remote sensing

MDPI

Article

Hyperspectral Images Weakly Supervised Classification with Noisy Labels

Chengyang Liu, Lin Zhao * and Haibin Wu

Heilongjiang Province Key Laboratory of Laser Spectroscopy Technology and Application, Harbin University of Science and Technology, Harbin 150080, China; 1920600028@stu.hrbust.edu.cn (C.L.); woo@hrbust.edu.cn (H.W.)
* Correspondence: zhaolin@hrbust.edu.cn

Abstract: The deep network model relies on sufficient training samples to achieve superior processing performance, which limits its application in hyperspectral image (HSI) classification. In order to perform HSI classification with noisy labels, a robust weakly supervised feature learning (WSFL) architecture combined with multi-model attention is proposed. Specifically, the input noisy labeled data are first subjected to multiple groups of residual spectral attention models and multi-granularity residual spatial attention models, enabling WSFL to refine and optimize the extracted spectral and spatial features, with a focus on extracting clean samples information and reducing the model's dependence on labels. Finally, the fused and optimized spectral-spatial features are mapped to the multilayer perceptron (MLP) classifier to increase the constraint of the model on the noisy samples. The experimental results on public datasets, including Pavia Center, WHU-Hi LongKou, and HangZhou, show that WSFL is better at classifying noise labels than excellent models such as spectral-spatial residual network (SSRN) and dual channel residual network (DCRN). On Hangzhou dataset, the classification accuracy of WSFL is superior to DCRN by 6.02% and SSRN by 7.85%, respectively.

Keywords: hyperspectral images; weakly supervised classification; noisy labels; multilayer perceptron

Citation: Liu, C.; Zhao, L.; Wu, H. Hyperspectral Images Weakly Supervised Classification with Noisy Labels. *Remote Sens.* **2023**, *15*, 4994. https://doi.org/10.3390/rs15204994

Academic Editor: Pedro Melo-Pinto

Received: 16 August 2023
Revised: 9 October 2023
Accepted: 14 October 2023
Published: 17 October 2023

1. Introduction

Hyperspectral remote sensing uses a large number of narrow electromagnetic wave channels to obtain spatial, radiation, and spectral triple information of ground objects, which obtains information about ground objects through the band range of visible light and infrared light in the electromagnetic spectrum [1,2]. Due to the characteristic of "combination of image and spectrum" in hyperspectral images, they contain a much higher degree of ground information. By fully utilizing this feature, accurate classification of ground objects can be achieved [3]. Therefore, hyperspectral remote sensing has been widely applied in urban planning [4], environmental monitoring [5], and precision agriculture [6,7].

Early hyperspectral image classification mostly used supervised classification methods, whose performance relied on high-quality labels [8]. Pal et al. mapped hyperspectral data to a high-dimensional space, and found an optimal segmentation hyperplane in the space to maximize the distance between different types and achieve the best classification effect [9]. Cui et al. considered the relationship between hyperspectral data classes, effectively combining a sparse representation classifier and K-nearest neighbor to increase classification accuracy [10].

Due to intra-class complexity and the scarcity of labeled samples, it is challenging to achieve high-precision ground object classification only through spectral features [11]. Gu et al. proposed a multi-kernel learning (MKL) architecture to learn spectral and spatial information, combined with a support vector machine (SVM) [12]. Liu et al. proposed a multi-morphic superpixel method to extract spectral and spatial features and complete the

classification task [13]. The extracted features of these methods require manual adjustment, which is a complex process.

Recently, owing to the powerful automatic feature extraction ability of deep learning, great progress has been made in supervised learning algorithms. With the local feature extraction capability of convolutional neural networks (CNN) [14,15], 2DCNN [16,17] and 3DCNN [18,19] have achieved impressive results in spectral, spatial, and spectral-spatial feature extraction. The use of supervised classification methods to process hyperspectral images requires manual annotation of data. In the process of obtaining labeled data, due to the high complexity of terrain, the lack of on-site surveys by legal persons, and the subjective influence of human factors on the quality of annotated data labels, the existence of noise in labels cannot be avoided; that is, the labels do not match their actual categories [20]. When the labels in the training data are incomplete, inaccurate, or only partially labeled, this is a weakly supervised learning method [21].

