remote sensing

MDPI

Article

Lightweight-VGG: A Fast Deep Learning Architecture Based on Dimensionality Reduction and Nonlinear Enhancement for Hyperspectral Image Classification

Xuan Fei [1,2,3], Sijia Wu [3], Jianyu Miao [3], Guicai Wang [3] and Le Sun [4,*]

[1] Key Laboratory of Grain Information Processing and Control, Henan University of Technology, Ministry of Education, Zhengzhou 450001, China; feixuan@haut.edu.cn
[2] Henan Key Laboratory of Grain Photoelectric Detection and Control, Henan University of Technology, Zhengzhou 450001, China
[3] School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou 450001, China; wusijia@stu.haut.edu.cn (S.W.); jymiao@haut.edu.cn (J.M.); wangguicai@haut.edu.cn (G.W.)
[4] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing 210044, China
* Correspondence: sunlecncom@nuist.edu.cn

Abstract: In the past decade, deep learning methods have proven to be highly effective in the classification of hyperspectral images (HSI), consistently outperforming traditional approaches. However, the large number of spectral bands in HSI data can lead to interference during the learning process. To address this issue, dimensionality reduction techniques can be employed to minimize data redundancy and improve HSI classification performance. Hence, we have developed an efficient lightweight learning framework consisting of two main components. Firstly, we utilized band selection and principal component analysis to reduce the dimensionality of HSI data, thereby reducing redundancy while retaining essential features. Subsequently, the pre-processed data was input into a modified VGG-based learning network for HSI classification. This method incorporates an improved dynamic activation function for the multi-layer perceptron to enhance non-linearity, and reduces the number of nodes in the fully connected layers of the original VGG architecture to improve speed while maintaining accuracy. This modified network structure, referred to as lightweight-VGG (LVGG), was specifically designed for HSI classification. Comprehensive experiments conducted on three publicly available HSI datasets consistently demonstrated that the LVGG method exhibited similar or better performance compared to other typical methods in the field of HSI classification. Our approach not only addresses the challenge of interference in deep learning methods for HSI classification, but also offers a lightweight and efficient solution for achieving high classification accuracy.

Keywords: hyperspectral image classification; dimensionality reduction; multi-layer perceptron; lightweight network

Citation: Fei, X.; Wu, S.; Miao, J.; Wang, G.; Sun, L. Lightweight-VGG: A Fast Deep Learning Architecture Based on Dimensionality Reduction and Nonlinear Enhancement for Hyperspectral Image Classification. *Remote Sens.* **2024**, *16*, 259. https://doi.org/10.3390/rs16020259

Academic Editors: Gemine Vivone, Junjun Jiang, Bihan Wen, Jiayi Ma, Leyuan Fang and Kui Jiang

Received: 13 November 2023
Revised: 5 January 2024
Accepted: 8 January 2024
Published: 9 January 2024

1. Introduction

As a potent remote sensing technique, hyperspectral imaging might capture high-dimensional data about the Earth's surface or other objects [1–3]. Unlike traditional remote sensing, which captures only a few broad spectral bands, hyperspectral imaging measures the reflectance or emission of light by spanning hundreds of contiguous and narrow spectral bands. These bands come from the shortwave-infrared, near-infrared, and visible sections of the electromagnetic spectrum. The resulting hyperspectral images (HSIs) contain plenty of information about the composition of materials on the surface that can be used in diverse applications, such as vegetation research, ocean exploration, mineral exploration, and disaster response. By analyzing the spectral signatures of various materials, hyperspectral imaging can be used to assist in the identification and mapping of different land cover types,

such as forests, wetlands, urban areas, impervious (nonporous) surfaces, and agricultural fields [4–7].

In recent decades, many methods relying on artificial features have been developed [8–12]. However, artificial feature extraction is often constrained by domain-specific knowledge and expertise, making it challenging to adapt to different datasets and applications, and it lacks sufficient adaptability [13].

Unlike traditional images, remote sensing image data typically contains more dimensions. With the increase in data dimensionality, classifier performance often shows an initial improvement, because the added dimensions provide more information, enabling the classifier to better distinguish between different categories. However, as dimensionality continues to increase, performance eventually reaches a saturation point, where classification accuracy no longer improves and may even slightly decrease. This suggests that adding dimensions no longer provides useful information after, and may actually introduce noise or redundancy; this is referred to as the Hughes Phenomenon [14]. Therefore, dimensionality reduction is necessary to mitigate the effects of the dimensionality curse.

For remote sensing images, several dimensionality reduction methods have been presented, which can be broadly classified into two categories: feature extraction [15] and band selection [16].

Representative feature extraction methods include principal component analysis (PCA), independent component analysis, and linear discriminant analysis [17,18], among others. With the advancement of computer technology, supervised classification methods have evolved in sophistication. Supervised classification utilizes labeled sample data to train classification models or algorithms, such as support vector machines (SVM) [19] and random forests [20]. These models and algorithms enable the automatic classification of different land cover categories, and also have achieved significant improvements in accuracy and efficiency.

The rise of deep learning methods signifies a significant advancement in remote sensing image classification. Deep learning methods, particularly convolutional neural networks (CNNs) [21–25], exhibit excellent performance in image classification. Notably, they have the ability to automatically extract high-level features from remote sensing images without the need for complex feature engineering. This results in increased accuracy in remote sensing image classification. Furthermore, there are also methods that combine transformers and CNNs for HSI classification [26–31]. Deep learning models, especially CNNS, can automatically learn features from raw data. This means that there is no need for manual feature engineering, and that the model itself can discover the most important features. In contrast, SVM typically requires manual feature selection and extraction [32].

Band selection (BS) can help remove redundant spectral bands, thereby reducing the dimensionality of the HSI data. Furthermore, this can decrease computational costs, accelerate model training speed, and mitigate the risk of overfitting. Feature extraction also allows the model to automatically extract advanced features, while BS assists in selecting spectral bands relevant to the task. By integrating these two approaches, it is possible to achieve comprehensive use of the rich information in the HSI data. This includes various spectral features of the original data, as well as domain knowledge related to the task. This helps improve the model's representational capacity and performance.

Harnessing the capabilities of these recent advances in technology, we propose a novel lightweight network architecture for HSI classification based on Visual Geometry Group (VGG) networks [33] and the incorporation of dimensionality reduction and multi-layer perceptron (MLP). In this method, we performed band selection to eliminate duplicate or redundant bands. Prioritizing band selection allowed us to identify the most relevant bands for classification. Furthermore, by using PCA dimensionality reduction, not only was the impact of noise reduced, but the amount of training data was reduced as well. These data preprocessing also improved the efficiency of subsequent network training and increased classification accuracy.

The contributions of this paper can be summarized as follows:

1. We simultaneously used band selection and whitened PCA to relieve the impact of random seeds on experimental results;
2. We propose a novel lightweight-VGG (LVGG) network architecture for HSI classification, which aims to maintain high performance while reducing the complexity and computational resource requirements of the network, enabling efficient HSI classification even in resource-constrained environments;
3. Our proposed method achieved better accuracy in three publicly available HSI datasets as compared to several other existing methods.

The remainder of this paper is organized as follows: Section 2 describes some related data preprocessing studies. In Section 3, the proposed LVGG network is introduced in detail. Section 4 introduces three publicly available HSI datasets and divides them into three parts (i.e., training, validation, and test sets). Section 5 analyzes comparative experimental results on three HSI datasets, and demonstrates the effectiveness of our method. Finally, some conclusions and discussions are provided in Section 6.

2. Related Work

2.1. Patch-Based Classification

In recent years, deep learning has experienced significant growth, and has had a profound impact on multiple domains, particularly in the field of hyperspectral image (HSI) classification, which is relevant to many deep learning-based models reaching widespread application. Models, such as autoencoders [34] and recurrent neural networks [35], have been extensively employed. Previously to this, CNNs, known for their local receptive fields and translational invariance, were most effective at feature extraction, thereby enhancing classification accuracy.

In HSI classification, various classification frameworks and architectures have been proposed to acquire spatial features and train classifiers. These frameworks often involve generating patches centered around sample pixels from an original image, which are then employed for feature extraction and classification using different network architectures. Some studies have introduced end-to-end networks that take 3D patches as inputs and produce specific labels for each patch using the last fully connected (FC) layer [36]. Other approaches have employed 2D CNNs with 1×1 convolution kernels and global average pooling to extract spectral information and prevent overfitting [37]. Neural networks with band-adaptive, spectral–spatial feature learning have also been proposed to solve the dimensionality curse and the spatial variability problems of spectral signatures. Additionally, deeper and wider networks with residual learning have been suggested, such as the contextual deep CNN, which utilizes multi-scale filter banks to simultaneously process spectral–spatial information [38].

To improve the aggregation of spectral–spatial information, researchers have introduced two-stream CNN-based architectures [39,40]. Before fusing the outputs for classification, these architectural designs employ 2D CNNs and other algorithms for the extraction of spatial and spectral information, respectively. Another category of spectral–spatial-based CNN architecture uses 3D CNNs to extract joint spectral–spatial features for HSI classification [36,41]. The spectral–spatial residual network (SSRN) [42], alternatively, utilizes residual blocks to learn spectral and spatial information in hyperspectral images. This enables the model to enhance its ability to identify different land cover categories and features. A fast and dense spectral–spatial convolution architecture has also been proposed [20], which uses different scales of 3D convolution and residual structures to learn spectral and spatial information. Another approach employs a hybrid convolution network that utilizes 3D convolution followed by 2D convolution to learn spectral–spatial features [21].

