualization, J.H.; methodology, J.H.; software, J.H.; validation, J.H. and P.Z.; formal analysis, J.H.; investigation, J.H.; resources, J.H.; data curation, J.H.; writing—original draft preparation, J.H.; writing—review and editing, J.H. and P.Z.; visualization, J.H. and P.Z.; supervision, J.H.; project administration, J.H.; funding acquisition, P.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 62003251.

Data Availability Statement: The source code can be found at https://github.com/JiaxiangHuang/PFSSA.

Acknowledgments: The authors would like to thank Yuanchao Su from XUST (Xi'an University of Science and Technology) for providing the VAEUN code and suggestions for our experiments.

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

References

- 1. Chang, C.I. *Hyperspectral Imaging: Techniques for Spectral Detection and Classification;* Springer Science & Business Media: Berlin, Germany, 2003; Volume 1.
- 2. Landgrebe, D. Hyperspectral image data analysis. IEEE Signal Process. Mag. 2002, 19, 17–28. [CrossRef]
- Li, H.; Lin, Z.; Ma, T.; Zhao, X.; Plaza, A.; William J. Hybrid Fully Connected Tensorized Compression Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* 2023, *61*, 1–16. [CrossRef]
- Li, H.; Hu, W.; Li, W.; Li, J.; Du, Q.; Plaza, A. A³ CLNN: Spatial, Spectral and Multiscale Attention ConvLSTM Neural Network for Multisource Remote Sensing Data Classification. *IEEE Trans. Neural Netw. Learn. Syst.* 2020, 33, 747–761. [CrossRef] [PubMed]
- Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* 2017, 26, 4843–4855. [CrossRef] [PubMed]
- 6. Chang, C.I. Hyperspectral Data Exploitation: Theory and Applications; John Wiley & Sons: Hoboken, NJ, USA, 2007.
- 7. Pistellato, M.; Bergamasco, F.; Torsello, A.; Barbariol, F.; Yoo, J.; Jeong, J.Y.; Benetazzo, A. A physics-driven CNN model for real-time sea waves 3D reconstruction. *Remote Sens.* **2021**, *13*, 3780. [CrossRef]
- 8. Wang, L.; Chang, C.I.; Lee, L.C.; Wang, Y.; Xue, B.; Song, M.; Yu, C.; Li, S. Band subset selection for anomaly detection in hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* 2017, 55, 4887–4898. [CrossRef]
- 9. Plaza, A.; Benediktsson, J.A.; Boardman, J.W.; Brazile, J.; Bruzzone, L.; Camps-Valls, G.; Chanussot, J.; Fauvel, M.; Gamba, P.; Gualtieri, A.; et al. Recent advances in techniques for hyperspectral image processing. *Remote Sens. Environ.* **2009**, *113*, S110–S122. [CrossRef]
- 10. Zhu, M.; Jiao, L.; Liu, F.; Yang, S.; Wang, J. Residual spectral–spatial attention network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 449–462. [CrossRef]
- Pistellato, M.; Traviglia, A.; Bergamasco, F. Geolocating time: Digitisation and reverse engineering of a roman sundial. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2 2020; pp. 143–158.
- 12. Gong, M.; Zhang, M.; Yuan, Y. Unsupervised band selection based on evolutionary multiobjective optimization for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* 2015, 54, 544–557. [CrossRef]
- 13. Ghamisi, P.; Yokoya, N.; Li, J.; Liao, W.; Liu, S.; Plaza, J.; Rasti, B.; Plaza, A. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 37–78. [CrossRef]
- 14. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep learning for hyperspectral image classification: An overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
- Bioucas-Dias, J.; Plaza, A.; Dobigeon, N.; Parente, M.; Du, Q.; Gader, P.; Chanussot, J. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* 2012, *5*, 354–379. [CrossRef]
- 16. Feng, X.; Li, H.; Wang, R.; Du, Q.; Jia, X.; Plaza, A. Hyperspectral unmixing based on nonnegative matrix factorization: A comprehensive review. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2022**, *15*, 4414–4436. [CrossRef]
- Bhatt, J.; Joshi, M. Deep learning in hyperspectral unmixing: A review. In Proceedings of the IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 2189–2192.
- 18. Chen, J.; Zhao, M.; Wang, X.; Richard, C.; Rahardja, S. Integration of physics-based and data-driven models for hyperspectral image unmixing: A summary of current methods. *IEEE Signal Process. Mag.* **2023**, *40*, 61–74. [CrossRef]
- 19. Borsoi, R.; Imbiriba, T.; Bermudez, J.; Richard, C.; Chanussot, J.; Drumetz, L.; Tourneret, J.; Zare, A.; Jutten, C. Spectral variability in hyperspectral data unmixing: A comprehensive review. *IEEE Geosci. Remote Sens. Mag.* 2021, *9*, 223–270. [CrossRef]
- Li, H.; Feng, X.; Zhai, D.; Du, Q.; Plaza, A. Self-supervised robust deep matrix factorization for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* 2021, 60, 5513214. [CrossRef]
- Hong, D.; Gao, L.; Yao, J.; Yokoya, N.; Chanussot, J.; Heiden, U.; Zhang, B. Endmember-guided unmixing network (EGU-Net): A general deep learning framework for self-supervised hyperspectral unmixing. *IEEE Trans. Neural Netw. Learn. Syst.* 2021, 33, 6518–6531. [CrossRef]