At present, there are roughly two methods for dealing with noise labeling problems, one of which is noise removal. For example, Kang et al. utilized domain transformation recursive filtering to enhance the distinguishability of spectral-spatial features and used a constrained energy minimization strategy to correct noisy samples [22]. Tu et al. proposed a spatial density peak clustering algorithm that utilizes a local density strategy to obtain inter-sample density values, where noisy samples are removed through decision functions [23,24]. Fang et al. used a confidence learning framework to accurately detect label errors [25].

The noise samples also have rich spatial-spectral information, which is also beneficial and can be effectively utilized. Another method is to design a more robust network model that directly utilizes a training set with incorrect labels [26,27]. Sukhbaatar et al. introduced a constrained linear "noise" layer on the softmax layer, so that the network output can simulate the noise label distribution and significantly improve the performance of the deep network [28]. Jiang et al. learned noisy labels through a small loss strategy, which indicates that deep neural networks will use small loss samples as "clean" samples and only use such samples for feedback propagation to update network parameters [29]. These methods have excellent noise processing capabilities, but their ability to handle complex noisy samples is limited, which limits the further improvement of classification accuracy. Xu et al. proposed a simple and efficient dual channel residual network (DCRN), which, respectively, extracted more refined features from the spectral and spatial dimensions of hyperspectral data to reduce the impact of noisy labels on the model to achieve an excellent training effect [30].

The presence of noisy samples brings many uncertainties to hyperspectral image classification, which limits the performance of many network models. However, due to the scarcity of hyperspectral image data, even noise samples cannot be easily discarded, otherwise a lot of information will be lost. In addition, deep learning models have the problem of easily memorizing clean samples in the early stages and gradually memorizing noisy samples in the later stages. As the number of iterations increases, the model gradually begins to fit noisy samples, which leads to the model continuously learning the features of noisy samples in later training and ignoring more important clean samples. Finally, there is a supervised learning model that is highly dependent on correctly labeled data, which will lead to the influence of noise samples being equal to clean samples. How to reduce the dependence of the model on noisy samples and further improve the performance of weakly supervised classification models has become a research hotspot in hyperspectral image classification.

Therefore, this paper designs a novel weakly supervised feature learning (WSFL) classification model for processing HSI with noisy samples. In order to better utilize noisy samples and preserve the diversity of features, this method adaptively learns features through multi-model attention feature learning. By comparing the similarity between samples, it obtains clean samples' features with higher information content, reducing the weight of noisy samples and the impact of noisy samples on the model. Secondly, in order to reduce the ability of the model to fit noisy samples in the later stage, multiple

sets of residual spectral attention models were designed in the spectral dimension. The spectral features in the later stage were differentiated in the space of multiple sets of spectral features to avoid excessive concentration of single layer spectral features and memory of noisy samples. In addition, in order to obtain more high-quality features and reduce the dependence of the model on samples, a multi-granularity residual spatial attention model was designed in the spatial dimension. In the multi-granularity space, the spatial features were further refined to obtain finer spatial features. Finally, in order to eliminate the adverse effects of local connectivity in the model and focus on the spatial structure information of more data, a MLP model was introduced, with a focus on learning spectral-spatial features to enhance the overall model's feature capture ability. The main contributions are summarized as follows:

1. This paper proposes a weakly supervised feature learning architecture combined with multi-model attention, which can build a more robust network that can classify noisy samples more stably and accurately;
2. In order to enhance the constraint of spectral dimension on noisy samples, multiple sets of residual spectral attention models were designed to enhance the ability to learn clean samples and weaken the model's fitting ability for noisy samples;
3. In order to improve the utilization of clean samples in weakly supervised models, a multi-granularity residual spatial attention model was designed to gradually extract clean sample information from spatial dimensions and obtain more significant features;
4. We introduced a MLP model to further extract spectral-spatial features, eliminate the adverse effects of local connectivity of the model, pay more attention to the spatial structure information of the data, and improve the overall model's anti-interference ability against noise.