In recent years, various attention mechanisms have been integrated to enhance classification performance in the field of HSI classification [43]. These attention mechanisms enable models to dynamically focus on the most relevant and information-rich parts of an image, thereby improving classification accuracy [44]. One common type of attention

module is the Squeeze-and-Excitation module [45], which generates channel attention vectors through global pooling and fully connected layers; these vectors are then used to recalibrate the responses of spectral features. Additionally, many spatial attention modules and channel attention modules are used in convolutional networks to enhance spectral and spatial features [46,47]. Furthermore, a double-branch dual-attention (DBDA) mechanism network is employed, incorporating both position attention modules and channel attention modules [24] to enhance the feature representations of remote sensing images. Thus, the above attention mechanisms have significantly improved the performance of remote sensing image classification methods and have enabled models to better adapt to the characteristics of different images and scenes.

In stark contrast to CNNs, transformers segment images into non-overlapping blocks, often referred to as tokens, and then pass these blocks as input sequences to the model. Transformers use attention mechanisms to learn the relationships between these blocks, and thus primarily model global information. As a result, they require substantial amounts of data for training. Recently, transformer models have also been employed for HSI classification and achieved good results. For instance, an improved transformer-based, spatial–spectral feature extraction method has previously been proposed as a means to obtain features [48]. Another study used a partial partition restore module to introduce a novel local transformer [27]. A spectral–spatial feature tokenization transformer (SSFTT) has also been developed to excavate spectral–spatial features and high-level semantic features [28].

As the vision transformer has gained popularity in computer vision, researchers have begun to explore its potential for HSI classification. However, compared to its application in natural image processing, the transformer model is not always as effective for remote sensing image classification. Some reasons for this include: (1) single viewpoints; remote sensing images are mostly captured from a top-down perspective, while natural images exhibit diverse viewpoint. (2) Limited background and objects; remote sensing images typically consist of specific land cover types, with the spatial distribution of objects often influenced by geographical information. Thus, neighboring regions may not have significant correlations. As a result, capturing global features may not be as meaningful, and local features become more crucial. Therefore, transformers may not offer substantial improvements over CNNs, and may even be less effective than CNNs in this context.

2.2. Dimensionality Reduction

Feature extraction and band selection are two commonly employed techniques for reducing dimensionality in hyperspectral data processing. The primary objective of these methods is to enhance data analysis and processing efficiency by reducing overall data dimensionality [49].

Feature extraction is a fundamental process in data analysis and machine learning. It involves selecting and transforming relevant information from raw data to create a more compact and informative representation, making it easier for algorithms to process and analyze the data. Feature extraction is used to reduce the dimensionality of data while retaining essential information. It simplifies data, making it more manageable and often improving the performance of machine learning algorithms. This transformation helps to more effectively capture the relevant characteristics present in the data. Common feature extraction methods include PCA and linear discriminant analysis, among others. These methods employ mathematical transformations to identify the most significant features within the data, thereby facilitating improved classification or analysis [50].

Band selection plays a pivotal role in remote sensing image processing, enabling the selection and extraction of the most informative spectral bands, rendering remote sensing data more amenable to analysis and interpretation. This process helps reduce computational complexity, improve classification accuracy, decrease storage requirements, and better cater to the specific needs of remote sensing applications. Optimal band selection enhances the richness of information in remote sensing imagery, offering critical support for precise

land cover classification, monitoring, and analysis. Common band selection methods include ranking-based approaches, such as correlation coefficients and information gain, as well as clustering-based methods. Ranking methods sort bands based on their correlation or information content and select the top-ranked bands, while clustering methods group similar bands together and then select representative bands from each group for subsequent processing [51].

In a previous study, an improved, fast peak-based clustering method was introduced, which takes into account both intra-cluster distance and local density to rank bands. In the literature [52], an adaptive and distance-based band hierarchy clustering method was introduced by using the same ranking approach. Additionally, a fast neighborhood grouping band selection method [53] was proposed, which employs a coarse-to-fine grouping strategy to reduce redundancy, and then utilizes the product of information entropy and local density to rank bands.

Here, we introduce a preprocessing method for dimensionality reduction based on feature extraction and band selection, followed by the application of a deep learning network for HSI classification. First, a combined feature extraction and band selection approach was designed to extract valuable information for dimensionality reduction. An improved, lightweight-VGG architecture is then proposed for classification.

3. Proposed Method

The workflow of the proposed lightweight-VGG method is depicted in Figure 1 and encompasses four primary stages. In this method, dimensionality reduction preprocessing is performed on the original HSI data. Next, an expand convolution is introduced to increase the network's capacity, enabling the network to capture more data features, thus enhancing its performance. The dimension-enhanced data is then fed into a DyVGG block for patch-level feature extraction, and the extracted features are inputted into MLP and linear classifier layers to obtain the predicted labels of the sample data.

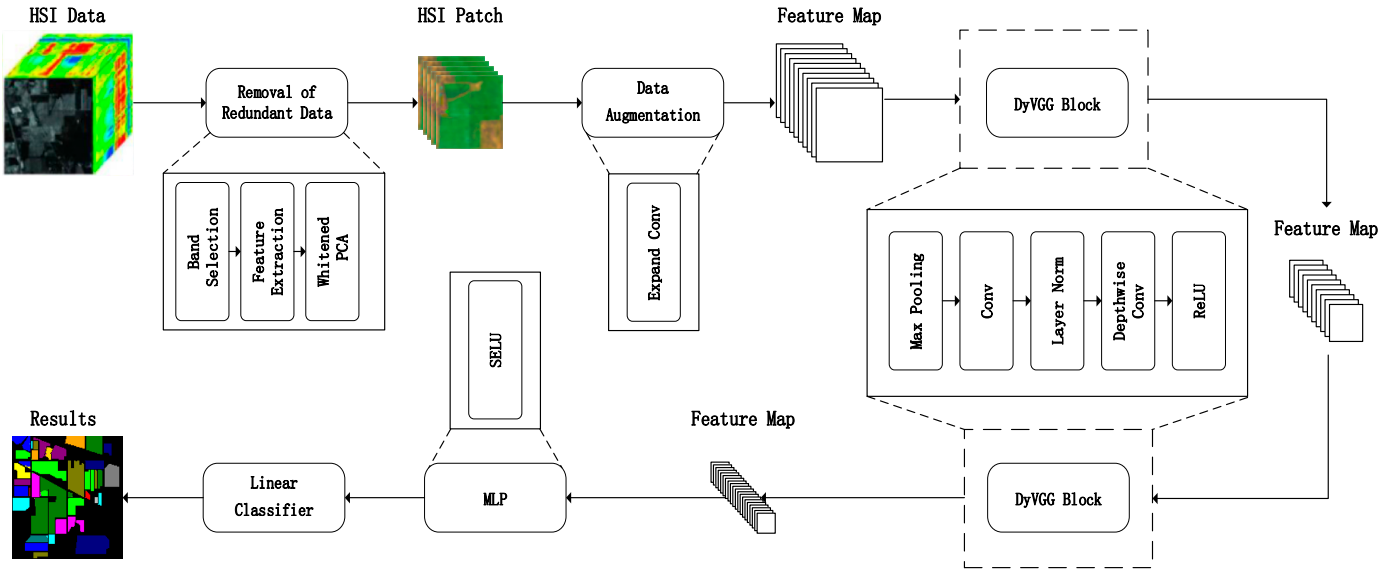

Figure 1. Overview of the proposed LVGG framework, which consists of two main components: a strategy for removing redundant data and a strategy for designing a lightweight VGG-based network structure.

3.1. The Removal of Redundant Data

In HSI preprocessing, band selection is an effective way of filtering out redundant bands while retaining important information relevant to the target task. Additionally, PCA is usually used to further reduce the impact of noise and extract key feature information.

Hence, we propose a redundant data removal method that integrates band selection and PCA.

3.1.1. Band Selection

This passage describes a process for grouping neighboring bands in a dataset represented by X. The bands in X, denoted as C bands, are initially divided into K equal groups. Each group, labeled X_k (where $k = 1, \ldots, K$), is defined as follows:

$$X_k = \begin{cases} \left\{ X^{(k-1)[\frac{C}{K}]+1}, \ldots\ldots, X^{k[\frac{C}{K}]} \right\} k[C/K] \leq C \\ \left\{ X^{(k-1)[\frac{C}{K}]+1}, \ldots\ldots, X^{C} \right\} k[C/K] > C \end{cases} \quad (1)$$

where X^i represents the i-th band image of X and $[C/K]$ represents the smallest integer greater than or equal to C/K.

Next, a fine partition algorithm [53] is applied to each of the initial band groups X_k ($k = 1, \ldots, K$) to create new band groups $X_k\prime$ ($k = 1, \ldots, K$). In this new partition, the number of bands may not be the same for different groups, and it is thus designed to group highly correlated spectral bands together, thereby resulting in lower correlation between the different band groups. For simplicity in this description, the new band groups are still marked with X_k ($k = 1, \ldots, K$), so the final grouping representation of X is given by:

$$X = \{X_1, X_2, \ldots\ldots, X_K\} \quad (2)$$

These highly correlated spectral bands are grouped together, resulting in lower correlation between band groups. Therefore, selecting representative bands from each band group becomes a more reasonable approach. For each band group, we utilize correlation coefficients to find a representative band that is most relevant to other bands in its group. The above process is described in detail as follows:

Step 1: Calculation of Pearson correlation coefficient. For band x and band y, the correlation coefficient r_{xy} can be calculated using the following formula:

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \overline{y})^2}} \quad (3)$$

where N represents the number of samples, that is, the number of spatial pixels.

Step 2: Construction of a correlation coefficient matrix. Assuming the number of bands is C_k for band group X_k, then the correlation coefficient matrix R_k (with a size of $C_k \times C_k$) is constructed as follows:

$$R_k = \left[r_{xy}^k \right]_{C_k \times C_k} \quad (4)$$

where the element r_{xy}^k comes from Formula (3).

