

Article

A Split-Frequency Filter Network for Hyperspectral Image Classification

Jinfu Gong^{1,2,3}, Fanming Li^{1,3}, Jian Wang^{1,2,3}, Zhengye Yang^{1,2,3} and Xuezhuan Ding^{1,3,*}

¹ Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China; gongjinfu20@mailsucas.ac.cn (J.G.); lifanming@mail.sitp.ac.cn (F.L.); wangjian@mail.sitp.ac.cn (J.W.); yangzhengye20@mailsucas.ac.cn (Z.Y)

² University of Chinese Academy of Sciences, Beijing 100049, China

³ CAS Key Laboratory of Infrared System Detection and Imaging Technology, Shanghai Institute of Technical Physics, Shanghai 200083, China

* Correspondence: dingxuezhuan@mail.sitp.ac.cn

Abstract: The intricate structure of hyperspectral images comprising hundreds of successive spectral bands makes it challenging for conventional approaches to quickly and precisely classify this information. The classification performance of hyperspectral images has substantially improved in the past decade with the emergence of deep-learning-based techniques. Due to convolutional neural networks' (CNNs) excellent feature extraction and modeling, they have become a robust backbone network for hyperspectral image classification. However, CNNs fail to adequately capture the dependency and contextual information of the sequence of spectral properties due to the restrictions inherent in their fundamental network characteristics. We analyzed hyperspectral image classification from a frequency-domain angle to tackle this issue and proposed a split-frequency filter network. It is a simple and effective network architecture that improves the performance of hyperspectral image classification through three critical operations: a split-frequency filter network, a detail-enhancement layer, and a nonlinear unit. Firstly, a split-frequency filtering network captures the interactions between neighboring spectral bands in the frequency domain. The classification performance is then enhanced using a detail-improvement layer with a frequency-domain attention technique. Finally, a nonlinear unit is incorporated into the frequency-domain output layer to expedite training and boost performance. Experiments on various hyperspectral datasets demonstrate that the method outperforms other state-of-art approaches (an overall accuracy(OA) improvement of at least 2%), particularly when the training sample is insufficient.

Keywords: hyperspectral image classification; split-frequency filter network; detail-enhancement layer; nonlinear unit



Citation: Gong, J.; Li, F.; Wang, J.; Yang, Z.; Ding, X. A Split-Frequency Filter Network for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 3900. <https://doi.org/10.3390/rs15153900>

Academic Editors: Chein-I Chang and Hossein M. Rizeei

Received: 20 June 2023

Revised: 29 July 2023

Accepted: 3 August 2023

Published: 7 August 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/>).

1. Introduction

Hyperspectral imaging has emerged as a popular field of study in optical remote sensing in recent years due to the rapid growth of remote sensing technologies. By generating tens to hundreds of related spectral bands with a particular spectrometer, a hyperspectral image, i.e., a three-dimensional image integrating spatial and spectral information to detect specific features, captures minute spectral differences across various materials. Therefore, hyperspectral remote sensing technology is widely used for agricultural land cover [1], urban green belt planning [2], water quality and pollution detection [3], ecological forest monitoring [4], and military target detection [5].

Hyperspectral image (HSI) classification uses the spectral variation among image elements in different wavelength bands and the spatial structure feature information to accurately classify features. There are still many aspects that can be improved, even though hyperspectral remote sensing is widely employed for remote sensing detection.

Specifically, the phenomenon of “same object with different spectrums” lessens the classification accuracy, the small number of labeled samples makes training difficult, and the redundancy of data between bands results in a dimensional explosion. In the past decade, feature extraction has become the most critical aspect of hyperspectral image classification, and many artificially designed shallow feature extraction and deep learning algorithms have emerged [6].

Shallow feature extraction initially adopted statistical methods to measure the similarity in spectral information. However, this type of method can only achieve a limited accuracy. With the advancement of machine learning, HSI classification based on machine learning is now commonly applied. These machine learning methods usually first require feature engineering on the data and then classification of the pre-processed features using a classifier. Standard feature engineering methods include the principal component analysis (PCA) [7], linear-discriminant analysis (LDA) [8], and independent component analysis (ICA) [9]. Common classifiers include the K-nearest neighbor (KNN) [10], support vector machine (SVM) [11], random forest (RF) [12], and other methods. PCA-based methods are also widely used in hyperspectral radiative transfer modeling [13,14]. With the new concepts proposed in other fields, the performance of traditional machine learning algorithms has dramatically improved. Kang et al. [15] combined edge-preserving filtering with the SVM to propose a feature suitable for extracting spatial-spectral features. Zhong et al. [16] However, as the training size grows to be more prominent and the complexity of the training data increases, shallow feature extraction algorithms experience performance bottlenecks.

With the development of deep learning, deep feature extraction has grown exponentially. These types of feature extraction techniques construct an end-to-end framework by automatically learning aspects of the data from the original data. Deep-learning-based feature extraction methods are more robust, differentiated, and abstract than shallow feature extraction methods [6]. Among the various deep-learning-based models, stacked autoencoders (SAEs) [17], recurrent neural networks (RNNs) [18], convolutional neural networks (CNNs) [19], graph convolutional neural networks (GNNs) [20], UNet-based neural networks [21], and the Transformer [22] are the most popular model frameworks.

To extract hyperspectral features, autoencoders (AEs) are the most frequently used method in deep learning. In [23], Chen et al. originally used deep learning to categorize images from downsampled hyperspectral images obtained via PCA by stacking multiple self-encoders. To simplify the model, Zebalza et al. [24] presented a segmented SAE, which divided the original spectral information into more minor spectral features and processed them using numerous SAEs. An AE usually requires the data to be downsampled into one-dimensional vectors in spatial dimensions, ignoring the rich spectral-spatial structure information of the hyperspectral data.

Developments in sequential data processing applications such as speech recognition and machine translation have resulted in the widespread application of RNNs, while spectral data can also be considered sequential. Mou et al. [18] proposed the first RNN framework applied to hyperspectral classification by using an improved gated cyclic unit PRtanh and treating hyperspectral image pixels as sequential data. Hang et al. [25] grouped adjacent spectra of HSIs and used RNNs for the grouped spectral bands to eliminate redundant information. Learning long-term correlations is challenging for RNNs, because they learn spectral characteristics sequentially, which is highly dependent on the sequential input of the spectral bands; therefore, long short-term memory (LSTM) was proposed as a solution to the gradient disappearance problem. For this reason, a LSTM is often used to solve this problem. Liu et al. [26] proposed a bidirectional convolutional LSTM that takes all spectra as the input to a bidirectional LSTM to learn the dependencies in the frequency domain. Zhou et al. [27] proposed a spectral-spatial LSTM in which the spectral information of each pixel is first input to the spectral LSTM for learning. Then, the spatial information near to the pixel is input to the spatial LSTM for learning, and finally, decision fusion is used to obtain the spectral classification results. RNNs operate in a recursive-like

manner and fail to perform parallelized computations, which limits the computational efficiency of RNNs.

CNNs are mainly used to extract local, two-dimensional spatial or spectral features from images. Hu et al. [19] utilized 1D CNN models for HSI classification to extract each pixel's spectral information. After that, Zhao et al. [28] used the 2D CNN for HSI classification and preserved the spatial information of the HSI as much as possible compared to SAE. Chen et al. [17] applied 3D CNNs to HSI classification and compared the features of 1D CNNs, 2D CNNs, and 3D CNNs in detail. All three of these works are representative and are early attempts to apply CNNs for hyperspectral image classification. After that, CNNs have mainly been used to examine how to use HSI data efficiently and synthetically in both spectral and spatial dimensions. Lee et al. [29] proposed ContextNet to explore local contextual interactions by jointly exploiting local spatial–spectral relationships between individual pixel vectors. Roy et al. [30] proposed a 3D–2D CNN (HybridSN) using a network with a mixture of 3D CNNs and 2D CNNs to extract features and effectively extracted the complementary spectral–spatial information. Roy et al. [31] proposed A²S²KResNet, which enhances the classification performance by using an efficient feature recalibration and 3D convolution to extract features. CNNs are powerful methods for extracting spatial structure and local context information, but they inevitably encounter performance bottlenecks for data with sequence properties, such as spectral data.

Graph neural networks were created to process graph data, and with the proposed graph convolutional neural networks, they became a popular research area for hyperspectral classification. Hong et al. first proposed a miniGCN in [20], explored the feasibility of fusing CNNs and GCNs, and illustrated the usage scenarios and advantages of a miniGCN. Zhang et al. [32] proposed a global random graph convolution and network in which graphs can be generated via random sampling from labeled data. The graph size can be small to save computational resources. The CNN–Enhanced Graph Convolutional Network (CEGCN) was proposed by Liu et al. [33]. The CEGCN is a CNN-enhanced GCN architecture that generates complementary spectral–spatial information in various dimensions of pixels and superpixels by extracting features from GCNs and CNNs in large-scale irregular regions. Graphical neural volumes and networks inevitably face the problem of computationally intensive and insufficient processing of hyperspectral spectral information when processing hyperspectral data.

U-Net [34] is a classical deep image segmentation network structure composed of an encoder and decoder. This network better represents deeper semantic features by combining positional and semantic information. Lin et al. [35] proposed a novel network structure, CAGU (Context-Aware Attentional Graph U-Net), which combines UNet and a graph neural network. It can transform the spectral features into a highly cohesive state, and the classification effect is very good. Li et al. [21] proposed a PSE-UNet model combining a PCA, attention mechanism, and UNet and analyzed the factors affecting the model's performance. Liu et al. [36] combined a CNN, UNet, and graph neural nets and proposed a Multi-Stage Superpixel Structured Hierarchical Graph UNet (MSSHU) to learn multiscale features and achieve better classification results. UNet-based networks are often combined with other network structures and could be a popular field for hyperspectral analysis in the future.

The Transformer was proposed by Vaswani et al. [37] in 2017 and was initially applied to NLP. When the Vision Transformer was proposed [38], the difficulty of applying the Transformer to images was solved by segmenting the image into several image blocks. Moreover, the Transformer uses self-attention to process and analyze sequential data more efficiently, which is well-suited for HSI data processing. He et al. [39] was based on a BERT language model using a multi-headed self-attentive mechanism (MHSA) that can capture global correlations between input spectral regions. Meanwhile, the number of papers borrowing the structure of the Transformer model is increasing, and Liu et al. [40] proposed a CAN (Central Attention Network) to optimize the computational mechanism of the Transformer and improve the classification performance.