2. Methodology

In this section, we will introduce the main architecture of WSFL in detail as shown in Figure 1, including multi-group residual spectral attention model (MGRSAM), multi-granularity residual spatial attention model (MRSAM), MLP model, noise loss function and Lion optimizer. In addition, samples labeled incorrectly are called noisy samples, and samples labeled correctly are called clean samples. First, the 3D data cube is input to MGRSAM and MRSAM to extract spectral and spatial features. In MGRSAM, the first two convolutional layers are used to perform coarse feature extraction on the spectral dimension. Subsequently, the extracted features are mapped to multiple sets of spectral feature spaces through the Group Convolution (GConv) layer to reduce the model's ability to fit noise samples. In addition, the features of the first layer are mapped into this space by means of skip connections, which solves the problem of gradient descent due to the increase in network depth. Secondly, the output features are mapped to the spectral feature attention space, focusing on extracting clean samples' features and suppressing the influence of noisy samples.

2.1. Spectral and Spatial Feature Extraction

In order to improve the robustness and generalization ability of image classification with noisy labels, this paper addresses two aspects separately. On the one hand, in response to the fact of neural networks easily remembering clean samples in the early stage and gradually remembering noisy samples in the later stage, this paper designs multiple sets of residual spectral attention models in the spectral dimension of hyperspectral data. In the early training of the spectral dimension, rough extraction of spectral dimension features is performed to obtain higher quality feature maps and enhance noise resistance. Secondly, in the later training, in order to avoid the model overfitting the features of noisy samples, the input features are mapped to multiple sets of spectral feature spaces, and the later spectral features are processed in a grouping manner. Each set of spectral features is finely extracted to avoid a single layer of spectral features being too concentrated, thus memorizing the

noisy samples. Secondly, while reducing the fitting of noise samples, the ability to fit clean samples is strengthened, and more clean spectral features are learned through the spectral attention model.