Step 3: Select the most representative spectral band. For band group X_k, we sum the correlation coefficient matrix R_k by column, resulting in a vector S_k with a size of $1 \times C_k$:

$$S_k = \left[\sum_{x=1}^{C_k} r_{xy}^k \right]_{1 \times C_k} \quad (5)$$

Then, it is easy to find the maximum value in vector S_k and locate the index of its column:

$$max\_sum\_index = \underset{y}{argmax} \sum_{x=1}^{C_k} r_{xy}^k \quad (6)$$

Hence, the band corresponding to the index max_sum_index is the most representative spectral band of band group X_k.

By selecting the most representative band from each group, a spectral band set Y with a dimension of K is formed:

$$Y = \left\{ Y^1, Y^2, \ldots, Y^K \right\} \tag{7}$$

where Y^k represents the chosen band from the k-th band group.

3.1.2. Whitened PCA

In order to further reduce HSI data dimensionality and eliminate the noise effect in downstream classification, the PCA technique is employed to handle the selected band set Y; this preserves most of the information from the original dataset to ensure accuracy in the subsequent training.

Whitened PCA, an extension of PCA, not only reduces data dimensionality, but also eliminates data inter-correlations through linear transformations. Subsequently, whitened data is partitioned into smaller blocks or patches, each of which is centered around a pixel; the label of the central pixel serves as the ground truth label for the patch. To ensure accurate patch extraction, mirror padding is applied to the HSI data. Mirror padding duplicates data values at the boundaries, creating a seamless extension of the original HSI data. Eventually, these patches are inputted into the proposed lightweight-VGG network for the training process.

3.2. Expand Convolution

The proposed method uses $1 \times 1 \times 128$ convolutional kernels to expand the data after dimensionality reduction. This is because the reduced-dimensional data contains too few channels, making it challenging for the network to effectively fit the data. By using more convolutional kernels, features from the reduced dimensional data can be extracted more comprehensively, enabling better learning of critical information within the data. Additionally, this approach enhances the model's generalization capabilities, ultimately leading to improved classification performance.

Simultaneously increasing the dimensionality can lower the impact of the activation function on the results. If the current activation space has a high degree of integrity for the manifold of interest, passing through the rectified linear unit (ReLU) activation function can cause the activation space to collapse, inevitably resulting in information loss.

3.3. DyVGG Block

The primary characteristic of the previous VGG network is its use of stacked 3×3 convolutional layers and 2×2 max-pooling layers to create a deep neural network. Using 3×3 convolution offers fast computation and a streamlined single-path architecture, while also offering good memory efficiency compared to other structures (such as ResNet's shortcuts, which do not add to computational load but require double the GPU memory). Additionally, the use of 2×2 max-pooling layers effectively reduces the overall number of parameters.

Due to the depth and larger model size of the VGG network, it contains a significant number of parameters, requiring more computational resources and storage space. This can potentially limit its applicability in resource-constrained environments. Additionally, the large number of parameters in the VGG network results in relatively long training times.

To address these above issues, we have introduced a DyVGG block instead of the traditional VGG network, which includes three key points: reduction of network depth, replacement of batch normalization (BN), and use of depthwise convolution.

Point 1: reduction of network depth. We significantly reduced the number of convolutional layers, relying primarily on two sets of 3×3 convolutions and two rounds of max-pooling for feature learning. Max-pooling with a stride of 2 is also employed. Max-pooling preserves the maximum value within each pooling window, reducing image resolution and consequently decreasing the number of parameters. This helps alleviate the computational burden on the model, reduces the risk of overfitting, and enhances the

network's generalization capability. Our network structure has 11 fewer CNN layers and 3 fewer max-pool layers than the VGG-16 network.

Point 2: replacement of batch normalization. In this method, layer normalization (LN) is used instead of batch normalization. Following the optimization strategy from ConvNeXt [54], LN is employed after the convolutional layer. The advantages of layer normalization compared to batch normalization include: (1) insensitivity to small batch sizes; LN's performance is less affected by small batch sizes, making it effective even when dealing with small batches. (2) Lower parameter count; LN generally requires fewer parameters, resulting in more efficient model sizes. (3) No additional computation overhead; BN necessitates the calculation of batch-wise means and variances, while LN avoids these additional computations, leading to lower computational overhead.

Point 3: use of depthwise convolution. For enhancing the flexibility of the model, we improved the activation function, named DyReLU. After convolution, data passes through a 1×1 depthwise convolution [55] before application of the activation function. DyReLU can be formulated as:

$$\mathrm{DyReLU}(x) = \mathrm{ReLU}(a_i x + b_i) \tag{8}$$

where x represents the input, a_i represents the 1×1 depthwise learned parameters, and b_i represents the bias associated with these parameters. Additionally, as the network architecture is relatively shallow, residual structures are not used, which also helps reduce computation time.

3.4. Multi-Layer Perceptron

After the processing of the DyVGG block, the extracted features are flattened and inputted into the FC layers. The primary issue with the original VGG network is the excessive number of neurons used in the FC layers, which leads to a large overall parameter count and slow computation. Our approach significantly optimizes the total number of neurons. The FC layer in VGG16 has 4096 neurons, whereas our approach reduces it to 128, resulting in a substantial reduction while maintaining high classification accuracy. This operation also significantly enhances computational speed.

Furthermore, the activation function is switched from ReLU to scaled exponential linear unit (SELU) [56] in the fully connected layers, which is defined as follows:

$$selu(x) = \lambda \begin{cases} x \; if \; x > 0 \\ \alpha e^x - \alpha \; if \; x < 0 \end{cases} \tag{9}$$

where x is the input, α and λ ($\lambda > 1$) are hyperparameters, and e denotes the exponent.

SELU promotes the self-normalization of hidden layer activations, addressing vanishing/exploding gradient issues common in deep neural networks. It also maintains a stable mean and variance of activations, improving generalization and potentially eliminating the need for additional normalization techniques.

4. Datasets

In this section, we first introduce three commonly used hyperspectral image (HSI) datasets, including the Indian Pines (IP) dataset, Pavia University (PU) dataset, and Salinas (SA) datasets. Then, each dataset is divided into training, validation, and test sets for subsequent experiments.

4.1. Indian Pines Dataset

The IP dataset was acquired from airborne remote sensing data from the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the city of West Lafayette, Indiana, USA. This data is primarily used for research in agricultural fields, vegetation, and land use classification. The dataset consists of a set of grayscale images with a spatial resolution of 145×145, where each image corresponds to a hyperspectral band. The spectral range spans from 400 to 2500 nanometers, encompassing information from visible to infrared

spectra. After eliminating noisy bands, the dataset includes 200 bands for classification (1–103, 109–149, 164–219). The dataset includes 16 distinct land cover categories, including various types of crops, roads, buildings, and natural vegetation. In total, there are 10,249 labeled pixels, 10% of which were used for training, 10% for validation, and the remaining 80% for testing. The class name and the number of training and test samples are listed in Table 1.

Table 1. The class names and the training, validation and test sample numbers in IP.

Class	Land Cover Type	Train.	Val.	Test.
1	Alfalfa	5	5	36
2	Corn-No till	143	143	1142
3	Corn-Min till	83	83	664
4	Corn	24	24	189
5	Grass-Pasture	48	48	387
6	Grass-Trees	73	73	584
7	Grass-Pasture-Mowed	3	3	22
8	Hay-Windrowed	48	48	382
9	Oats	2	2	16
10	Soybean-No till	97	97	778
11	Soybean-Min till	245	245	1965
12	Soybean-Clean	59	59	475
13	Wheat	20	20	165
14	Woods	126	126	1013
15	Buildings-Grass-Trees-Drives	39	39	308
16	Stone-Steel-Towers	9	9	75
	Total	1024	1024	8201

4.2. Pavia University Dataset

The PU dataset was collected in 2003 using the Reflective Optics System Imaging Spectrometer (ROSIS-3) sensor over a part of the city of Pavia, Italy. The dataset consists of a set of grayscale images with spatial resolution 610×340, where each image corresponds to a hyperspectral band. The spectral range of these bands spans from 430 to 860 nanometers. Initially, the dataset contained 115 spectral bands, but during the data preprocessing phase, 12 bands affected by noise were removed, leaving a total of 103 bands for classification. The dataset includes nine distinct land cover categories, such as asphalt, meadows, and gravel, representing various land cover types in the city and its surrounding areas. In total, there are 42,776 labeled pixels. For machine learning and image classification evaluations, the dataset is typically split into training and test sets. In this case, 5% of the labeled samples are used for training, another 5% for validation, and the remaining samples for testing. The class name and the number of training and test samples are listed in Table 2.

Table 2. The class names and the training, validation and test sample numbers in PU.

Class	Land Cover Type	Train.	Val.	Test.
1	Asphalt	332	332	5967
2	Meadows	932	932	16,785
3	Gravel	105	105	1889
4	Trees	153	153	2758
5	Painted Metal Sheets	67	67	1211
6	Bare Soil	251	251	4527
7	Bitumen	67	67	1196
8	Self-Blocking Bricks	184	184	3314
9	Shadows	47	47	853
	Total	2138	2138	38,500

4.3. Salinas Dataset

The SA dataset is a hyperspectral dataset collected by the 224-band AVIRIS sensor over Salinas Valley, California. The dataset consists of 224 spectral band grayscale images with spatial resolution of 512 × 217. These spectral bands cover a wavelength range of 0.36 µm to 2.5 µm. During the data preprocessing stage, 20 bands affected by noise and water absorption were removed: 108–112, 154–167, and 224. The ground truth data for the Salinas dataset contains 16 classes, with a total of 54,129 pixels representing different land cover categories. For machine learning and image classification evaluation, 1% of the labeled samples were used for training, another 1% for validation, and the remainder for testing. The class names and the number of training and test samples are listed in Table 3.