We hope to reconsider the hyperspectral classification problem from different perspectives based on the above discussion. Frequency-domain hyperspectral classification is yet to be studied. We hope to reconsider the hyperspectral classification problem from different perspectives based on the above discussion. Rao et al. [41] proposed a global filter network that overcomes these drawbacks by learning the frequency domain's medium- and long-term dependencies. To address the issue of insufficient spectral-spatial feature extraction in the frequency domain with limited samples, we present a split-frequency filter network for detailed hyperspectral data, which was inspired by the Global Filter Network (GFNet). The contributions of this study, specifically, are outlined as follows:

1. The proposed network can model the medium- and long-term dependencies between bands in frequency-domain hyperspectral sequences by converting the hyperspectral data feature extraction problem into a frequency-domain sequence learning problem using a split-frequency filtering network. Compared with the GFNet, our proposed network can be better adapted to hyperspectral data.
2. For the discrete Fourier transforms, the assumption of global convolution for periodic images does not apply to hyperspectral images. To compensate for local features and non-periodic boundaries, we add a detail-enhancement layer after the separation filter network to improve the classification performance of HSIs.
3. The split-frequency filter network is modified by adding the nonlinear activation function Mish, which alters the network's original single linear structure and increases the classification performance and network throughput.
4. On three well-known HSI datasets, Indian Pines, Pavia University, and WHU Hi Longkou, we qualitatively and quantitatively assess the classification performance of the proposed SFFN. The experimental findings demonstrate that our proposed SFFN significantly outperforms other state-of-art networks (an OA improvement of at least 2%).

The remaining sections of the essay are arranged as follows. Section 2 reviews the necessary knowledge and describes the design of the proposed method. Section 3 presents the dataset, experimental settings, and results. Section 4 carries out a discussion and analysis of the experiment. Section 5 summarizes and concludes the article.

2. Methods

In this subsection, we first review the global filtering network. On this basis, we propose improvements such as split-frequency filter networks and frequency-domain detail-enhancement layers to make them more applicable to hyperspectral classification image tasks. Finally, we also make other improvements that can improve the classification accuracy.

2.1. Overview of the Global Filter Network

The global filtering network (GFNet) is a novel MLP network proposed by Rao et al. [41]. The fundamental idea of this architecture is to learn about the spatial inter-connections of images by exploiting the global features of the frequency domain. This method learns the relationships between image tokens through a series of global filters that can be learned, as opposed to the self-attention mechanism [37] of the Vision Transformer [38] and the MLP model [42]. Global filters differ from CNN networks, which usually process images using a person's relatively small convolutional kernel to mine local contextual information. Global filters can cover all frequency domains when processing an image and, therefore, can model the image globally to capture medium and long-term dependencies.

The global filtering network's discrete Fourier transform (DFT) is a crucial component and is necessary for the design of this network. Images are usually a two-dimensional type

of information, and for a given piece of two-dimensional information $S[x, y]$, $0 \leq x \leq X - 1$, $0 \leq y \leq Y - 1$, the 2D **DFT** of $S[x, y]$ is given by Equation (1):

$$S[x, y] = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} s[x, y] e^{-j2\pi(\frac{ux}{X} + \frac{vy}{Y})} \quad (1)$$

where $S[x, y]$ represents an image of size $X \times Y$. Equation (1) must be considered for the discrete variables u and v in $u = \{0, 1, 2, \dots, X - 1\}$ and $v = \{0, 1, 2, \dots, Y - 1\}$.

Given $F(u, v)$, the original signal $S[x, y]$ can be recovered via inverse discrete Fourier transform (**IDFT**), as shown in Equation (2):

$$S[x, y] = \frac{1}{XY} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} S[u, v] e^{j2\pi(\frac{ux}{X} + \frac{vy}{Y})} \quad (2)$$

For the actual input $S[x, y]$ for **DFT**, the nature of the conjugate symmetry can be proven using Equation (3), $S[X - u, Y - v] = S^*[u, v]$. Similarly, the real discrete signal can be recovered using **IDFT** for the conjugate symmetric $S[k]$. Furthermore, the fast Fourier transform (**FFT**) algorithm [43] can improve the computational efficiency of the 2D **DFT**. The **FFT** is a fast algorithm for **DFT**, and all references to **DFT** in the article were implemented using the **FFT**. With this property, we can save only the value of S to save all information.

$$\begin{aligned} S[X - u, Y - v] &= \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} s[x, y] e^{-j2\pi(\frac{(X-u)m}{X} + \frac{(Y-v)n}{Y})} \\ &= \sum_{m=0}^{X-1} \sum_{n=0}^{Y-1} s[x, y] e^{j2\pi(\frac{ux}{X} + \frac{vy}{Y})} = S^*[u, v] \end{aligned} \quad (3)$$

In [41], the $H \times W$ image is split into several non-overlapping image patches as $h \times w$ and then projected into flattened patches of dimension d as $L = XY$. For the data s , a two-dimensional **FFT** operation is performed using Equation (2), as shown in Equation (4).

$$S = \mathcal{F}[s] \in \mathbb{C}^{h \times w \times d} \quad (4)$$

where $\mathcal{F}[\cdot]$ represents the two-dimensional **FFT**. A complex tensor named S is used to indicate the spectrum of c . As illustrated in Equation (5), the spectrum can be modified by multiplying by a global filter ($W \in \mathbb{C}^{h \times w \times d}$).

$$\tilde{S} = W \odot S \quad (5)$$

where the filter W is referred to as the global filter, and \odot is the Hadamard product. Finally, as stated in Equation (6), the modulated spectrum \tilde{S} is converted into the spatial domain, and s is updated using the inverse fast Fourier transform (**IFFT**) of Equation (3).

$$s \leftarrow \mathcal{F}^{-1}[\tilde{S}] \quad (6)$$

The global filter W , which has a filter size of $h \times w$, is analogous to the global circular convolution in deep learning and can be thought of as a collection of learnable frequency filters with various hidden dimensions.

2.2. Split-Frequency Filter Network

We propose a new general network based on improving the global filter network, the split-frequency filter network (SFFN). Compared with the global filtering network, this network focuses on the spectral characteristics in hyperspectral images and can be effectively applied to high-precision and fine hyperspectral image classification. For this purpose, we design two focus modules, the split-frequency filter module (SF) and the frequency-domain detail-enhancement module (FDE), to improve the spectral discrimina-

tion of fine spectral differences and reduce the detail loss caused by the frequency-domain transform, respectively. Our proposed network architecture is depicted in Figure 1.

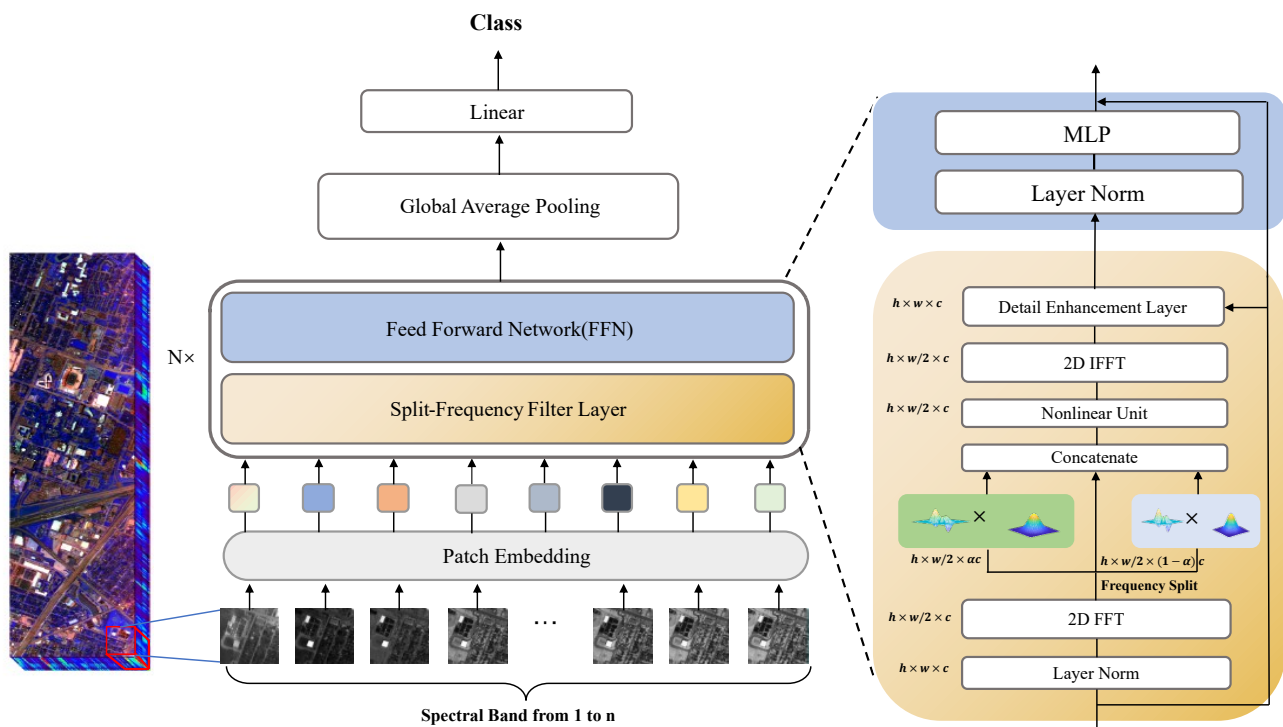


Figure 1. An overview of the split-frequency filter network. Our network structure based on GFNet with numerous alterations. We added a detail-enhancement layer, a nonlinear network, and a split-frequency filter in favor of a global filter.

Unlike the RGB images presented in [41], the hyperspectral images are densely sampled by hundreds of spectral channels at tiny intervals (e.g., 10 nm) from the electromagnetic spectrum to produce near-continuous spectral features. The spectral dimensions at different locations in the image reflect the absorption characteristics of different objects for different wavelengths. They can be learned as object-classification features to capture small differences in the spectra of different objects. Unlike in the previous analysis, we reconsidered the backbone network design from the frequency-domain perspective. Because of the similarity between spectral information and sequence data (i.e., a continuous and strong interclass correlation between data), they are used as sequence information input to the network. The spectral bands of the hyperspectral datasets (such as the Indian Pave Data without bands 104–108, 150–163, and 220) are not continuous and contain some discontinuity. Sometimes this redundancy may reduce the classification accuracy due to its increase within the variance in the feature space and the decrease between the category variances. Inspired by [44], we propose the split-frequency filter network to improve this phenomenon.