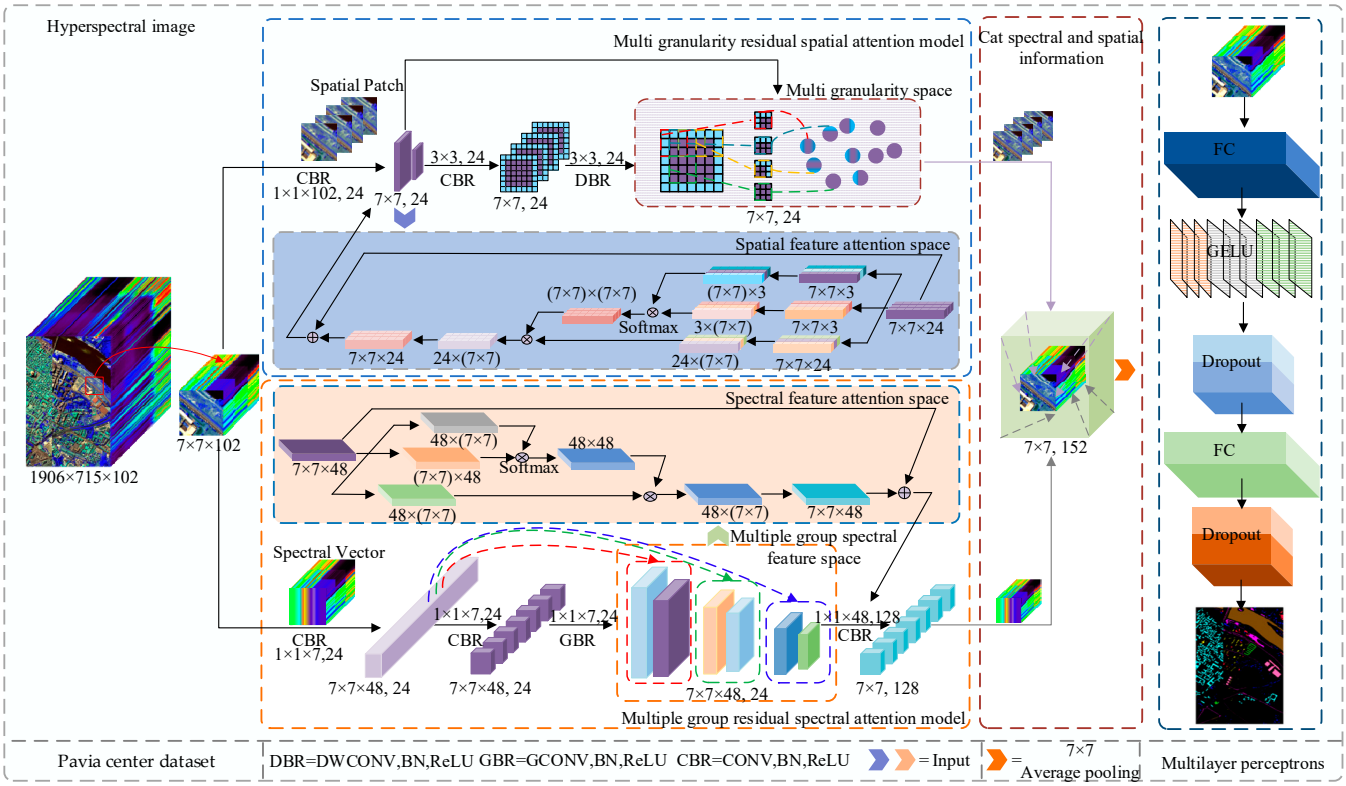


Figure 1. Framework of the proposed WSFL for HSI classification.

On the other hand, a multi-granularity residual spatial attention model is designed in the spatial dimension of HSI to solve the problem that supervised learning is too dependent on labeled samples and easy to overfit noisy samples. In early training of spatial dimensions, learning more discriminative spatial features through the spatial attention space weakens the spatial feature weights of noisy samples. Secondly, we map the input features to a multi-granularity space, extract important features in the spatial domain, obtain the similarity features of a large number of positive and negative sample pairs, mine the feature representation information of the dataset, obtain predictive tags, and reduce the dependence of supervised learning on labels. These two parts will be introduced in detail as follows.

2.1.1. Spectral Feature Extraction

This paper carefully designed the network architecture in the spectral dimension of hyperspectral data. In the early training of spectral dimension, the focus is on obtaining higher quality feature maps to improve the anti-noise ability of spectral dimension in the early stage. Secondly, in order to prevent the model from overfitting noisy samples in the later training, different channels are grouped to avoid excessive concentration of noise features in one layer, reducing the ability of spectral dimension to overfitting noisy samples in the later stage. However, this approach also reduces the ability to fit clean samples. To achieve this, spectral feature attention space is used to focus on more discriminative clean sample features among numerous features, while suppressing unnecessary noise information and enhancing the spectral dimension's ability to fit clean samples in the later stage. Multiple residual spectral attention models are composed of convolutional layers, spectral feature attention spaces, multiple spectral feature spaces, and residual blocks, as shown in Figure 2.