Table 3. The class names and the training, validation and test sample numbers in SA.

Class	Land Cover Type	Train.	Val.	Test.
1	Brocoli-green-weeds-1	20	20	1969
2	Brocoli-green-weeds-2	37	37	3652
3	Fallow	20	20	1936
4	Fallow-rough-plow	14	14	1366
5	Fallow-smooth	27	27	2624
6	Stubble	39	39	3881
7	Celery	36	36	3507
8	Grapes-untrained	113	113	11,045
9	Soil-vinyard-develop	62	62	6079
10	Corn-senesced-green-weeds	33	33	3212
11	Lettuce-romaine-4wk	11	11	1046
12	Lettuce-romaine-5wk	19	19	1889
13	Lettuce-romaine-6wk	9	9	898
14	Lettuce-romaine-7wk	11	11	1048
15	Vinyard-untrained	72	72	7124
16	Vinyard-vertical-trellis	18	18	1771
	Total	541	541	53,047

5. Experiments

In this section, we describe the experimental setup, including comparative methods, evaluation metrics, and parameter configurations. Subsequently, we conducted quantitative experiments and ablation studies to assess the effectiveness of our proposed method.

5.1. Experimental Settings

Experimental Environment: The entire experimental process was conducted on a computer equipped with a GeForce RTX 3070Ti and an Intel i7-12700F 12-core processor, with 32 GB of memory, using Python 3.9 and PyTorch 2.0.1.

Data Acquisition: for the IP, PU, and SA datasets, training sets were randomly sampled at 10%, 5%, and 1%, respectively, using random seeds ranging from 1331 to 1340. The remaining data were used as the test sets.

Evaluation Metrics: three universal indicators were employed, i.e., overall accuracy (OA), average accuracy (AA), and the Kappa coefficient (Kappa).

Comparison Methods: to validate the effectiveness of our proposed method, we compared it with several other classification methods, including methods based on CNN and transformers. We evaluated each method with the most effective configuration. The comparison methods used here were as follows:

1. The 3D CNN (2016) [25] contains 3D convolution blocks and a softmax layer;
2. The SSRN (2017) [42], built upon 3D CNN by incorporating residual structures for shortcuts;
3. The HybridSN (2019) [23], which uses PCA for dimensionality reduction, and for which reduced data is fed into a 3DCNN, followed by a 2DCNN, so as to consider both spectral and spatial information simultaneously;

4. The SSFTT (2022) [28], which first learns features through 3DCNN and 2DCNN, and then utilizes a transformer for feature-based classifications;
5. The GAHT (2022) [27], a network structure composed of three convolutional layers and the integration of transformers;
6. The DBDA (2020) [24], a hyperspectral image classification approach that uses a two-branch CNN architecture with a dual-attention mechanism to effectively capture spectral and spatial features;
7. The FDSSC (2018) [22], a fast and efficient hyperspectral image classification framework that utilizes dense spectral–spatial convolutions for accurate classification.

Implementation Details: our LVGG model was implemented using the PyTorch framework. First, we reduced the number of bands to 35 through band selection. The PCA dimension and patch size were set to 7 and 41 × 41, respectively. We then adopted the Adam optimizer, with a batch size of 256 and a learning rate of 1×10^{-4}. Then, we used cross-entropy loss in lightweight classifiers. The training schedule involved the use of the Adam optimizer and Cosine Annealing. The original learning rate and minimum learning rate were set to 1×10^{-4} and 5×10^{-6}, respectively, and the number of epochs was set to 100 for all datasets. Finally, by repeating this experiment ten times with different training sample selections, we obtained the average values as the final results.

5.2. Classification Results

Compared to other methods, the proposed LVGG model achieved the highest OA and Kappa on three benchmark datasets. The classification plots are shown in Figures 2–4, respectively, while Tables 4–6 display the algorithms' classification accuracy regarding the three datasets in detail.

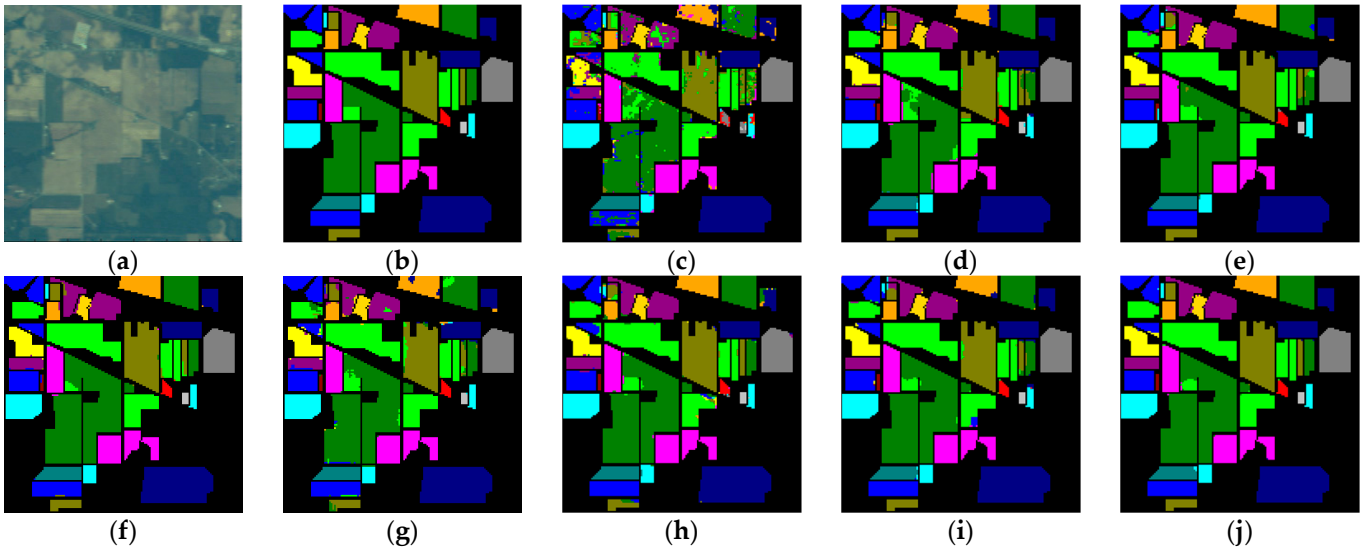

Figure 2. Classification maps of different methods for the IP dataset. (**a**) False-color image. (**b**) Ground truth image. (**c**) 3D CNN. (**d**) SSRN. (**e**) HybridSN. (**f**) SSFTT. (**g**) GAHT. (**h**) DBDA. (**i**) FDSSC. (**j**) LVGG.

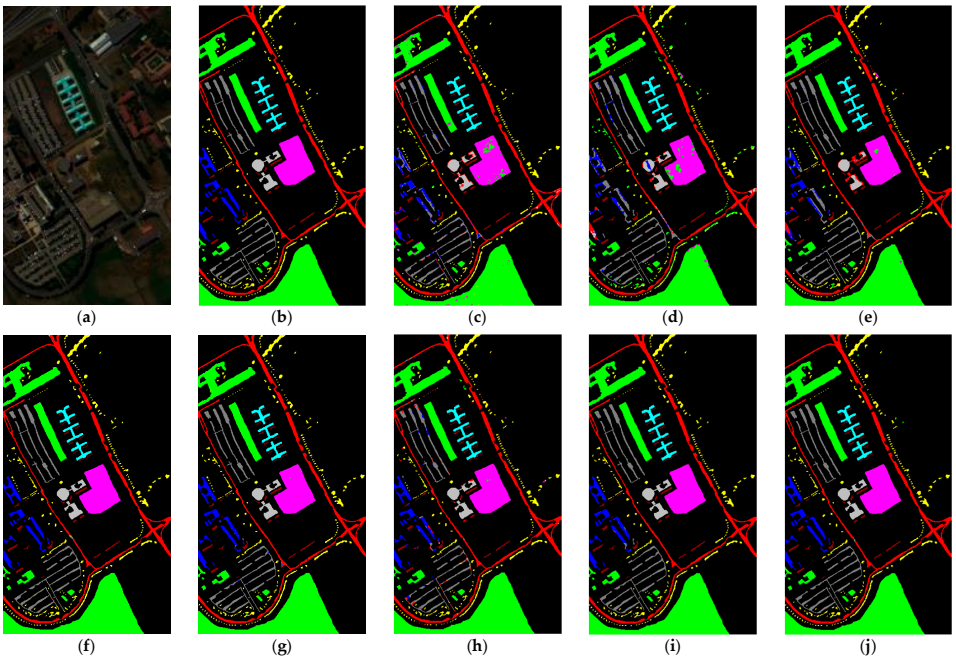

Figure 3. Classification maps of different methods for the PU dataset. (**a**) False-color image. (**b**) Ground truth image. (**c**) 3D CNN. (**d**) SSRN. (**e**) HybridSN. (**f**) SSFTT. (**g**) GAHT. (**h**) DBDA. (**i**) FDSSC. (**j**) LVGG.