- Jin, Q.; Ma, Y.; Fan, F.; Huang, J.; Mei, X.; Ma, J. Adversarial autoencoder network for hyperspectral unmixing. *IEEE Trans. Neural Netw. Learn. Syst.* 2023, 34, 4555–4569. [CrossRef]
- Rasti, B.; Koirala, B.; Scheunders, P.; Ghamisi, P. UnDIP: Hyperspectral unmixing using deep image prior. *IEEE Trans. Geosci. Remote Sens.* 2021, 60, 5504615. [CrossRef]
- Zhao, M.; Wang, M.; Chen, J.; Rahardja, S. Hyperspectral unmixing for additive nonlinear models with a 3-D-CNN autoencoder network. *IEEE Trans. Geosci. Remote Sens.* 2021, 60, 5509415. [CrossRef]
- Su, Y.; Li, J.; Plaza, A.; Marinoni, A.; Gamba, P.; Chakravortty, S. DAEN: Deep autoencoder networks for hyperspectral unmixing. IEEE Trans. Geosci. Remote Sens. 2019, 57, 4309–4321. [CrossRef]
- Su, Y.; Xu, X.; Li, J.; Qi, H.; Gamba, P.; Plaza, A. Deep autoencoders with multitask learning for bilinear hyperspectral unmixing. IEEE Trans. Geosci. Remote Sens. 2020, 59, 8615–8629. [CrossRef]
- Palsson, B.; Sigurdsson, J.; Sveinsson, J.R.; Ulfarsson, M.O. Hyperspectral unmixing using a neural network autoencoder. *IEEE Access* 2018, *6*, 25646–25656. [CrossRef]
- Su, Y.; Marinoni, A.; Li, J.; Plaza, J.; Gamba, P. Stacked nonnegative sparse autoencoders for robust hyperspectral unmixing. *IEEE Geosci. Remote Sens. Lett.* 2018, 15, 1427–1431. [CrossRef]
- Ghosh, P.; Roy, S.K.; Koirala, B.; Rasti, B.; Scheunders, P. Hyperspectral unmixing using transformer network. *IEEE Trans. Geosci. Remote Sens.* 2022, 60, 5535116. [CrossRef]
- Zhao, M.; Shi, S.; Chen, J.; Dobigeon, N. A 3-D-CNN framework for hyperspectral unmixing with spectral variability. *IEEE Trans. Geosci. Remote Sens.* 2022, 60, 5521914. [CrossRef]
- Zhao, M.; Yan, L.; Chen, J. LSTM-DNN based autoencoder network for nonlinear hyperspectral image unmixing. *IEEE J. Sel. Top. Signal Process.* 2021, 15, 295–309. [CrossRef]
- 32. Zhang, X.; Sun, Y.; Zhang, J.; Wu, P.; Jiao, L. Hyperspectral unmixing via deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1755–1759. [CrossRef]
- 33. Tao, X.; Paoletti, M.E.; Han, L.; Wu, Z.; Ren, P.; Plaza, J.; Plaza, A.; Haut, J.M. A new deep convolutional network for effective hyperspectral unmixing. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2022**, *15*, 6999–7012. [CrossRef]
- Zou, L.; Zhu, X.; Wu, C.; Liu, Y.; Qu, L. Spectral–spatial exploration for hyperspectral image classification via the fusion of fully convolutional networks. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* 2020, 13, 659–674. [CrossRef]
- 35. Nalepa, J.; Myller, M.; Kawulok, M. Validating hyperspectral image segmentation. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1264–1268. [CrossRef]
- Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv* 2017, arXiv:1704.06857.
- Guo, Y.; Liu, Y.; Georgiou, T.; Lew, M.S. A review of semantic segmentation using deep neural networks. *Int. J. Multimedia Inf. Retr.* 2018, 7, 87–93. [CrossRef]
- Mo, Y.; Wu, Y.; Yang, X.; Liu, F.; Liao, Y. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing* 2022, 493, 626–646. [CrossRef]
- Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
- Qu, Y.; Qi, H. uDAS: An untied denoising autoencoder with sparsity for spectral unmixing. *IEEE Trans. Geosci. Remote Sens.* 2018, 57, 1698–1712. [CrossRef]
- 41. Zhao, M.; Wang, X.; Chen, J.; Chen, W. A plug-and-play priors framework for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* 2021, 60, 5501213. [CrossRef]
- Palsson, B.; Ulfarsson, M.O.; Sveinsson, J.R. Convolutional autoencoder for spectral–spatial hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* 2020, 59, 535–549. [CrossRef]
- 43. Wang, M.; Zhao, M.; Chen, J.; Rahardja, S. Nonlinear unmixing of hyperspectral data via deep autoencoder networks. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1467–1471. [CrossRef]
- Shahid, K.T.; Schizas, I.D. Unsupervised hyperspectral unmixing via nonlinear autoencoders. *IEEE Trans. Geosci. Remote Sens.* 2021, 60, 5506513. [CrossRef]
- 45. Shi, S.; Zhao, M.; Zhang, L.; Altmann, Y.; Chen, J. Probabilistic generative model for hyperspectral unmixing accounting for endmember variability. *IEEE Trans. Geosci. Remote Sens.* 2021, *60*, 5516915. [CrossRef]
- 46. Borsoi, R.A.; Imbiriba, T.; Bermudez, J.C.M. Deep generative endmember modeling: An application to unsupervised spectral unmixing. *IEEE Trans. Comput. Imaging* **2019**, *6*, 374–384. [CrossRef]
- 47. Han, Z.; Hong, D.; Gao, L.; Zhang, B.; Chanussot, J. Deep half-siamese networks for hyperspectral unmixing. *IEEE Geosci. Remote Sens. Lett.* 2020, *18*, 1996–2000. [CrossRef]
- 48. Zhou, C.; Rodrigues, M.R. ADMM-based hyperspectral unmixing networks for abundance and endmember estimation. *IEEE Trans. Geosci. Remote Sens.* 2021, 60, 5520018. [CrossRef]
- 49. Zhang, J.; Zhang, X.; Tang, X.; Chen, P.; Jiao, L. Sketch-based region adaptive sparse unmixing applied to hyperspectral image. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8840–8856. [CrossRef]
- Xiong, F.; Zhou, J.; Tao, S.; Lu, J.; Qian, Y. SNMF-Net: Learning a deep alternating neural network for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* 2021, 60, 5510816. [CrossRef]