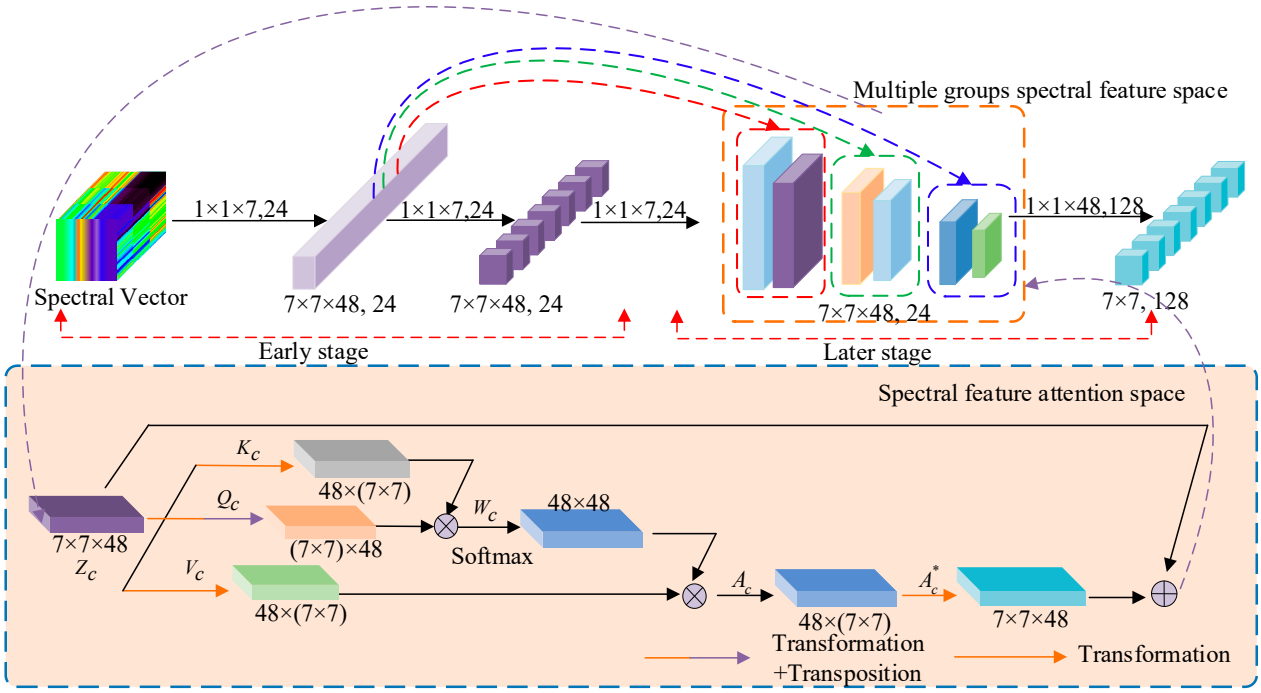


Figure 2. Multiple groups residual spectral attention model.

In the early training process of spectral dimension, we use a $1 \times 1 \times 7$ convolutional layer for coarse feature extraction. The step of the first convolutional kernel is (1, 1, 2) to remove information redundancy in adjacent bands, allowing the model to focus on more important spectral features and maintain the original spatial correlation, improving the early noise resistance of the spectral dimension model.

In the later stage of spectral dimension training, when the concatenated features of the spectral dimension are mapped to multiple sets of spectral feature spaces, different channels are grouped to prevent noise features from being too concentrated in one layer, and to avoid fitting noisy samples in that layer and affecting the overall training of the spectral dimension in the later stage. Multiple spectral feature spaces can reduce the ability of later models to fit noisy samples. In addition, feature extraction for each group of spectral features can also better explore spectral information and enhance the noise resistance of multiple spectral feature spaces.

As shown in Figure 3, there are multiple groups of spectral feature spaces, where Group = 3 and the size of each group of feature maps is $H \times W \times C_1/3$, which corresponds to the height, width, and number of channels. The size of each group of convolution kernels is $h_1 \times w_1 \times C_1/3$, which corresponds to the height, width, and number of channels of the convolution kernels. Convolution is performed in the corresponding group, and the output features are obtained by stacking the output features.