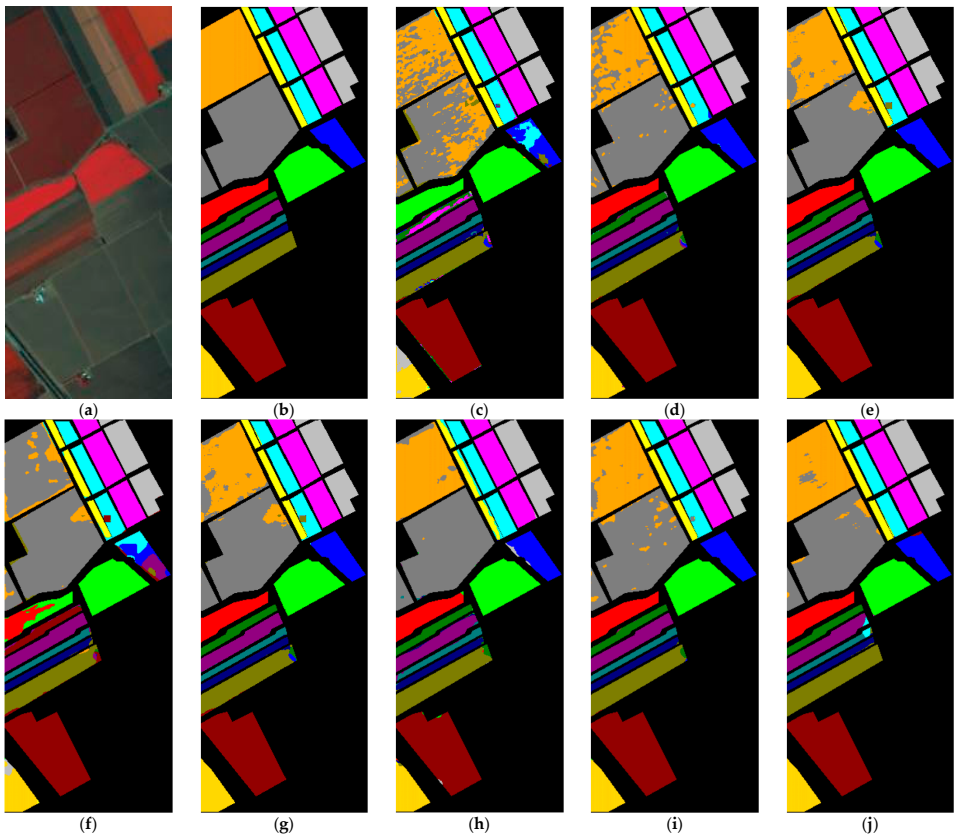

Figure 4. Classification maps of different methods for the SA dataset. (**a**) False-color image. (**b**) Ground truth image. (**c**) 3D CNN. (**d**) SSRN. (**e**) HybridSN. (**f**) SSFTT. (**g**) GAHT. (**h**) DBDA. (**i**) FDSSC. (**j**) LVGG.

Table 4. The classification accuracy of the IP dataset.

Class	3D CNN	SSRN	HybridSN	SSFTT	GAHT	DBDA	FDSSC	LVGG
1	58.54 ± 3.15	94.14 ± 2.16	92.68 ± 1.73	95.12 ± 3.51	97.56 ± 1.23	98.30 ± 1.66	**100± 0**	97.62 ± 1.71
2	76.19 ± 1.69	97.84 ± 0.86	96.62 ± 0.66	97.67 ± 1.13	98.05 ± 1.25	97.95 ± 1.29	**99.73 ± 0.26**	97.74 ± 0.79
3	77.64 ± 2.30	97.54 ± 1.79	98.41 ± 0.54	98.87 ± 0.35	98.66 ± 0.88	**99.40 ± 0.41**	98.89 ± 0.80	98.28 ± 0.73
4	52.11 ± 4.48	90.70 ± 3.10	98.97 ± 1.17	91.55 ± 4.73	95.31 ± 3.26	**100 ± 0**	**100 ± 0**	99.52 ± 0.73
5	93.56 ± 0.88	97.75 ± 1.32	97.70 ± 1.26	96.32 ± 1.52	95.17 ± 2.69	97.84 ± 1.57	98.83 ± 0.67	**99.10 ± 1.01**
6	98.17 ± 0.72	99.24 ± 0.55	99.69 ± 0.11	99.54 ± 0.45	**99.85 ± 0.14**	98.82 ± 1.52	99.65 ± 0.33	98.70 ± 0.20
7	36.00 ± 5.12	81.60 ± 4.01	98.85 ± 0.38	**100 ± 0**	**100 ± 0**	92.91 ± 9.86	95.23 ± 3.62	99.16 ± 1.67
8	98.60 ± 1.19	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**	99.67 ± 0.31	98.97 ± 0.40	**100 ± 0**
9	55.56 ± 2.45	74.44 ± 1.03	98.38 ± 3.26	88.89 ± 3.48	**100 ± 0**	93.75 ± 3.16	**100 ± 0**	94.74 ± 0.96
10	82.86 ± 3.02	94.77 ± 3.06	98.08 ± 0.81	97.71 ± 1.42	94.29 ± 2.30	98.47 ± 1.77	95.3 ± 2.21	**99.66** ± 0.40
11	90.45 ± 1.33	98.87 ± 0.90	97.47 ± 1.33	98.69 ± 1.28	99.37 ± 0.48	98.39 ± 0.91	**99.53 ± 0.44**	96.97 ± 0.64
12	62.55 ± 1.17	97.83 ± 2.13	98.85 ± 1.02	98.13 ± 1.77	96.63 ± 0.97	99.36 ± 0.26	98.73 ± 0.72	**99.40** ± 0.51
13	88.65 ± 3.58	99.24 ± 0.32	99.82 ± 0.24	97.28 ± 2.64	99.68 ± 0.40	96.36 ± 1.46	**100 ± 0**	**100 ± 0**
14	99.39 ± 0.60	99.18 ± 0.77	99.13 ± 0.92	**99.91 ± 0.07**	97.89 ± 2.16	99.40 ± 0.44	99.02 ± 0.49	**99.91 ± 0.09**
15	86.17 ± 2.85	93.95 ± 2.38	**100 ± 0**	98.84 ± 2.49	97.12 ± 1.58	95.59 ± 2.59	96.15 ± 1.31	99.14 ± 0.95
16	45.24 ± 5.36	**98.33 ± 1.55**	91.57 ± 4.07	95.54 ± 3.07	94.05 ± 3.13	93.33 ± 4.10	96.05 ± 2.45	94.54 ± 5.62
OA (%)	85.42 ± 3.75	96.88 ± 0.41	97.59 ± 0.56	97.47 ± 0.33	97.95 ± 0.28	98.10 ± 0.21	98.31 ± 0.17	**98.53 ± 0.38**
AA (%)	85.10 ± 2.98	97.51 ± 0.53	97.27 ± 0.93	96.57 ± 0.48	97.75 ± 0.55	98.35 ± 0.72	**98.63 ± 0.72**	97.44 ± 1.03
Kappa × 100	83.24 ± 2.31	97.23 ± 0.52	96.90 ± 0.37	97.11 ± 0.44	97.66 ± 0.38	98.01 ± 0.41	98.22 ± 0.65	**98.26 ± 0.44**

The highest values in each row are marked in bold to highlight the performance.

Table 5. The classification accuracy of the PU dataset.

Class	3D CNN	SSRN	HybridSN	SSFTT	GAHT	DBDA	FDSSC	LVGG
1	97.02 ± 1.70	99.29 ± 0.24	98.81 ± 0.76	99.33 ± 0.15	99.38 ± 0.15	99.67 ± 0.24	99.76 ± 0.10	**99.83 ± 0.06**
2	98.82 ± 1.17	99.71 ± 0.19	99.83 ± 0.17	99.92 ± 0.05	99.80 ± 0.10	99.85 ± 0.14	99.80 ± 0.04	**99.94 ± 0.03**
3	92.98 ± 1.31	96.92 ± 3.15	92.45 ± 4.90	98.29 ± 0.95	98.35 ± 0.85	99.41 ± 0.72	98.95 ± 1.14	**99.72 ± 0.26**
4	97.53 ± 1.58	99.89 ± 0.11	98.32 ± 0.88	98.49 ± 1.40	99.52 ± 0.40	98.96 ± 0.69	**99.92 ± 0.08**	99.30 ± 0.33
5	99.06 ± 1.20	99.52 ± 0.49	99.65 ± 0.29	99.53 ± 0.50	99.88 ± 0.12	99.82 ± 0.17	**99.91 ± 0.07**	99.65 ± 0.39
6	99.10 ± 0.85	99.87 ± 0.16	99.43 ± 0.56	99.21 ± 0.48	99.75 ± 0.05	99.91 ± 0.09	99.90 ± 0.03	**99.96** ± 0.04
7	79.10 ± 8.97	**99.76 ± 0.26**	**99.76 ± 0.19**	99.13 ± 1.01	99.60 ± 0.32	99.44 ± 0.11	99.68 ± 0.22	99.06 ± 1.27
8	97.34 ± 1.02	97.36 ± 1.68	**99.52 ± 0.50**	98.05 ± 0.57	98.63 ± 0.27	96.13 ± 3.10	99.45 ± 0.16	99.39 ± 0.59
9	95.22 ± 2.55	99.66 ± 0.23	99.82 ± 0.23	95.44 ± 2.17	99.53 ± 0.16	98.89 ± 1.15	**99.83 ± 0.05**	99.10 ± 0.22
OA (%)	97.88 ± 0.35	99.33 ± 0.29	99.10 ± 0.44	99.21 ± 0.36	99.43 ± 0.15	99.37 ± 0.40	99.55 ± 0.17	**99.76 ± 0.08**
AA (%)	95.26 ± 0.29	99.16 ± 0.18	98.60 ± 0.32	98.69 ± 0.44	99.37 ± 0.21	99.12 ± 0.49	99.22 ± 0.23	**99.56 ± 0.21**
Kappa × 100	97.19 ± 0.39	99.14 ± 0.22	98.81 ± 0.71	99.15 ± 0.39	99.39 ± 0.08	99.09 ± 0.53	99.18 ± 0.24	**99.69 ± 0.06**

The highest values in each row are marked in bold to highlight the performance.