In order to create a spectral cube, the input vector is created by selecting nearby pixels and centering them on the training pixel. Along with spectral information about the point, this spectral cube also includes spatial data about the area around the point. For a given spectral feature $S = [s_1, s_2, \dots, s_c] \in \mathbf{R}^{1 \times C}$, to analyze the medium- and long-term dependencies between the spectra, it is necessary to introduce the relative and absolute positional information about the spectra, where the position encoding approach is used. After the FFT, the feature map can be transformed into $S_f \in \mathbf{R}^{H \times W \times C}$, where S_f, H, W and C represent the feature map after the FFT, the height, the width, and the dimension of the feature, respectively. The intended spectral space patch size defines H and W , while the hyperspectral number is defined as C . We split and analyze the spectra using specific ratios to better investigate the frequency-domain correlations between the surrounding spectra. The segmentation S_f is first performed along the spectral channel dimension,

i.e., $S_f = \{S_l, S_s\}$, where $S_l \in \mathbf{R}^{H \times W \times (1-\alpha_{in})C}$ represents the segmented long features, $S_s \in \mathbf{R}^{H \times W \times \alpha_{in}C}$ represents the segmented short features, and $\alpha_{in} \in [0, 1]$ represents the percentage assigned to the feature channels. $\tilde{S} \in \mathbf{R}^{H \times W \times C}$ is used as the output tensor. Similarly, let $\tilde{S} = \{\tilde{S}_l, \tilde{S}_s\}$ be the global split, and the global ratio of the output tensor is determined by the hyperparameter $\alpha_{out} \in [0, 1]$. The specific flowchart of split-frequency-domain filtering is shown in Figure 2. We assume that the segmentation ratio is 0.5 to simplify the computation. The spectral information of two branches learns different types of filtering information through two uncorrelated local filters. Finally, the global news and the two other pieces of filtered information are concatenated to ensure that the fused channel size is the same as the input channel size.

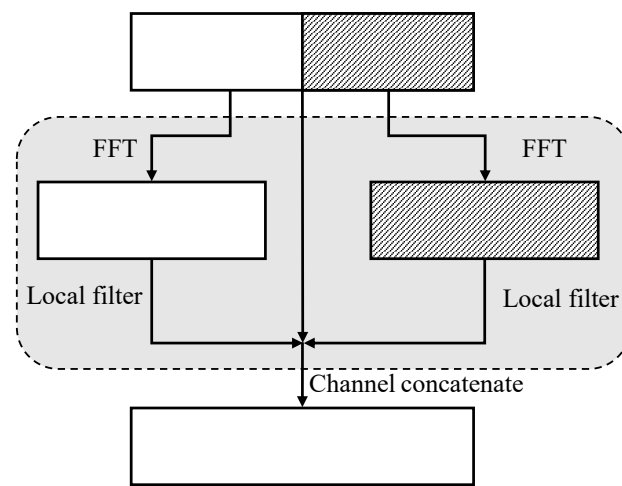


Figure 2. The overall flowchart of the split-frequency filter. In this figure, the split ratio is 0.5. By determining the proper division ratio, the hyperspectral image classification capabilities may have been significantly enhanced.

The updated Equations (7) and (8) are written as follows:

$$\tilde{S}_l = \mathbf{W}_1 \odot S_l \quad (7)$$

$$\tilde{S}_s = \mathbf{W}_2 \odot S_s \quad (8)$$

where K_1 and K_2 represent two different frequency-domain filtering kernels of $\mathbf{W}_1 \in \mathbf{R}^{H \times W \times (1-\alpha_{in})C}$ and $\mathbf{W}_2 \in \mathbf{R}^{H \times W \times \alpha_{in}C}$, respectively. After applying the split-frequency filter, the two feature maps are fused, and skip connections are set to reduce the information loss before and after split-frequency filter convolution and enhance the information exchange between layers. The equation is shown in (9).

$$\mathbf{Y} = \text{Concatenate}\{\mathbf{W}_2 \odot S_s, \mathbf{W}_1 \odot S_l\} + S \quad (9)$$

We discuss the specific role of the split-frequency filter in Section 4.

2.3. Detail-Enhancement Layer

When hyperspectral images are being used, the discrete Fourier transform is not assumed to conduct convolution operations on periodic images. We build a detail-enhancement layer to carry out information compensation, increase the classification accuracy, and make up for the loss of local features and non-periodic boundaries. The detail-enhancement layer is mainly composed of FcaNet and a skip connection.

FcaNet is a novel channel attention mechanism that was proposed by [45], and the main idea of FcaNet is to view the channel of attention. The two-dimensional DCT is used

to compress more feature information on behalf of multiple frequency-domain components for better information compression. The main flowchart of Fca is shown in Figure 3. The different colored blocks represent the results of dot multiplication with different 2D DCT frequency-domain components in Figure 3.

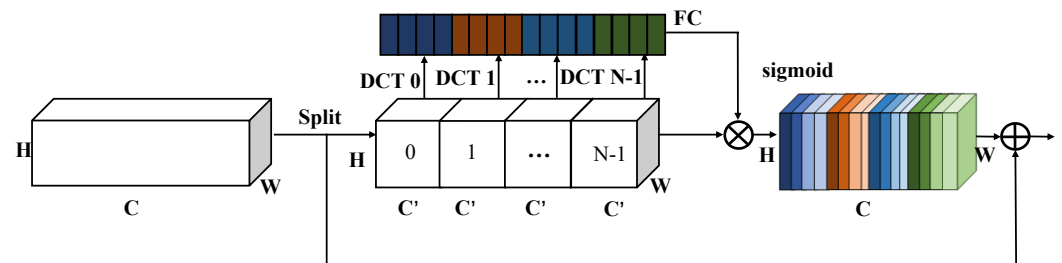


Figure 3. The overall architecture of the detail-enhancement layer. Our architecture is based on FcaNet with some minimal modifications.

The input feature map X is divided into several sections along the channel dimension, indicated as $[X_0, X_1, \dots, X_{n-1}]$, where X_i is divisible by n and C is equal to X_i in $X_i \in \mathbf{R}^{C' \times H \times W}$, where i in $0, 1, \dots, n-1, C' = \frac{C}{n}$. By multiplying the corresponding X_i by the corresponding 2D DCT frequency components and summing all of the components, the compression result of the channel's attention can be obtained. The equation is shown in (10).

$$F^i = 2DDCT^{u_i, v_i}(X^i),$$

$$= \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} X_{:,h,w}^i B_{h,w}^{u_i, v_i} \quad i \in \{0, 1, \dots, n-1\} \quad (10)$$

where $[u_i, v_i]$ corresponds to the two-dimensional component of the frequency of X_i , and F^i is the compressed C' . The compressed vectors can be obtained by merging, and the Equation (11) is shown below.

$$F = \text{compress}(X)$$

$$= \text{cat}\left(\left[F^0, F^1, \dots, F^{n-1}\right]\right) \quad (11)$$

where F is the obtained multi-frequency vector. The acquired multi-frequency vector is processed by the fully connected layer and sigmoid function to reveal the data on the feature map. The obtained detailed information is placed on the feature map to compensate for the loss of local features and non-periodic boundaries. The Equation (12) expresses the whole process.

$$ms_att = \text{sigmoid}(fc(F)) \odot X + X \quad (12)$$

2.4. Other Optimization Methods

2.4.1. Nonlinear Unit

After frequency-domain transformation, the image's information is distributed unevenly; most is concentrated in the low-frequency range, while the high-frequency information is frequently ignored. The performance and accuracy of the network can be enhanced by making the network more nonlinear at the frequency-domain depth, which can better represent the interdependence between low and high frequencies. The nonlinear activation function can introduce richer features and greater expressiveness, allowing the neural network to adapt better to hyperspectral image classification.

The Mish activation function [46] is a nonlinear activation function proposed by Diganta Misra in 2019. Compared to other common activation functions, it has the following characteristics:

- Smoothness: The Mish is a smooth, nonlinear function whose first- and second-order derivatives are continuous throughout the real domain.
- Negative values are supported: Unlike ReLU [47], the Mish activation function can support negative-valued inputs without the problem of dead neurons.
- Non-monotonicity: The Mish activation function has a local minimum when the input is 0. This monotonicity can help the network to avoid falling into local optima under certain conditions.
- The Mish activation function morphology is similar to that of tanh. However, it has a more expansive “plateau area” than the tanh function and can be more efficient than the tanh function in certain situations.

The Mish activation function equation can be written as (13).

$$\text{Mish}(x) = x \tanh(\text{softplus}(x)) \quad (13)$$

where $\text{softplus} = \ln(1 + e^x)$. The Mish activation module is placed after Formula (9), through which the operation increases the nonlinearity of the network in the frequency domain and adaptively adjusts the frequency characteristics according to the split-frequency filters in the frequency domain, improving the network’s robustness and hyperspectral classification ability. A comparison between the nonlinear unit and the linear unit can be seen in Figure 4.

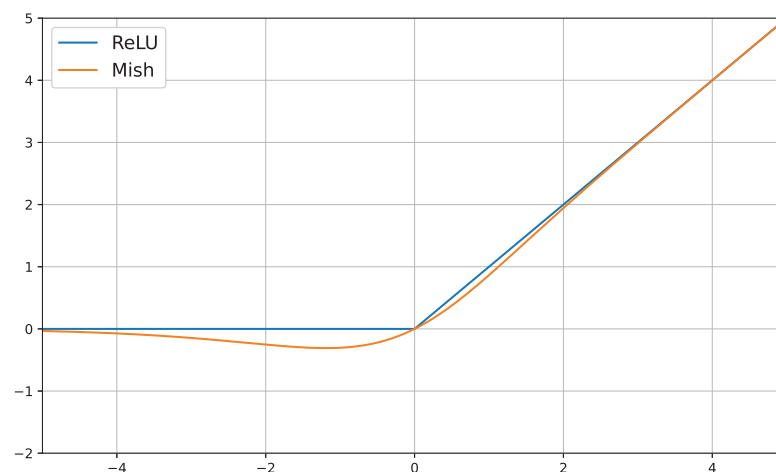


Figure 4. Graph of the difference between the nonlinear unit and the linear unit.