- 51. Feng, X.R.; Li, H.C.; Liu, S.; Zhang, H. Correntropy-based autoencoder-like NMF with total variation for hyperspectral unmixing. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 5500505.
- 52. Zhang, S.; Li, J.; Li, H.C.; Deng, C.; Plaza, A. Spectral–spatial weighted sparse regression for hyperspectral image unmixing. *IEEE Trans. Geosci. Remote Sens.* 2018, 56, 3265–3276. [CrossRef]
- Min, A.; Guo, Z.; Li, H.; Peng, J. JMnet: Joint metric neural network for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* 2021, 60, 5505412. [CrossRef]
- 54. Palsson, B.; Sveinsson, J.R.; Ulfarsson, M.O. Blind hyperspectral unmixing using autoencoders: A critical comparison. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2022**, *15*, 1340–1372. [CrossRef]
- 55. Ozkan, S.; Kaya, B.; Akar, G.B. Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* 2018, 57, 482–496. [CrossRef]
- 56. Qian, Y.; Xiong, F.; Qian, Q.; Zhou, J. Spectral mixture model inspired network architectures for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* 2020, *58*, 7418–7434. [CrossRef]
- 57. Khajehrayeni, F.; Ghassemian, H. Hyperspectral unmixing using deep convolutional autoencoders in a supervised scenario. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2020**, *13*, 567–576. [CrossRef]
- 58. Yang, B. Supervised nonlinear hyperspectral unmixing with automatic shadow compensation using multiswarm particle swarm optimization. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5529618. [CrossRef]
- Xu, X.; Shi, Z.; Pan, B. A supervised abundance estimation method for hyperspectral unmixing. *Remote Sens. Lett.* 2018, 9, 383–392. [CrossRef]
- Li, J.; Li, X.; Huang, B.; Zhao, L. Hopfield neural network approach for supervised nonlinear spectral unmixing. *IEEE Geosci. Remote Sens. Lett.* 2016, 13, 1002–1006. [CrossRef]
- 61. Wan, L.; Chen, T.; Plaza, A.; Cai, H. Hyperspectral unmixing based on spectral and sparse deep convolutional neural networks. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2021**, *14*, 11669–11682. [CrossRef]
- Altmann, Y.; Halimi, A.; Dobigeon, N.; Tourneret, J.Y. Supervised nonlinear spectral unmixing using a polynomial post nonlinear model for hyperspectral imagery. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 1009–1012.
- 63. Koirala, B.; Khodadadzadeh, M.; Contreras, C.; Zahiri, Z.; Gloaguen, R.; Scheunders, P. A supervised method for nonlinear hyperspectral unmixing. *Remote Sens.* 2019, *11*, 2458. [CrossRef]
- Lei, M.; Li, J.; Qi, L.; Wang, Y.; Gao, X. Hyperspectral Unmixing via Recurrent Neural Network With Chain Classifier. In Proceedings of the IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 2173–2176.
- 65. Mitraka, Z.; Del Frate, F.; Carbone, F. Nonlinear spectral unmixing of landsat imagery for urban surface cover mapping. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2016**, *9*, 3340–3350. [CrossRef]
- 66. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
- Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
- 68. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.