Table 6. The classification accuracy of the SA dataset.

Class	3D CNN	SSRN	HybridSN	SSFTT	GAHT	DBDA	FDSSC	LVGG
1	97.07 ± 1.19	**100 ± 0**	99.76 ± 0.24	**100 ± 0**	99.45 ± 1.07	**100 ± 0**	**100 ± 0**	**100 ± 0**
2	99.86 ± 0.14	**100 ± 0**	99.97 ± 0.02	99.66 ± 0.52	**100 ± 0**	99.31 ± 0.11	99.93 ± 0.05	**100 ± 0**
3	92.14 ± 2.20	98.60 ± 1.07	95.81 ± 3.39	96.73 ± 1.71	95.76 ± 3.83	99.67 ± 0.32	**99.89 ± 0.11**	99.78 ± 0.19
4	98.33 ± 0.62	98.39 ± 2.24	96.05 ± 2.32	**99.22 ± 0.61**	98.90 ± 0.51	98.03 ± 1.01	98.36 ± 0.92	95.90 ± 4.13
5	96.23 ± 0.97	**100 ± 0**	93.62 ± 3.07	96.59 ± 2.68	98.88 ± 0.61	99.73 ± 0.26	99.83 ± 0.07	95.96 ± 2.41
6	99.58 ± 0.40	99.98 ± 0.01	99.46 ± 0.53	99.69 ± 0.37	**100 ± 0**	**100 ± 0**	**100 ± 0**	98.87 ± 1.33
7	99.38 ± 0.31	99.98 ± 0.02	94.59 ± 1.31	99.70 ± 0.21	99.91 ± 0.09	99.96 ± 0.06	99.90 ± 0.09	**100 ± 0**
8	83.96 ± 1.84	91.42 ± 5.45	96.78 ± 1.75	87.59 ± 1.46	93.18 ± 2.04	89.86 ± 6.44	94.53 ± 1.46	**97.68 ± 3.32**
9	97.89 ± 1.15	99.83 ± 0.16	98.85 ± 1.20	98.91 ± 1.22	**99.87 ± 0.10**	99.37 ± 0.66	99.86 ± 0.11	99.55 ± 0.65
10	89.27 ± 1.71	98.86 ± 0.66	98.60 ± 1.69	94.51 ± 1.71	96.88 ± 1.19	99.30 ± 0.33	**99.33 ± 0.09**	98.97 ± 1.19
11	83.08 ± 9.27	98.39 ± 0.86	93.23 ± 1.31	96.70 ± 2.25	99.06 ± 0.81	98.15 ± 1.43	98.13 ± 0.98	**99.70 ± 0.33**
12	99.06 ± 0.95	98.71 ± 1.55	97.62 ± 1.99	**99.88 ± 0.07**	99.84 ± 0.23	99.82 ± 0.15	99.75 ± 0.25	97.97 ± 3.48
13	97.62 ± 2.49	99.51 ± 0.53	90.57 ± 5.42	97.84 ± 2.66	99.60 ± 0.38	99.86 ± 0.16	**100 ± 0**	**100 ± 0**
14	96.36 ± 2.67	98.23 ± 0.47	94.63 ± 2.00	91.80 ± 4.85	98.07 ± 2.75	98.19 ± 1.59	98.48 ± 1.11	**99.77 ± 0.21**
15	74.55 ± 6.08	93.65 ± 5.82	98.50 ± 2.29	81.42 ± 3.77	90.82 ± 3.47	93.17 ± 6.17	97.39 ± 1.27	**99.29 ± 0.62**
16	91.76 ± 4.12	99.82 ± 0.13	99.27 ± 0.72	97.09 ± 1.58	96.87 ± 0.72	**100 ± 0**	99.26 ± 0.54	**100 ± 0**
OA (%)	90.89 ± 0.61	96.90 ± 0.69	97.16 ± 0.57	93.73 ± 0.39	96.80 ± 0.37	96.44 ± 1.51	98.01 ± 0.24	**98.69 ± 0.70**
AA (%)	93.51 ± 0.62	98.38 ± 0.30	96.54 ± 0.68	96.08 ± 0.47	98.00 ± 0.24	**98.95 ± 0.29**	98.54 ± 0.17	98.80 ± 0.81
Kappa × 100	89.86 ± 0.57	96.50 ± 0.77	96.84 ± 0.44	93.01 ± 0.44	96.44 ± 0.33	97.79 ± 0.73	97.94 ± 0.27	**98.54 ± 0.65**

The highest values in each row are marked in bold to highlight the performance.

5.2.1. IP

Table 4 shows the classification performance of different models on the IP dataset. Among the compared methods, SSRN, HybridSN, DBDA, and FDSSC employ relatively complex CNN-based frameworks, incorporating elements such as 3D CNN, residual connections, and MLP to enhance classification performance. However, these approaches come at the cost of consumed computational time. In contrast, SSFTT, based on a lightweight transformer architecture, offers a shorter training time but exhibits a slight decrease in accuracy compared to other methods.

GAHT adopts a hybrid approach, combining elements of both transformers and CNNs. This hybrid strategy results in shorter training times while improving accuracy. In comparison to other methods, compared to FDSSC, SSFTT, and GAHT, our LVGG improved 0.22%, 1.06%, and 0.58% in OA and 0.04%, 1.15%, and 0.60% in Kappa, respectively. However, it did show a slightly increased overall training time compared to SSFTT.

5.2.2. PU

For the PU dataset, because we chose a training set percentage of 5%, the training data volume was relatively large. As a result, all methods yielded good results. Among these methods, GAHT and FDSSC were two models that performed well, in addition to our method. The accuracy of Class 3, Class 8, and Class 9 significantly impacted the overall OA. While our method did not perform well in classifying Class 7, it did achieve the highest accuracy in most other classes. Furthermore, the transformer-based SSFTT did not perform as well as the hybrid model GAHT, which combines transformers and CNN. This suggests that convolutional methods still performed effectively on the PU dataset. However, our method still achieved the best OA, AA, and Kappa results while also being the fastest in terms of execution time.

5.2.3. SA

For the SA dataset, we selected only 1% of the data as the training set, which led to lower classification accuracy for the SSFTT method. The low accuracy of SSFTT in classifying Class 8 and Class 15 may be due to the limited number of samples that the transformer learned, coupled with the similarity of features. When dealing with a limited number of HSI samples, there are limitations associated with transformer-based methods; possibly, these methods require a larger amount of data to learn effectively due to weaker inductive biases. In contrast, CNN-based methods tend to perform better under these conditions, and our approach outperformed all others. This could be attributed to the use of a larger patch size, allowing for a broader learning scope that works well with small datasets.

5.3. Ablation Studies

We analyzed the proposed LVGG framework by transitioning from the VGG block to the lightweight-VGG block, and conducted experiments on the IP dataset with the corresponding results summarized in Table 7. First, the activation function of the FC layers was modified from ReLU to SELU, as SELU has self-normalizing properties and performs well in FC layers. Using SELU as the activation function for FC layers improved performance compared to the basic VGG structure.

We also improved the activation function for regular convolutions. While VGG uses ReLU as the activation function, we introduced DyReLU as an alternative to enhance the non-linearity of shallow networks. Using DyReLU as the activation function also resulted in improved accuracy.

Table 7. Ablation analysis of the proposed LVGG on the IP dataset.

Changes	OA (%)
(a) Baseline	97.13
(b) Redundant Data Removal	97.75
(c) Expand Convolution	97.70
(d) DyVGG Block	98.03
(e) Multi-Layer Perceptron	97.55
(b) + (c) + (d)	98.22
(b) + (c) + (e)	97.86
(b) + (c) + (d) + (e)	98.36

Regarding normalization, for the IP dataset, using a large batch size with batch normalization decreased the classification accuracy, while using layer normalization improved the accuracy. Therefore, layer normalization was adopted for normalization in this proposed network.

Activation functions and normalization have an impact on network classification accuracy. Additionally, the expand convolution also contributes to improved accuracy. As we had seven channels after the whitened PCA, using only seven channels for the classification limited the overall convolutional features. Therefore, expanding the convolution before regular the convolution enhances the network's learning ability and achieves better classification accuracy.

5.4. Parameter Analysis

We analyzed the effect of various parameters on the classification performance of our proposed lightweight-VGG method by conducting experiments using different parameter settings in the same experimental setting as described in Section 5.1.

5.4.1. The Effectiveness of Band Selection

The criteria for determining the number of bands is to minimize it as much as possible while ensuring the accuracy of SVM. As shown in Table 8, the highest accuracy appears when the number of selected bands is 35 on the IP dataset. In addition, when taken as 35, the number of bands and accuracy achieve a certain degree of balance on the other two HSI datasets. Therefore, 35 bands were chosen in this study.

Table 8. Impact of band selection on OA (%) for three datasets.

	15	20	25	30	35	40	45	50
IP	78.66	80.2	80.81	81.26	82.89	82.7	82.87	82.46
PU	89.52	90.09	93.02	93.17	93.26	93.29	93.41	93.48
SA	91.71	91.85	92.07	92.33	92.45	92.41	92.50	92.55

5.4.2. The Effectiveness of PCA Size Selection

We investigated the impact of different PCA dimensions on classification performance, evaluated using overall accuracy. The purpose of PCA dimensionality reduction is to reduce the data's dimensions, lowering computational cost and noise. By selecting an appropriate PCA dimension, noise or less significant variations can be filtered out. Lower PCA dimensions may filter out important information, while higher PCA dimensions may retain noise. Lower PCA dimensions result in higher data compression but may lead to information loss. Higher PCA dimensions retain more information but increase data dimensionality. Lower PCA dimensions typically enhance computational efficiency, as processing low-dimensional data is faster. However, in our case, we performed PCA after band selection, so setting the PCA dimensions to seven achieved outstanding performances.