2.4.2. Improvement in MLP

The Vision Transformer mainly uses gaussian error linear unit (GELU) [48] as the activation function between MLP and the layers. We were inspired by [49] to use StarReLU as the activation function. The equation that can be used for GELU is (14).

$$\text{GELU}(x) = x\Phi(x) \approx 0.5 \times x \left(1 + \tanh \left(\sqrt{2/\pi} \left(x + 0.044715 \times x^3 \right) \right) \right) \quad (14)$$

The StarReLU equation can be written as (15).

$$\text{StarReLU}(x) = s \cdot (\text{ReLU}(x))^2 + b \quad (15)$$

where $s \in \mathbf{R}$ and $b \in \mathbf{R}$ are learnable parameters. StarReLU requires four FLOP, which is much lower than the fourteen FLOP of GELU and indicates a better performance. Following testing, this activation function can increase the classification accuracy of hyperspectral classification while speeding up training. Figure 5 reflects the variation in the StarReLU curve for different values of s and a . Compared to the fixed and constant GELU curve, StarReLU can be well-adapted to the changes in the model.

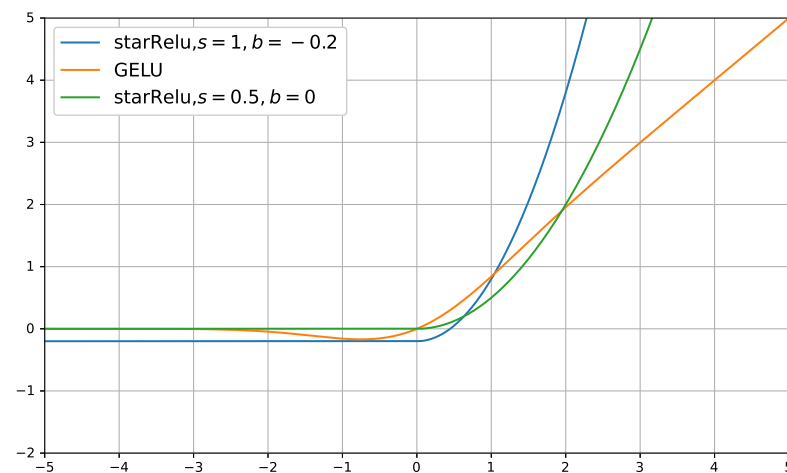


Figure 5. Graph of the differences in the activation functions (GELU and different values of StarReLU).

3. Results

This section first presents the characteristics and parameters of the three datasets, followed by the precise optimization details of the implementation, which are contrasted with those of state-of-the-art techniques. Finally, the proposed hyperspectral classification performance method is evaluated qualitatively and quantitatively.

3.1. Description of the Experimental Datasets

In this experiment, we selected three real-shot hyperspectral datasets as experimental objects to verify the effectiveness of SFFNet. The three hyperspectral datasets were the Indian Pines (IP), the University of Pavia (PU), and The WHU-Hi-LongKou (WHU), and their essential information is shown in Table 1. Among these datasets, the spectral bands of the IP and PU datasets were removed from the noise and water vapor bands to prevent interference with the classification task. The Indian Pines dataset originally had 220 bands, and after removing the noisy bands 104–108, 150–163, and 220, the remaining 200 bands of data were retained. The Pavia University dataset originally had 115 bands, and the dataset producer removed 12 noisy bands. Our search of the data did not find the removed band numbers. It should be noted that the WHU-Hi-LongKou dataset [50] is a hyperspectral dataset collected by the Intelligent Remote Sensing Data Extraction Analysis and Application Research Group of Wuhan University, and it was made using unmanned aerial photography of agricultural areas. We did not perform a deletion wave operation on this dataset.

Table 1. Basic information on IP, PU, and the WHU-Hi-LongKou.

No.	IP	PU	The WHU-Hi Longkou
Number of bands used for classification	200	103	270
Spectral range (μm)	0.4–2.5	0.43–0.86	0.4–1
Data size (pixel)	145×145	610×340	550×400
Spatial resolution (m)	20	1.3	0.463
Spectral resolution (nm)	10	5	6
Number of classes	16	9	9
Number of labeled data points	10,249	42,776	204,542

The truth samples of the three datasets were divided into training, validation, and test sets in this experiment at random. Only the truth samples of 5% IP, 1% PU, and 0.1% WHU were utilized to train the models to explore the robustness of the classification performance of different models under minor sample conditions. Additionally, the validation set was designed to contain the same number of samples as the training set, but it was not used in

training to track changes in performance. All of the sample characteristics of these datasets are dissimilar. The sample distribution for the IP dataset is imbalanced, such that some categories contain only a small number of training samples, which might have a significant impact on the classification results. A similar issue of unequal sample distribution exists for the PU and WHU datasets. However, at the same time, there is a greater chance of misclassification because there are substantially fewer training samples than test samples. Tables 2–4 illustrate the corresponding ranges of the training, validation, and testing samples for the IP, PU, and WHU datasets.

Table 2. Detailed sample distribution of the training, validation, and testing datasets from Indian Pines.

No.	Category	Labeled Samples	Training	Validation	Testing
1	Alfalfa	46	2	3	41
2	Corn	1428	71	72	1285
3	Corn-mintill	830	42	41	747
4	Corn	237	12	12	213
5	Grass-pasture	483	24	24	435
6	Grass-tree	730	36	37	657
7	Grass-pasture-moved	28	2	1	25
8	Hay-windrowed	478	24	24	430
9	Oats	20	1	1	18
10	Soybean-notill	972	48	49	875
11	Soybean-mintill	2455	123	122	2210
12	soybean-clean	593	30	29	534
13	Wheat	205	10	10	185
14	Woods	1265	63	63	1139
15	Buidings-Grass-Trees-Drives	386	19	20	347
16	Stone-Steel-Towards	93	5	4	84
	Total	10,249	512	512	9225

Table 3. Detailed sample distribution of the training, validation, and testing datasets from the University of Pavia.

Class	Class Name	Labeled Samples	Training	Validation	Testing
class 1	ASphalt	6631	66	66	6499
class 2	Meadows	18,649	186	186	18,277
class 3	Gravel	2099	20	20	2049
class 4	Trees	3064	30	30	3004
class 5	Painted metal sheet	1345	13	13	1319
class 6	Bare Soil	5029	50	50	4929
class 7	Bitumen	1330	13	13	1304
class 8	Self-Blocking Bricks	3682	36	36	3610
class 9	Shadows	947	9	9	929
	Total	42,776	423	423	41,930

Table 4. Detailed sample distribution of the training, validation, and testing datasets from the WHU-Hi-LongKou.

Class	Class Name	Labeled Samples	Training	Validation	Testing
class 1	Corn	34,511	34	34	34,443
class 2	Cotton	8374	8	8	8358
class 3	Sesame	3031	3	3	3025
class 4	Broad-leaf soybean	63,212	63	63	63,086
class 5	Narrow-leaf soybean	4151	4	4	4143
class 6	Rice	11,854	11	11	11,832
class 7	Water	67,056	67	67	66,922
class 8	Roads and houses	7124	7	7	7110
class 9	Mixed weed	5229	5	5	5219
	Total	204,542	202	202	204,340

3.2. Experimental Environment and Contrasting Models

The hardware environment employed for this experiment was as follows: the graphics card was NVIDIA GTX 3090Ti 24 GB, the CPU is i7-12700K, and the memory was

64 GB. The software environment for the experiment was deployed in Pytorch 1.11.0 and Python 3.8. To verify the validity of SFFN, the above-mentioned SVM [15], ResNet [51], SSRN [52], DPyResnet [53], ContextualNet [29], A2S2KResNet [31], PSE-UNet [21], SpectralFormer [22], and GFNet [41] were used as comparison models for the experiments. It is important to note that SVM refers to the algorithm developed by Kang et al. [15], not the conventional SVM.

The comparison models' network and parameter designs were in line with the corresponding publications. AdamW served as the optimizer for the SFFN described in this research, with the starting learning rate set to $\frac{\text{batch size}}{1024} \times 0.001$ and used cosine scheduling to decay the learning rate to 2×10^{-5} . In the experiments, linearity was used to warm up the learning rate in the first ten cycles, and gradient cropping was later used to stabilize the training process for 300 epochs.

The evaluation metrics quantified the classification performance of each model using the overall accuracy (OA), average accuracy (AA), and kappa coefficient (κ). OA represents the percentage of correctly identified samples out of all samples. It is calculated by summing the confusion matrix's diagonal elements and dividing by the overall sample size. AA stands for the average accuracy for each class. It is obtained by averaging these values after dividing the confusion matrix's diagonal elements by the total number of samples for each class. A measure of agreement that accounts for the potential for random agreement is the kappa coefficient (κ), which indicates how well the classification and the reference data agree. The estimated value is obtained by comparing the observed agreement (OA) with the expected agreement (EA), which is based on the marginal frequencies of the confusion matrix.

3.3. Quantitative Results and Analysis

Tables 5–7 present quantitative classification results for three general metrics (i.e., OA, AA, and kappa) for IP, PU, and WHU, respectively. We mark the optimal results in bold in Tables 5–7. Overall, we propose that SFFN achieves the best classification accuracy in all datasets and that SFFN achieves the highest classification accuracy across all datasets. In Table 5, our proposed method has the best ratings in terms of the OA, AA, and kappa at 98.47%, 98.78%, and 98.26%, respectively. Compared with SVM, ResNet, SSRN, PyResNet, ConTextualNet, A2S2KNet, PSE-UNet, SpectralFormer, and the basic framework GFNet, the OA was improved by 9.91%, 8.4%, 1.76%, 11.47%, 11.04%, 2.08%, 6.23%, 25.68%, and 3.11%, respectively.

Table 5. Classification results (%) of different models in the Indian Pines dataset.

Class	SVM	ResNet	SSRN	PyResNet	ContextualNet	A2S2KNet	PSE-UNet	SpectralFormer	GFNet	Proposed
class 1	1.0000	1.0000	1.0000	0.9444	0.7878	0.9743	0.6511	0.6315	0.9473	1.0000
class 2	0.7982	0.9177	0.9657	0.8350	0.9174	0.9759	0.9013	0.7449	0.9589	0.9953
class 3	0.8198	0.8244	0.9757	0.8077	0.7441	0.9755	0.9271	0.4953	0.9393	0.9815
class 4	0.5228	0.9592	0.9681	0.9714	0.7500	0.9490	0.7476	0.4954	0.9393	0.9862
class 5	0.9395	0.9676	0.9832	0.9496	0.7556	0.9739	0.9445	0.6923	0.9570	0.9615
class 6	0.9956	0.9700	0.9571	0.8825	0.9307	0.9640	0.9501	0.8092	0.9200	0.9984
class 7	1.0000	1.0000	0.8064	0.8889	1.0000	0.8620	0.6086	0.480	1.0000	1.0000
class 8	0.9978	0.9260	0.9954	0.9504	0.9821	0.9954	0.9931	0.9614	1.0000	1.0000
class 9	0.8333	0.3333	1.0000	0.8461	0.6538	0.7727	0.9285	0.7647	1.0000	1.0000
class 10	0.7824	0.9406	0.9407	0.9101	0.8878	0.9055	0.9079	0.7172	0.9402	0.9724
class 11	0.9822	0.8453	0.9769	0.8060	0.9233	0.9838	0.9766	0.9423	0.9423	0.9846
class 12	0.8795	0.9205	0.9470	0.9212	0.6258	0.9312	0.8897	0.4158	0.8588	0.9477
class 13	1.0000	0.9831	0.9531	0.9150	0.7731	0.9104	0.8064	0.8961	0.9945	1.0000
class 14	0.9912	0.9288	0.9850	0.9437	0.9352	0.9816	0.9947	0.9238	0.9752	0.9911
class 15	0.7635	0.8794	0.9856	0.9634	0.8669	0.9853	0.6571	0.4797	0.9768	0.9855
class 16	0.8356	0.9836	0.7321	0.9333	0.9761	0.7663	0.45	0.2682	1.0000	1.0000
OA	0.8856	0.9007	0.9671	0.8700	0.8743	0.9639	0.9224	0.7279	0.9536	0.9847
AA	0.8838	0.8987	0.9482	0.9043	0.8444	0.9317	0.8334	0.6598	0.9588	0.9878
Kappa	0.8699	0.8859	0.9625	0.8500	0.8563	0.9588	0.9110	0.6885	0.9471	0.9826
Train Times (s)	8.26	419.16	133.00	299.81	81.73	192.6	173.88	89.63	76.03	103.24

Table 6. Classification results (%) of different models in the University of Pavia dataset.

Class	SVM	ResNet	SSRN	PyResNet	ContextualNet	A2S2KNet	PSE-UNet	SpectralFormer	GFNet	Proposed
class 1	0.9028	0.8456	0.9541	0.8074	0.8934	0.9920	0.9761	0.8802	0.9904	1.0000
class 2	0.9393	0.9295	0.9935	0.9533	0.9863	0.9988	0.9991	0.9728	0.9947	0.9991
class 3	1.0000	0.8304	0.9801	0.7810	0.74008	0.9287	0.6755	0.6928	0.9105	0.9647
class 4	0.9944	0.9975	0.9964	0.9830	0.9453	0.9938	0.9535	0.9232	0.9717	0.9857
class 5	0.9530	0.9802	0.9946	0.9795	0.9969	0.9984	0.9908	1.0000	1.0000	1.0000
class 6	0.9957	0.9261	0.9954	0.9720	0.9921	0.9935	0.9920	0.7460	0.9802	1.0000
class 7	1.0000	0.8305	0.9780	0.8438	0.8666	0.9852	0.6922	0.7372	0.9587	0.9976
class 8	0.7840	0.8009	0.9335	0.8163	0.8069	0.9197	0.8875	0.8913	0.9716	0.9827
class 9	0.9537	0.9940	0.9911	0.9729	0.8445	0.9956	0.6106	0.9563	0.9377	0.9924
OA	0.9291	0.9041	0.9814	0.9119	0.9362	0.9860	0.9476	0.9007	0.9824	0.9952
AA	0.9470	0.9039	0.9796	0.9010	0.8969	0.9784	0.8642	0.8666	0.9684	0.9914
Kappa	0.9047	0.8712	0.9753	0.8819	0.9152	0.9815	0.9302	0.8672	0.9767	0.9936
Train Times (s)	4.88	451.85	310.3	268.75	91.39	227.04	105.98	171.37	100.55	183.23

Table 7. Classification results (%) of different models in the WHU-Hi-LongKou dataset.

Class	SVM	ResNet	SSRN	PyResNet	ContextualNet	A2S2KNet	PSE-UNet	SpectralFormer	GFNet	Proposed
class 1	0.9620	0.9839	0.9980	0.8702	0.9686	0.9195	0.9905	0.9689	0.9853	0.9968
class 2	0.6561	0.6717	0.8526	0.5469	0.2566	0.7953	0.9385	0.3510	0.9384	0.9996
class 3	0.9201	0.5121	0.9684	0.5806	0.1686	0.9915	0.1779	0.2710	0.8290	0.9616
class 4	0.8958	0.7851	0.8984	0.7624	0.8673	0.9015	0.9850	0.9706	0.9853	0.9966
class 5	0.2790	0.8500	0.8154	0.6667	0.1710	0.9417	0.2540	0.7009	0.7304	0.7130
class 6	0.9502	0.9418	0.9950	0.9908	0.7822	0.9964	0.9023	0.8144	0.9667	0.9919
class 7	0.9989	0.7647	0.9820	0.7970	0.9985	0.9928	0.9999	0.9944	0.9992	0.9997
class 8	0.8495	0.9810	0.9106	1.0000	0.2934	0.8566	0.5941	0.65513	0.8943	0.8979
class 9	1.0000	0.4383	0.9184	0.4074	0.2272	0.9868	0.6852	0.1436	0.9035	0.9363
OA	0.9304	0.8115	0.9477	0.8008	0.8566	0.9375	0.9361	0.8917	0.9743	0.9861
AA	0.8346	0.7699	0.9265	0.7358	0.5259	0.9314	0.7253	0.6221	0.9148	0.9437
Kappa	0.9074	0.7379	0.9303	0.7221	0.8097	0.9167	0.9149	0.8560	0.9661	0.9817
Train Times (s)	3.88	118.8	76.67	72.24	34.60	107.44	58.34	41.34	21.09	41.32

The enhanced SVM performs significantly better than the original SVM. But, it suffers from an overfitting phenomenon during training with small samples, leading to accuracy degradation. ResNet uses only spatial features for analysis and employs PCA dimensionality reduction, which ignores the spectral properties between neighboring spectra and leads to poor classification. The SSRN is designed with continuous spectral and spatial residual layers to analyze the spectral correlation, overcoming the problems present in the 2DCNN and obtaining better classification results. However, the inherent characteristics of convolutional networks limit it, and it cannot analyze the sequence features of the spectrum. PyPesNet gathers features by enlarging the convolutional layer's feature map. However, its method needs to be able to perform well with a large number of training examples while producing subpar classification results with tiny samples. ContextualNet uses the local space-spectral vector of individual pixels to mine the spectral differences; however, employing fewer samples harms the classification accuracy. In order to learn spatial-spectrum spectral features, A2S2KResNet features a spectral attention mechanism. However, the classification performance is constrained by the structure's excessive complexity and processing time. The overall classification accuracy of PSE-UNet is high. However, in terms of the average accuracy, it is lower than other methods, indicating that it is not very effective for classifying certain classes. Furthermore, PSE-UNet uses the PCA to reduce the number of dimensions, which significantly reduces the number of parameters in the network. SpectralFormer is a framework based on a Vision Transformer (ViT) that mines different levels of spectral properties through group-wise spectral embedding and cross-layer adaptive fusion. However, because Vision Transformer networks require more training samples, training with fewer training samples leads to severe overfitting, so the classification accuracy is not substantial. By evaluating the spectral features in the frequency domain, the GFNet and our suggested method rethink the spatial-spectral properties and enhance the classification performance. Among these, GFNet is the original network, which lacks the targeted processing of spectral features. Since the proposed method has measures, such as separating frequency-domain filter networks to achieve effective spectral-space feature extraction and using detail enhancement to compensate for the drawbacks of frequency-domain networks and nonlinear modules to enhance frequency-domain spectral extraction, the most effective

framework is the one we have proposed for the SFFN. For the PU and WHU databases, as shown in Tables 6 and 7, our proposed SFFN achieves the best accuracy in both cases, and its classification performance is comparable to the prior ones.

3.4. Visual Evaluation

We visualized different methods to obtain classification maps for qualitative assessment. Figures 6–8 show the classification visualization graphs of the evaluation model for the IP, PU, and WHU databases, respectively. The false color map and the ground truth image are depicted in Figures 6a,b, 7a,b and 8a,b, respectively, whereas Figures 6–8 of (c) through (l) are produced using various deep learning algorithms. The SVM, ResNet, PyResNet, and ContextualNet may all be observed to produce higher levels of noise for classification with fewer training data, which indicates that these models are incapable of recognizing the categories of objects. In contrast, the SSRN, A2S2KResNet, SPectral Former, PSE-Unet, GFNet, and our proposed method show better visualization results due to the mining of more relationships between spectra. As an emerging network architecture, the frequency-domain analysis network can smooth the frequency-domain relationships from HS image spectral adjacencies and the edges by using a detail enhancement layer. We can produce perfect classification maps with fewer noise points than previous approaches with limited sample training, demonstrating a greater capacity to examine neighboring hyperspectral bands. With finer classification edges, the edges of the classified regions have better spatial continuity, demonstrating that our suggested method can accurately capture the variability of spatial information.