As shown in Figure 5, the results with PCA dimensions set to seven were comparable to those retaining more dimensions, indicating that band selection filtered out most of the noise, and that PCA dimensions of seven were sufficient to achieve good performance.

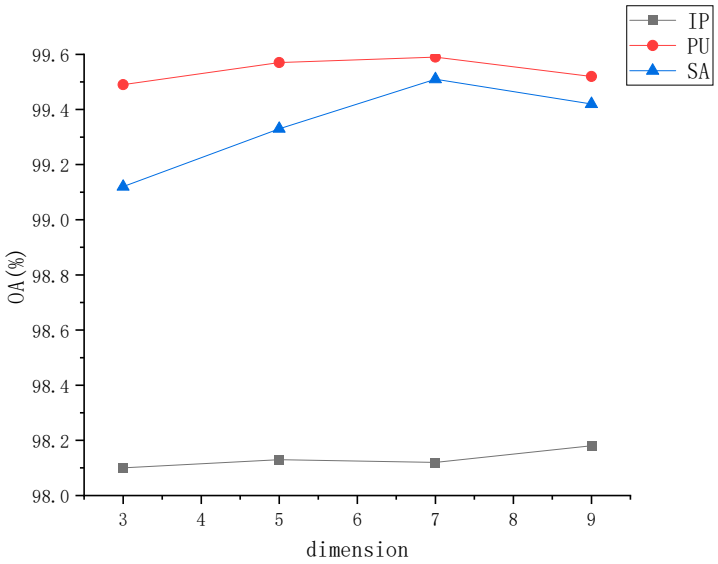

Figure 5. The three HSI datasets with distinct PCA dimensions for OA.

5.4.3. The Effectiveness of Patch Size Selection

We investigated the impact of different patch sizes on classification performance, which we evaluated based on overall accuracy. The patch sizes varied from 21 × 21 to 49 × 49, as shown in Figure 6. As the patch size increased, performance initially improved and then started to decline. When the patch size was set to 41 × 41, we obtained the best performance for the IP, PU, and SA datasets, with OAs of 98.10%, 99.72%, and 99.02%, respectively. The performance of the PU dataset remained relatively stable from 21 × 21 to 49 × 49, possibly due to the high training rate of the PU dataset. However, the IP and SA datasets showed variations in performance. This may be attributed to the use of max-pooling in our method, where smaller image patches did not have sufficient representation, and overly large patches could make the overall structure too complex. Therefore, we chose a patch size of 41 × 41.

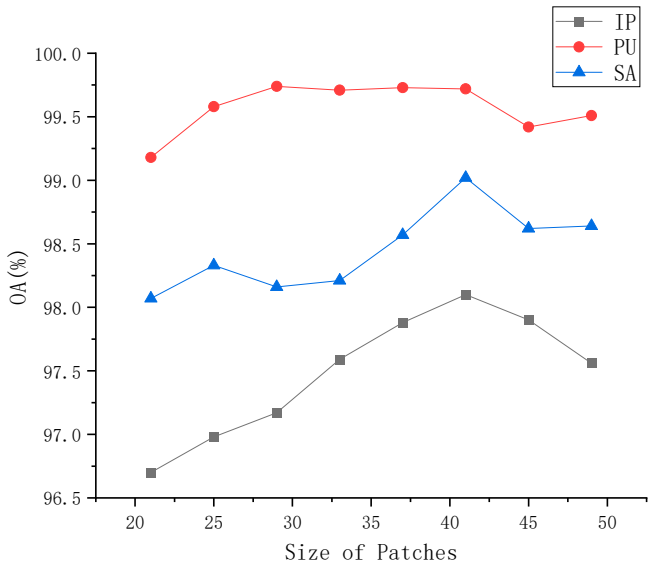

Figure 6. The three HSI datasets with distinct patch sizes for OA.

5.4.4. The Effectiveness of Neuron Number in the FC Layer

We investigated the impact of different neuron numbers in the FC layer on classification performance, evaluating based on overall accuracy. Neuron numbers ranged from 32 to 512, as shown in Figure 7. Performance improved with an increase in neurons but began to decline thereafter. On all three datasets, the best performance was generally achieved with 128 neurons, although 64 neurons also showed good results in the SA and PU datasets.

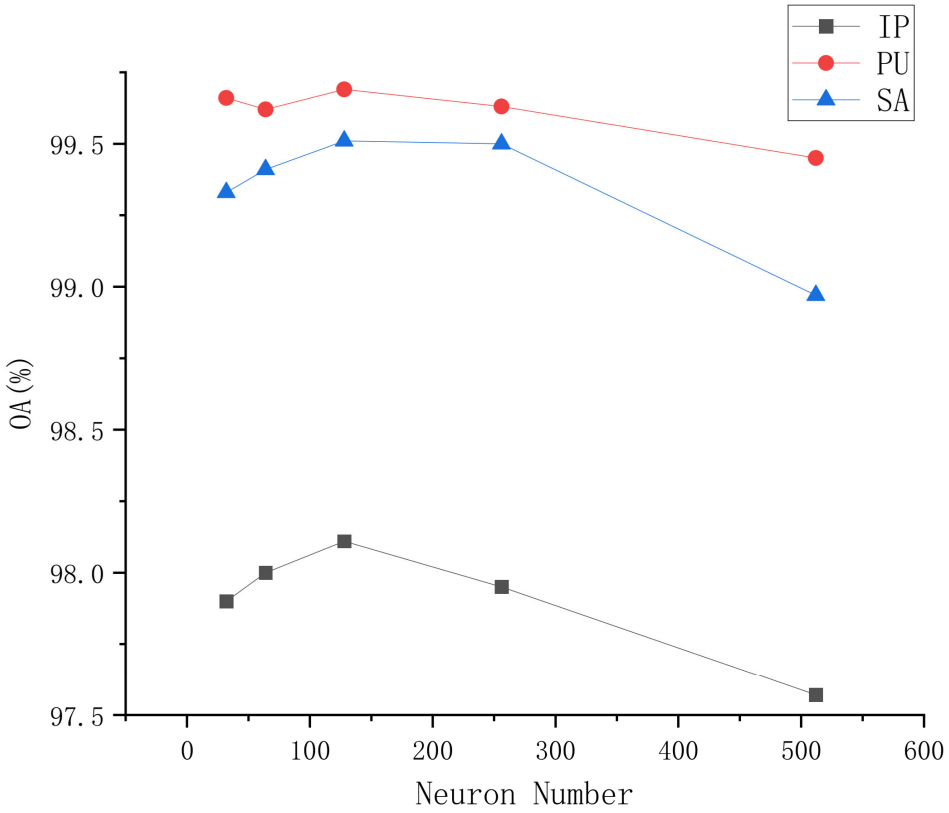

Figure 7. The three HSI datasets with distinct neuron numbers for OA.

5.5. Computational Efficiency

We also verified computational efficiency by observing the performance of these methods in terms of execution time, number of parameters, and the number of floating-point operations (FLOPs) on three HSI datasets. The statistical results are shown in Table 9.

SSRN is a combination of 3D CNN with a ResNet structure short-circuit.

HybridSN is a network that combines 3D CNN and 2D CNN, resulting in shorter execution times. Because it uses PCA for dimensionality reduction, followed by two rounds of 3D convolution for feature extraction, and because it uses a 2D convolutional network, it results in shorter execution times compared to SSRN.

Table 9. Analysis of computing efficiency among different methods on the IP dataset.

Model	3D CNN	SSRN	HybridSN	SSFTT	GAHT	DBDA	FDSSC	LVGG
Train Time (s)	161.04	80.96	67.43	10.70	53.36	116.01	798.77	19.95
Test Time (s)	6.33	3.82	2.85	0.29	2.41	6.16	6.25	2.62
Number of Parameters	623,281	364,168	5,122,176	148,488	972,624	382,326	1,227,490	758,672
GFLOPs	1.859	1.539	1.496	0.750	0.153	1.765	1.722	0.585

SSFTT is the fastest model in terms of execution time because it consists of only one layer of 3D CNN and one layer of 2D CNN, followed by the tokenization of features through the transformer. Due to the limited data, the transformer learns quickly. The model has the shortest training time and the least number of parameters. However, the overall accuracy is moderate.

GAHT is a method that combines transformers with CNN, so the overall execution time is similar as well. Although it has the lowest FLOPs, its execution time is not very short due to the involvement of group convolutions.

FDSSC is the most time-consuming CNN network because it involves reconstruction and feature extraction using complex 3D CNN. The high time complexity of 3D CNN results in the longest overall execution time.

DBDA employs dense 3D CNN for learning, much like FDSSC. However, it does not have the reconstruction step that FDSSC has, which significantly reduces its execution time.

In comparison to these methods, our proposed LVGG method has a short execution time because it only utilizes 1×1 and 3×3 2D convolutions. Its FLOPs are relatively low, but due to a relatively larger number of parameters compared to transformer methods, its runtime is somewhat longer than those of transformer-based approaches.

6. Conclusions

This study proposes a lightweight-VGG approach to improve HSI classification speed and accuracy. In this context, we introduce a method that performs PCA after selecting bands for classification. Most existing works either directly use raw images or apply PCA with a large number of retained dimensions, which can introduce noise and be sensitive to random seeds. Additionally, many transformer-based classification methods involve complex and computationally expensive operations, yet they do not achieve high accuracy in HSI classification. In contrast, our proposed LVGG approach reduces the number of parameters in the fully connected layers, resulting in a faster processing speed while maintaining high classification accuracy.