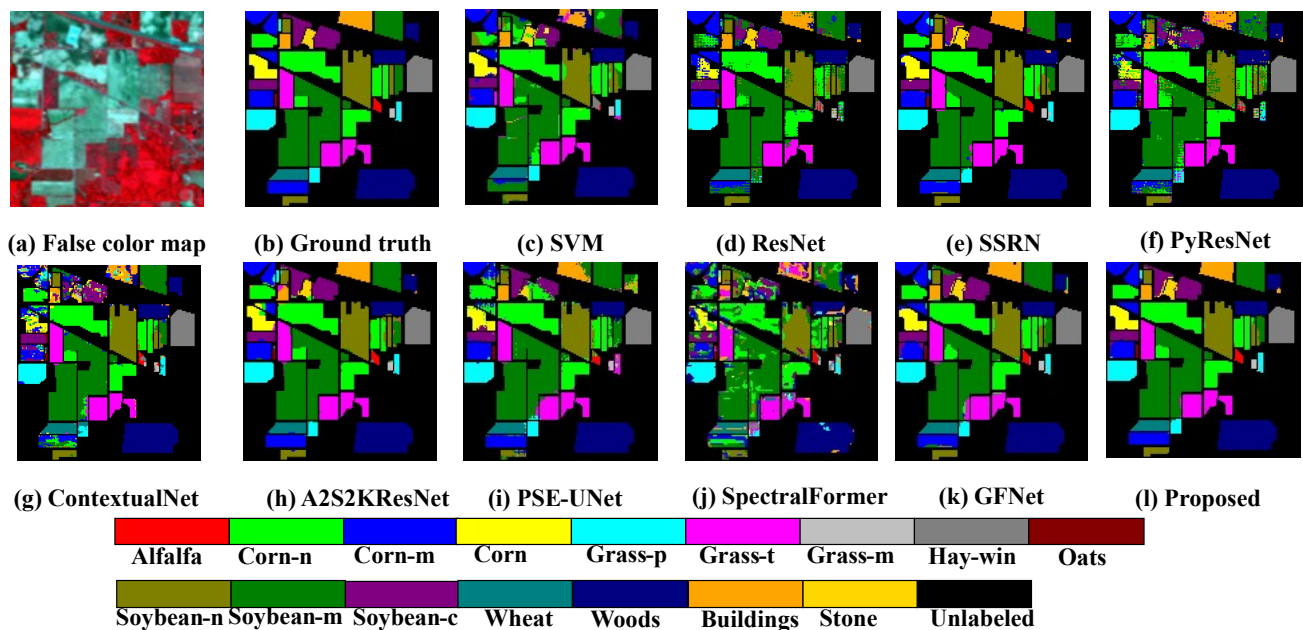


Figure 6. Classification maps of Indian Pines.

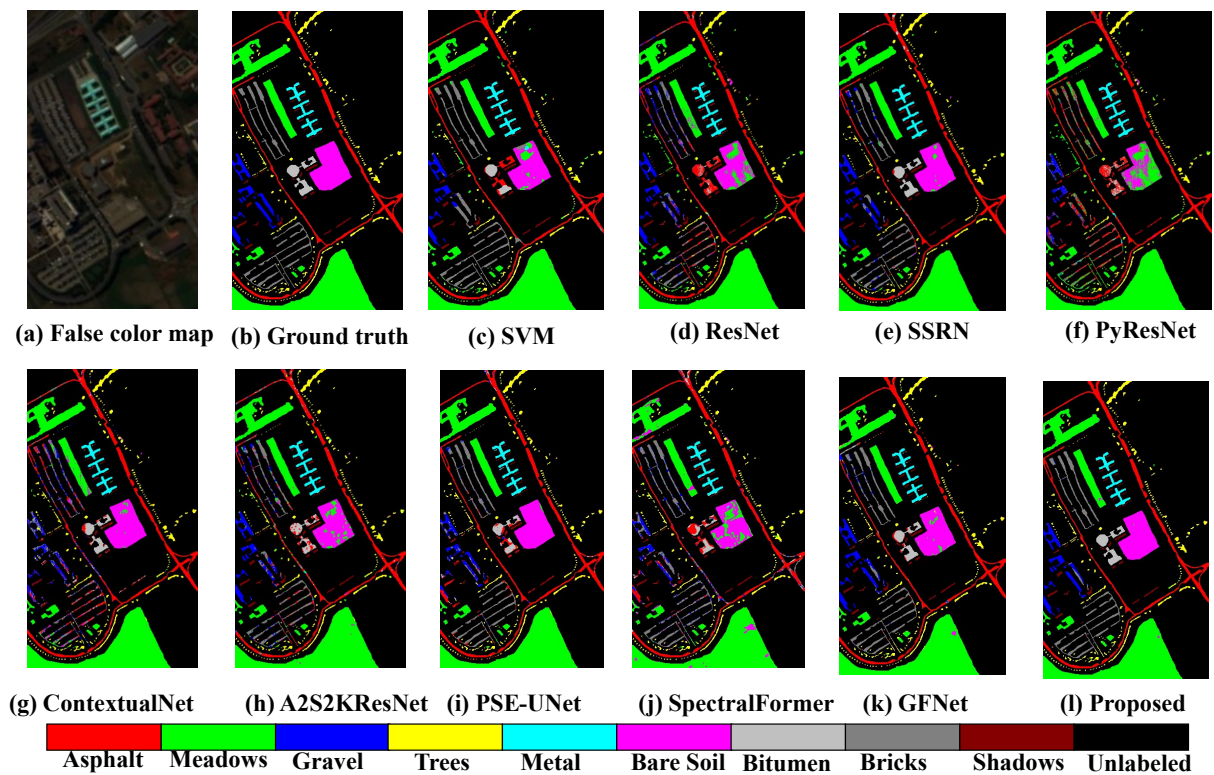


Figure 7. Classification maps of the University of Pavia.

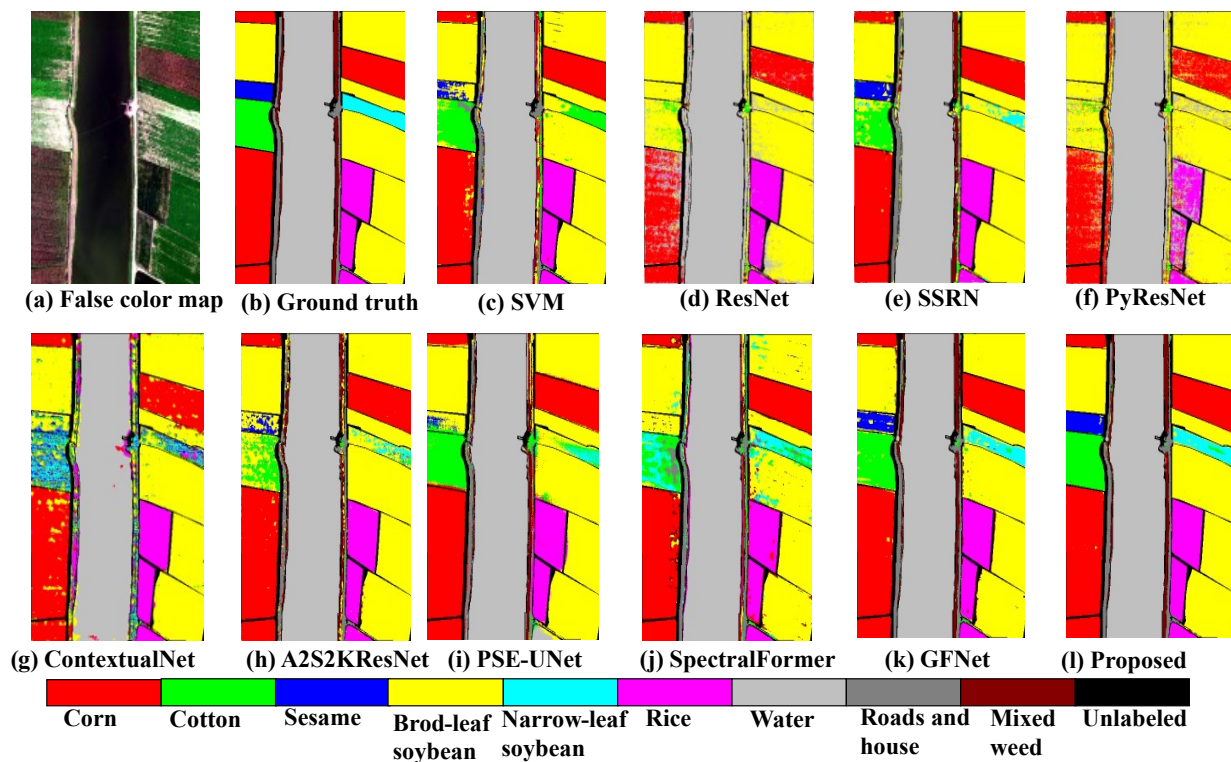


Figure 8. Classification maps of The WHI-HI-LongKou datasets.

4. Discussion

4.1. Ablation Study

Apart from the network's learnable parameters and the hyperparameters needed for training, the final classification performance is greatly influenced by the different

split-frequency filter ratios. Investigating the optimal separation filtering parameters is crucial. We look at the ablative sensitivity of this parameter for the Indian Pines dataset and how the classification parameters will change if only the ratio of the split parameter is applied. Table 8 shows the classification accuracy trend as the separation filtering changes by different proportions. We mark the optimal results in bold in Table 8. The first nine columns indicate the classification accuracy with different separation ratios, and the last column indicates the performance of the unimproved GFNet with the same parameter settings. The results show that the best accuracy occurs at a 65% separation ratio. However, the performance of the separated frequency-domain convolution method is better than that of the GFNet at different separation ratios (all have an OA of more than 1%). Therefore, we can analyze the local spectra of different segments to understand that separation-frequency-domain filtering exploits subtle spectral differences. The classification performance increases and drops as the separation ratio parameter changes, peaking at 70%. This model raises the top bound of the hyperspectral frequency-domain analysis, as shown by the fact that all three classification metrics are higher than those of GFNet.

Table 8. Ablation analysis of the split-frequency filter with different split ratios for the Indian Pines dataset.

The Split Ratio (%)	10	20	30	40	50	60	70	80	90	GFNet
OA	0.9700	0.9745	0.9763	0.9757	0.9779	0.9781	0.9847	0.9785	0.9698	0.9536
AA	0.9720	0.9774	0.9785	0.9787	0.9780	0.9795	0.9878	0.9825	0.9762	0.9588
Kappa	0.9658	0.9710	0.9729	0.9723	0.9748	0.9750	0.9826	0.9770	0.9655	0.9471

Additionally, by gradually installing several modules, we evaluate the development of the performance of each module for the SFFN network model. In order to confirm the suitability of each module in the SFFN model for applications requiring hyperspectral classification, we orchestrate experiments thorough ablation on the Indian Pines dataset. In Table 9, × means that this module is not added while ✓ means indicates that this module is added. As shown in Table 9, the classification result of the proposed split-frequency filter network without detail enhancement and the other modules is the lowest, and when the modules are added step-by-step, it can be noted that the detail-enhancement layer can enhance the model's overall reliability. Other improvements can make the classification accuracy better, and a better classification performance can be obtained.

Table 9. Ablation analysis of different modules and the model's performance gain for the Indian Pines dataset.

Method	Module			Metric			Times	
	Split-Frequency Filter	Detail-Enhancement Layer	Others	OA	AA	Kappa	Training Time (s)	Test Time (s)
GFNet	×	×	×	0.9536	0.9588	0.9471	76.03	1.57
SSFN	✓	×	×	0.9785	0.9825	0.9755	102.68	1.489
SSFN	✓	✓	×	0.9785	0.9849	0.9766	117.57	1.848
SSFN	✓	✓	✓	0.9847	0.9878	0.9826	103.24	1.037

4.2. Influence of the Size of the Image Patch

The influence of different spectral–spatial patch sizes on the efficacy of the classification is the main topic covered in this section. We mainly use the target spectrum and the surrounding spatial neighborhood as cube information for the network input, so the cube's size will significantly impact the classification performance. The network cannot thoroughly learn information in adjacent spatial domains when the spectral–spatial cube is too small, since it can only use a limited amount of spatial information. The receptive field also increases with an enormous spectral–spatial cube, contributing extraneous information and hindering network learning. As the size of the input space cube rises (see Figure 9), the OAs of the IP, PU, and WHI databases rise and subsequently fall, IP and PU reach a

maximum size of 9×9 with 98.47% and 99.01%, respectively. In comparison, WHI reaches a maximum size of 7×7 with 98.48%. By examining the experimental data from the three datasets, it was discovered that for IP and PU, the best performance is achieved with a spatial patch size of 9×9 , and the best performance on WHI is achieved with a size of 7×7 . As a result, for IP and PU, this research utilizes an input size of 9×9 ; for WHI, it applies a spatial input size of 7×7 . Additionally, we set the network's depth to 5. Overly deep networks might hamper the network performance in terms of classification.

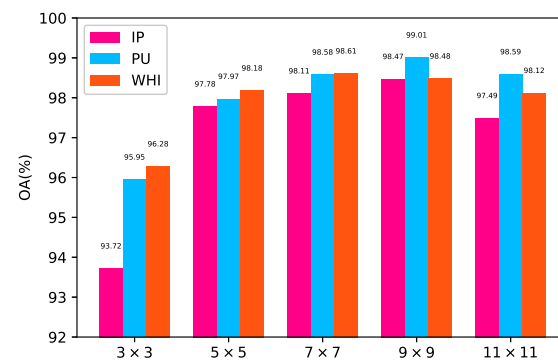


Figure 9. Overall accuracy (%) of input patches with different spectral–spatial sizes on the four datasets.

4.3. The Influence of a Spatial Disjoint Split

In this subsection, we focus on the effect of a spatial disjoint split on the test accuracy in the training and test sets. In [54], the effects of the random division of training and test sets, as well as the spatial disjoint split of the training and test sets on the training set are mentioned. However, the spatial disjoint split mentioned in the paper is not entirely disjointed. Therefore, we propose our method to verify the complete spatial disjointedness, as shown in Figure 10.

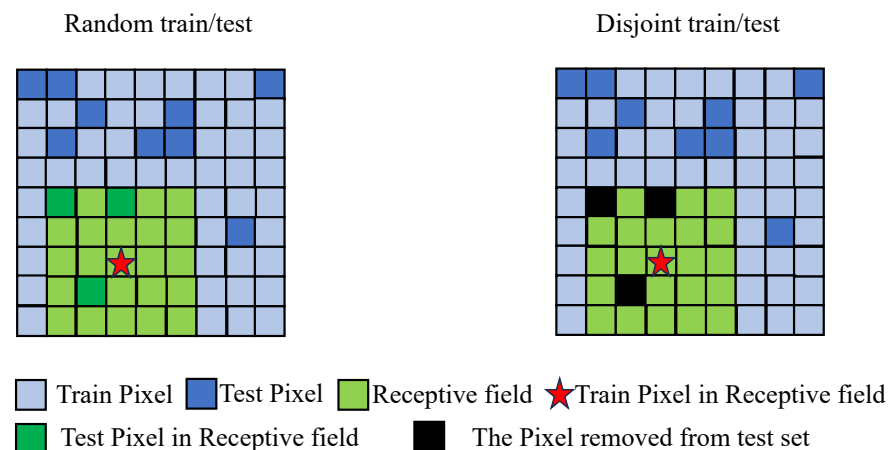


Figure 10. Illustration of a split that is random and spatially disjoint. The right figure depicts an entirely disjointed split, while the left figure displays a random selection.

We still use the same randomized division ratio as before, meaning that the training data do not change. The different thing from before is that we traverse all receptive fields in the training set and eliminate all contained test sets. As the test set shrinks, we reach a state of total spatial disjoint. Furthermore, we also perform a comparison test on the Pavia University data, as shown in Table 10.

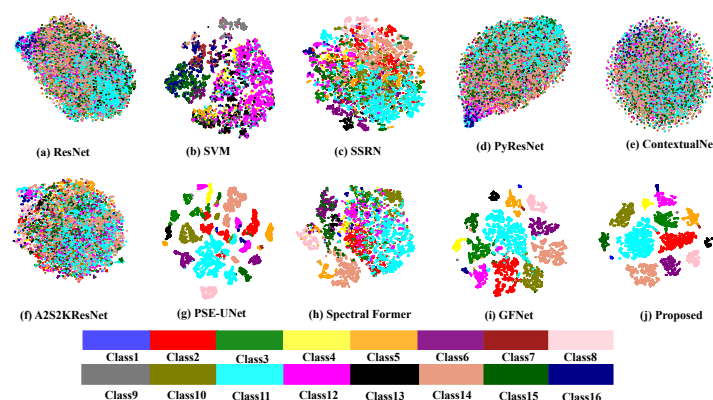
Table 10. Comparison of the accuracy of different methods under a random split and a spatially disjoint split for the University of Pavia dataset.

	Random			Disjoint		
	OA	AA	kappa	OA	AA	kappa
SVM	0.9291	0.9470	0.9047	0.9291	0.9470	0.9047
ResNet	0.8849	0.8816	0.8452	0.8081	0.8343	0.7502
SSRN	0.9626	0.9501	0.9504	0.9264	0.9024	0.9069
PyResNet	0.7972	0.8132	0.7183	0.7840	0.7909	0.7171
ContextualNet	0.8094	0.7572	0.7407	0.7603	0.7009	0.6916
A2S2KResNet	0.9413	0.9343	0.9216	0.9349	0.9313	0.9143
PSE-UNet	0.9224	0.8334	0.9110	0.8834	0.8163	0.8584
SpectralFormer	0.879	0.8647	0.8487	0.8156	0.7086	0.7613
GFNet	0.9663	0.9460	0.9551	0.9499	0.9329	0.9367
Proposed	0.9904	0.9820	0.9872	0.9843	0.9730	0.9801

SVM is a traditional machine learning algorithm that creates a spectral cube from a single point without using spatial neighborhood information during training. As a result, no points are removed from the test set, and both tests have identical results. For instance, algorithms like ResNet, ContextualNet, and PSE-UNet are more affected by total spatial disjointedness. They are more concerned with the spatial information from the neighborhood than the spectral information. However, methods like PyResNet, GFNet, and the proposed algorithm place greater emphasis on the spectrum's sequence information and, as a result, are more generalizable, even in the presence of complete spatial disjointedness. Comparing our algorithm's 1% OA degradation to that of other approaches demonstrates our algorithm's robustness. The drop in accuracy here may be due to the reduction in the test set involved in the evaluation. Whether in the case of random selection or the case of complete disjointedness, our proposed algorithm can achieve the best performance.

4.4. Visual Evaluation

To demonstrate the different feature classification abilities of different algorithms, we visualize the feature distribution of ten methods in two-dimensional space using the t-SNE algorithm [55] for the IP database. As seen in Figure 11, with the same t-SNE settings, the classification boundary of our proposed SFFN is more prominent, there is less overlapping of different classes, and the classification results can be seen more intuitively. Compared with other methods, such as ResNet and other algorithms, the clustering effect is poor and cannot reliably classify the same class with serious cross-over. Therefore, our proposed SFFN model can effectively learn the representative information from the spectral frequency-domain.

**Figure 11.** The feature representation capability of eight methods on the IP test set: t-SNE, (a) ResNet, (b) SVM, (c) SSRN, (d) PyResNet, (e) ContextualNet, and (f) A2S2KResNet, (g) PSE-UNet, and (h) Spectral Former, (i) GFNet, and (j) our proposed method.

5. Conclusions

In this paper, we validated the features of a split-frequency filter network for hyperspectral classification using intuitive experiments. The results show that hyperspectral frequency-domain networks can have a high classification accuracy and be used as a new backbone network. We proposed a new split-frequency filter network. In the proposed method, the spectral disparities between the hyperspectral spectral bands are taken advantage of and segmented for the frequency-domain analysis, which can significantly boost the accuracy of hyperspectral frequency classification. Our approach consists of three main components: a split-frequency filter network, a detail-enhancement layer, and a nonlinear function enhancement. Compared to the GFNet, the separated split-frequency filter network enhancement is the most obvious. Our experiments on three renowned hyperspectral datasets show that better classification results could be obtained than other recently proposed methods.

Future research will examine how to adaptively employ frequency-domain segmentation to fully utilize hyperspectral frequency-domain filtering networks. In order to increase the classification accuracy and model efficiency with fewer training samples, we will also enhance the model structure to fully exploit the spectrum frequency-domain characteristics of hyperspectral images.