By performing PCA on the data after selecting the bands, and then conducting network classification, our proposed method provides improved stability and speed. Moreover, its smaller parameter size allows for faster processing, making it suitable for deployment on lightweight devices. Quantitative experiments using three HSI datasets showed that our LVGG approach outperformed existing methods in both performance and speed.

Author Contributions: Methodology, software and writing—original draft preparation, X.F. and S.W.; investigation and data curation, J.M. and G.W.; modification and editing, L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62006072, 62106067), the Key Technologies Research and Development Program of Henan Province (222102210108), the Ministry of Education Key Laboratory Open Funded Project for Grain Information Processing and Control (KFJJ2022013), the Natural Science Project of Zhengzhou Science and Technology Bureau (21ZZXTCX21), the Innovative Funds Plan of Henan University of Technology (2022ZKCJ11), and the Cultivation Programme for Young Backbone Teachers in Henan University of Technology.

Data Availability Statement: The data presented in this study are available in the article.

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

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

Abbreviations

The abbreviations for all key terms in this paper are explained below:

HSI	Hyperspectral Image
SVM	Support Vector Machine
CNNs	Convolutional Neural Networks
MLP	Multi-Layer Perceptron
FC	Fully Connected
LN	Layer Normalization
BN	Batch Normalization
VGG	Visual Geometry Group
SELU	Scaled Exponential Linear Unit
ReLU	Rectified Linear Unit
SSFTT	Spectral–Spatial Feature Tokenization Transformer
SSRN	Spectral–spatial Residual Network
HybridSN	Hybrid Spectral–spatial Network
GAHT	Group-Aware Transformer
DBDA	Double-Branch-Dual-Attention-Mechanism-Network
FDSSC	A Fast Dense Spectral–spatial Convolution Network
IP	Indian Pines
PU	Pavia University
SA	Salinas
AA	Average Accuracy
OA	Overall Accuracy
Kappa	Kappa coefficient
FLOPs	Floating-Point Operations

References

1. Fan, J.; Chen, T.; Lu, S. Superpixel guided deep-sparse-representation learning for hyperspectral image classification. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 3163–3173. [CrossRef]
2. Li, M.; Liu, Y.; Xue, G.; Huang, Y.; Yang, G. Exploring the relationship between center and neighborhoods: Central vector oriented self-similarity network for hyperspectral image classification. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 1979–1993. [CrossRef]
3. Yu, Y.; Fu, L.; Cheng, Y.; Ye, Q. Multi-view distance metric learning via independent and shared feature subspace with applications to face and forest fire recognition, and remote sensing classification. *Knowl. Based Syst.* **2022**, *243*, 108350. [CrossRef]
4. Mok, T.C.; Chung, A. Fast symmetric diffeomorphic image registration with convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4644–4653.
5. Stuart, M.B.; McGonigle, A.J.; Willmott, J.R. Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems. *Sensors* **2019**, *19*, 3071. [CrossRef] [PubMed]
6. Alamús, R.; Bará, S.; Corbera, J.; Escofet, J.; Palà, V.; Pipia, L.; Tardà, A. Ground-based hyperspectral analysis of the urban nightscape. *ISPRS J. Photogramm. Remote Sens.* **2017**, *124*, 16–26. [CrossRef]
7. Sun, Z.; Leng, X.; Lei, Y.; Xiong, B.; Ji, K.; Kuang, G. BiFA-YOLO: A novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images. *Remote Sens.* **2021**, *13*, 4209. [CrossRef]
8. Bandos, T.V.; 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. 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]
10. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814. [CrossRef]
11. Hong, D.; Wu, X.; Ghamisi, P.; Chanussot, J.; Yokoya, N.; Zhu, X.X. Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3791–3808. [CrossRef]
12. Sun, Z.; Dai, M.; Leng, X.; Lei, Y.; Xiong, B.; Ji, K.; Kuang, G. An anchor-free detection method for ship targets in high-resolution SAR images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 7799–7816. [CrossRef]
13. Hong, D.; Yokoya, N.; Chanussot, J.; Zhu, X.X. An augmented linear mixing model to address spectral variability for hyperspectral unmixing. *IEEE Trans. Image Process.* **2018**, *28*, 1923–1938. [CrossRef] [PubMed]
14. Shahshahani, B.M.; Landgrebe, D.A. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. Geosci. Remote Sens.* **1994**, *32*, 1087–1095. [CrossRef]
15. Dash, M.; Liu, H. Feature selection for classification. *Intell. Data Anal.* **1997**, *1*, 131–156. [CrossRef]

16. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
17. Martinez, A.M.; Kak, A.C. Pca versus lda. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 228–233. [CrossRef]
18. Draper, B.A.; Baek, K.; Bartlett, M.S.; Beveridge, J.R. Recognizing faces with PCA and ICA. *Comput. Vis. Image Underst.* **2003**, *91*, 115–137. [CrossRef]
19. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
20. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
21. Yang, X.; Ye, Y.; Li, X.; Lau, R.Y.; Zhang, X.; Huang, X. Hyperspectral image classification with deep learning models. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5408–5423. [CrossRef]
22. Wang, W.; Dou, S.; Jiang, Z.; Sun, L. A fast dense spectral–spatial convolution network framework for hyperspectral images classification. *Remote Sens.* **2018**, *10*, 1068. [CrossRef]
23. 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.* **2019**, *17*, 277–281. [CrossRef]
24. Li, R.; Zheng, S.; Duan, C.; Yang, Y.; Wang, X. Classification of hyperspectral image based on double-branch dual-attention mechanism network. *Remote Sens.* **2020**, *12*, 582. [CrossRef]
25. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]
26. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking hyperspectral image classification with transformers. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5518615. [CrossRef]
27. Mei, S.; Song, C.; Ma, M.; Xu, F. Hyperspectral image classification using group-aware hierarchical transformer. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5539014. [CrossRef]
28. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–spatial feature tokenization transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5522214. [CrossRef]
29. Zhao, G.; Ye, Q.; Sun, L.; Wu, Z.; Pan, C.; Jeon, B. Joint classification of hyperspectral and lidar data using a hierarchical cnn and transformer. *IEEE Trans. Geosci. Remote Sens.* **2022**, *61*, 5500716. [CrossRef]
30. Zhang, Z.; Ma, Q.; Zhou, H.; Gong, N. Nested Transformers for Hyperspectral Image Classification. *J. Sens.* **2022**, *2022*, 6785966. [CrossRef]
31. Zhang, X.; Su, Y.; Gao, L.; Bruzzone, L.; Gu, X.; Tian, Q. A Lightweight Transformer Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5517617. [CrossRef]
32. Ye, Q.; Huang, P.; Zhang, Z.; Zheng, Y.; Fu, L.; Yang, W. Multiview Learning With Robust Double-Sided Twin SVM. *IEEE Trans. Cybern.* **2022**, *52*, 12745–12758. [CrossRef]
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. Mei, S.; Ji, J.; Geng, Y.; Zhang, Z.; Li, X.; Du, Q. Unsupervised spatial–spectral feature learning by 3D convolutional autoencoder for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6808–6820. [CrossRef]
35. 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]
36. Paoletti, M.E.; Haut, J.M.; Plaza, J.; Plaza, A. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 120–147. [CrossRef]
37. Yu, S.; Jia, S.; Xu, C. Convolutional neural networks for hyperspectral image classification. *Neurocomputing* **2017**, *219*, 88–98. [CrossRef]
38. 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]
39. Li, X.; Ding, M.; Pižurica, A. Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2615–2629. [CrossRef]
40. Hao, S.; Wang, W.; Ye, Y.; Nie, T.; Bruzzone, L. Two-stream deep architecture for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 2349–2361. [CrossRef]
41. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
42. 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.* **2017**, *56*, 847–858. [CrossRef]
43. Fu, L.; Zhang, D.; Ye, Q. Recurrent thrifty attention network for remote sensing scene recognition. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 8257–8268. [CrossRef]
44. Sun, L.; Fang, Y.; Chen, Y.; Huang, W.; Wu, Z.; Jeon, B. Multi-structure KELM with attention fusion strategy for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5539217. [CrossRef]
45. 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.
46. 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.
47. 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]

48. He, X.; Chen, Y.; Lin, Z. Spatial-spectral transformer for hyperspectral image classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
49. Van Der Maaten, L.; Postma, E.O.; van den Herik, H.J. Dimensionality reduction: A comparative review. *J. Mach. Learn. Res.* **2009**, *10*, 13.
50. Fu, L.; Li, Z.; Ye, Q.; Yin, H.; Liu, Q.; Chen, X.; Fan, X.; Yang, W.; Yang, G. Learning Robust Discriminant Subspace Based on Joint $L_{2,p}$- and $L_{2,s}$-Norm Distance Metrics. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 130–144. [CrossRef] [PubMed]
51. Ye, Q.; Li, Z.; Fu, L.; Zhang, Z.; Yang, W.; Yang, G. Nonpeaked discriminant analysis for data representation. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3818–3832. [CrossRef] [PubMed]
52. Li, W.; Prasad, S.; Fowler, J.E.; Bruce, L.M. Locality-preserving dimensionality reduction and classification for hyperspectral image analysis. *IEEE Trans. Geosci. Remote Sens.* **2011**, *50*, 1185–1198. [CrossRef]
53. Wang, Q.; Li, Q.; Li, X. A fast neighborhood grouping method for hyperspectral band selection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5028–5039. [CrossRef]
54. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
55. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
56. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-normalizing neural networks. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 971–980.