Author Contributions: Conceptualization, J.G.; data curation, X.D. and F.L.; formal analysis, J.G.; methodology, J.G.; software, J.G.; validation, J.W. and Z.Y.; writing—original draft, J.G.; writing—review and editing, J.W., X.D., Z.Y. and F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The publicly available datasets IP and PU were analyzed in this study and can be found here: http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes, accessed on 1 May 2023. Moreover, a publicly available dataset, WHU-Hi-LongKou, was analyzed in this study and can be found here: http://rsidea.whu.edu.cn/resource_WHUHi_sharing.htm, accessed on 1 May 2023.

Acknowledgments: All authors would like to thank the Hyperspectral Image Analysis group and RSIDEA (Intelligent Data Extraction, Analysis and Application of Remote Sensing) at Wuhan University. We also thank the anonymous reviewers for their constructive comments and suggestions.

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

References

1. Camps-Valls, G.; Tuia, D.; Bruzzone, L.; Benediktsson, J.A. Advances in Hyperspectral Image Classification: Earth Monitoring with Statistical Learning Methods. *IEEE Signal Process. Mag.* **2014**, *31*, 45–54.
2. Pan, Z.; Wang, F.; Xia, L.; Wang, X. Feature Extraction for Urban Vegetation Stress Identification Using Hyperspectral Remote Sensing. In Proceedings of the 2nd International Conference on Information Science and Engineering, Hangzhou, China, 4–6 December 2010; IEEE: Hangzhou, China, 2010; pp. 250–254. [\[CrossRef\]](#)
3. Bansod, B.; Singh, R.; Thakur, R. Analysis of Water Quality Parameters by Hyperspectral Imaging in Ganges River. *Spat. Inf. Res.* **2018**, *26*, 203–211. [\[CrossRef\]](#)
4. Ghiyamat, A.; Shafri, H.Z.M. A Review on Hyperspectral Remote Sensing for Homogeneous and Heterogeneous Forest Biodiversity Assessment. *Int. J. Remote Sens.* **2010**, *31*, 1837–1856. [\[CrossRef\]](#)
5. Bárta, V.; Racek, F.; Krejčí, J. NATO Hyperspectral Measurement of Natural Background. In *Target and Background Signatures IV*; Stein, K.U., Schleijpen, R., Eds.; SPIE: Berlin, Germany, 2018; p. 4. [\[CrossRef\]](#)
6. Rasti, B.; Hong, D.; Hang, R.; Ghamisi, P.; Kang, X.; Chanussot, J.; Benediktsson, J.A. Feature Extraction for Hyperspectral Imagery: The Evolution From Shallow to Deep: Overview and Toolbox. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 60–88. [\[CrossRef\]](#)
7. Prasad, S.; Bruce, L.M. Limitations of Principal Components Analysis for Hyperspectral Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 625–629. [\[CrossRef\]](#)
8. Bandos, T.; Bruzzone, L.; Camps-Valls, G. Classification of Hyperspectral Images with Regularized Linear Discriminant Analysis. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 862–873. [\[CrossRef\]](#)
9. Villa, A.; Benediktsson, J.A.; Chanussot, J.; Jutten, C. Hyperspectral Image Classification with Independent Component Discriminant Analysis. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 4865–4876. [\[CrossRef\]](#)

10. Blanzieri, E.; Melgani, F. Nearest Neighbor Classification of Remote Sensing Images with the Maximal Margin Principle. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1804–1811. [\[CrossRef\]](#)
11. Melgani, F.; Bruzzone, L. Classification of Hyperspectral Remote Sensing Images with Support Vector Machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [\[CrossRef\]](#)
12. Gislason, P.O.; Benediktsson, J.A.; Sveinsson, J.R. Random Forests for Land Cover Classification. *Pattern Recognit. Lett.* **2006**, *27*, 294–300. [\[CrossRef\]](#)
13. Su, M.; Liu, C.; Di, D.; Le, T.; Sun, Y.; Li, J.; Lu, F.; Zhang, P.; Sohn, B.J. A Multi-Domain Compression Radiative Transfer Model for the Fengyun-4 Geosynchronous Interferometric Infrared Sounder (GIIRS). *Adv. Atmos. Sci.* **2023**, 1–15.
14. Liu, X.; Smith, W.L.; Zhou, D.K.; Larar, A. Principal component-based radiative transfer model for hyperspectral sensors: Theoretical concept. *Appl. Opt.* **2006**, *45*, 201–209. [\[PubMed\]](#)
15. Kang, X.; Li, S.; Benediktsson, J.A. Spectral-spatial hyperspectral image classification with edge-preserving filtering. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2666–2677. [\[CrossRef\]](#)
16. Zhong, S.; Chang, C.I.; Zhang, Y. Iterative edge preserving filtering approach to hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 90–94. [\[CrossRef\]](#)
17. Chen, Y.; Zhao, X.; Jia, X. Spectral—Spatial Classification of Hyperspectral Data Based on Deep Belief Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2381–2392. [\[CrossRef\]](#)
18. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [\[CrossRef\]](#)
19. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep Convolutional Neural Networks for Hyperspectral Image Classification. *J. Sens.* **2015**, *2015*, 258619. [\[CrossRef\]](#)
20. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5966–5978. [\[CrossRef\]](#)
21. Li, J.; Wang, H.; Zhang, A.; Liu, Y. Semantic segmentation of hyperspectral remote sensing images based on PSE-UNet model. *Sensors* **2022**, *22*, 9678. [\[CrossRef\]](#)
22. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking Hyperspectral Image Classification with Transformers. *arXiv* **2021**, arXiv:2107.02988.
23. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep Learning-Based Classification of Hyperspectral Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [\[CrossRef\]](#)
24. Zabalza, J.; Ren, J.; Zheng, J.; Zhao, H.; Qing, C.; Yang, Z.; Du, P.; Marshall, S. Novel Segmented Stacked Autoencoder for Effective Dimensionality Reduction and Feature Extraction in Hyperspectral Imaging. *Neurocomputing* **2016**, *185*, 1–10. [\[CrossRef\]](#)
25. Hang, R.; Liu, Q.; Hong, D.; Ghamisi, P. Cascaded Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5384–5394. [\[CrossRef\]](#)
26. Liu, Q.; Zhou, F.; Hang, R.; Yuan, X. Bidirectional-Convolutional LSTM Based Spectral-Spatial Feature Learning for Hyperspectral Image Classification. *Remote Sens.* **2017**, *9*, 1330. [\[CrossRef\]](#)
27. Zhou, F.; Hang, R.; Liu, Q.; Yuan, X. Hyperspectral Image Classification Using Spectral-Spatial LSTMs. *Neurocomputing* **2019**, *328*, 39–47. [\[CrossRef\]](#)
28. Zhao, W.; Du, S. Spectral—Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [\[CrossRef\]](#)
29. Lee, H.; Kwon, H. Going Deeper with Contextual CNN for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [\[CrossRef\]](#)
30. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D—2-D CNN Feature Hierarchy for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 277–281. [\[CrossRef\]](#)
31. Roy, S.K.; Manna, S.; Song, T.; Bruzzone, L. Attention-Based Adaptive Spectral—Spatial Kernel ResNet for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7831–7843. [\[CrossRef\]](#)
32. Zhang, C.; Wang, J.; Yao, K. Global Random Graph Convolution Network for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 2285. [\[CrossRef\]](#)
33. Liu, Q.; Xiao, L.; Yang, J.; Wei, Z. CNN-Enhanced Graph Convolutional Network with Pixel- and Superpixel-Level Feature Fusion for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 8657–8671. [\[CrossRef\]](#)
34. 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; Proceedings, Part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
35. Lin, M.; Jing, W.; Di, D.; Chen, G.; Song, H. Context-aware attentional graph U-Net for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [\[CrossRef\]](#)
36. Liu, Q.; Xiao, L.; Yang, J.; Wei, Z. Multilevel superpixel structured graph U-Nets for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–15. [\[CrossRef\]](#)
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
38. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.

39. He, J.; Zhao, L.; Yang, H.; Zhang, M.; Li, W. HSI-BERT: Hyperspectral Image Classification Using the Bidirectional Encoder Representation From Transformers. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 165–178. [\[CrossRef\]](#)
40. Liu, H.; Li, W.; Xia, X.G.; Zhang, M.; Gao, C.Z.; Tao, R. Central Attention Network for Hyperspectral Imagery Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–15. [\[CrossRef\]](#)
41. Rao, Y.; Zhao, W.; Zhu, Z.; Lu, J.; Zhou, J. Global Filter Networks for Image Classification. *arXiv* **2021**, arXiv:2107.00645.
42. Tolstikhin, I.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. MLP-Mixer: An All-MLP Architecture for Vision. *arXiv* **2021**, arXiv:2105.01601.
43. Brigham, E.O.; Morrow, R.E. The Fast Fourier Transform. *IEEE Spectr.* **1967**, *4*, 63–70. [\[CrossRef\]](#)
44. Chi, L.; Jiang, B.; Mu, Y. Fast Fourier Convolution. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 4479–4488.
45. Qin, Z.; Zhang, P.; Wu, F.; Li, X. FcaNet: Frequency Channel Attention Networks. *arXiv* **2021**, arXiv:2012.11879.
46. Misra, D. Mish: A Self Regularized Non-Monotonic Activation Function. *arXiv* **2020**, arXiv:1908.08681.
47. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
48. Hendrycks, D.; Gimpel, K. Gaussian Error Linear Units (GELUs). *arXiv* **2020**, arXiv:1606.08415.
49. Yu, W.; Si, C.; Zhou, P.; Luo, M.; Zhou, Y.; Feng, J.; Yan, S.; Wang, X. MetaFormer Baselines for Vision. *arXiv* **2022**, arXiv:2210.13452.
50. Hu, X.; Zhong, Y.; Luo, C.; Wang, X. WHU-Hi: UAV-Borne Hyperspectral with High Spatial Resolution (H2) Benchmark Datasets for Hyperspectral Image Classification. *arXiv* **2021**, arXiv:2012.13920. [\[CrossRef\]](#)
51. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Las Vegas, NV, USA, 2016; pp. 770–778. [\[CrossRef\]](#)
52. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral—Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 847–858. [\[CrossRef\]](#)
53. Paoletti, M.E.; Haut, J.M.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.J.; Pla, F. Deep Pyramidal Residual Networks for Spectral—Spatial Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 740–754. [\[CrossRef\]](#)
54. Audebert, N.; Le Saux, B.; Lefèvre, S. Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 159–173. [\[CrossRef\]](#)
55. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

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.