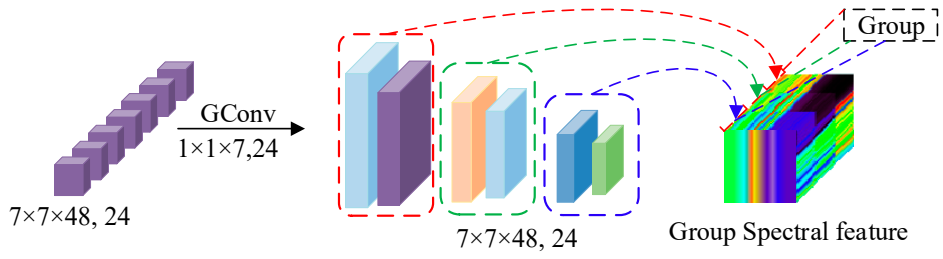


Figure 3. Multiple groups spectral feature space.

Although the fitting ability of noisy samples can be effectively reduced in multiple sets of feature spectral spaces, it also reduces the fitting ability of clean samples. In order to enhance the feature capture ability of clean samples in the later stage of spectral dimension, the features Z_c output from multiple sets of spectral feature spaces are mapped to the spectral attention space. In this space, different features have different weights, and the allocation of weights is based on similarity. Features with high similarity are considered clean sample features, while features with low similarity are considered noisy sample features.

Firstly, we transform Z_c from top to bottom to obtain $\boldsymbol{K_c}$, $\boldsymbol{Q_c}$, and $\boldsymbol{V_c}$ representing the key vector, query vector, and numerical vector, respectively. The subscript c represents the channel attention module, $P \times P$ is the spatial dimension with a channel count of 48. We calculate and compare the similarity between $\boldsymbol{K_c}$ and $\boldsymbol{Q_c}$, as shown in Formula (1).

$$f_c = \boldsymbol{K_c} \boldsymbol{Q_c^T} \tag{1}$$

Then, the Softmax classifier is used to obtain the pixel weight matrix $\boldsymbol{W_c}$, where $\boldsymbol{W_c}(i, j)$ represents the similarity of pixel i to pixel j.

$$W_c(i, j) = \frac{e^{f(i,j)}}{\sum_{j=1}^{p \times p} e^{f(i,j)}} \tag{2}$$

The spectral attention feature is obtained by multiplying $\boldsymbol{V_c}$ and $\boldsymbol{W_c}^T$, and the spectral attention feature obtained by weighting $\boldsymbol{V_c}$ with $\boldsymbol{W_c}^T$ is a discriminative feature.

$$A_c = \boldsymbol{W_c^T} \boldsymbol{V_c} \tag{3}$$

Finally, we transform the channel attention feature into A_c^*, so that its dimension is the same as the input feature, and add the spectral attention feature to the later training, as shown in Formula (4).

$$Z_c^* = A_c^* + Z_c \tag{4}$$

2.1.2. Spatial Feature Extraction

For spatial feature extraction, residual networks can effectively extract spatial features and prevent overfitting. However, the presence of noise samples can easily lead to a portion of the noise sample features being transmitted to the lower layer after each jump connection. In order to obtain more significant semantic features, noise information around the target pixel is suppressed at different spatial positions, highlighting clean sample features, improving the model's efficiency in feature utilization, and reducing the model's dependence on annotated data.

This paper designs a network architecture in the hyperspectral spatial dimension. In the early training of the spatial dimension, the attention space of spatial features is used to generate a weight value for each pixel in the input patch. This is done to suppress the negative impact of noisy samples on feature extraction and thereby strengthen spatial texture features. The size of the patch is 7×7. Compared to smaller domains, larger domains mean that the input contains more spatial information, which will also increase the number of noisy samples. Hence, relying solely on spatial feature attention space cannot completely reduce the interference of noise samples in spatial dimensions. Therefore, in the later training of the spatial dimension, each layer of feature maps is separated from one another through multi-granularity space, and each feature map is further subdivided into 3×3 regions, where multi-granularity refers to the processing and extraction of features in feature maps at different levels. In a multi-granularity space, emphasis is placed on the ground feature information within the granularity to obtain more discriminative spatial features and further enhance the feature capture ability for clean samples. The multi-granularity residual spatial attention model consists of convolutional layers, spatial

feature attention spaces, multi-granularity spaces, and residual blocks, with an architecture shown in Figure 4.

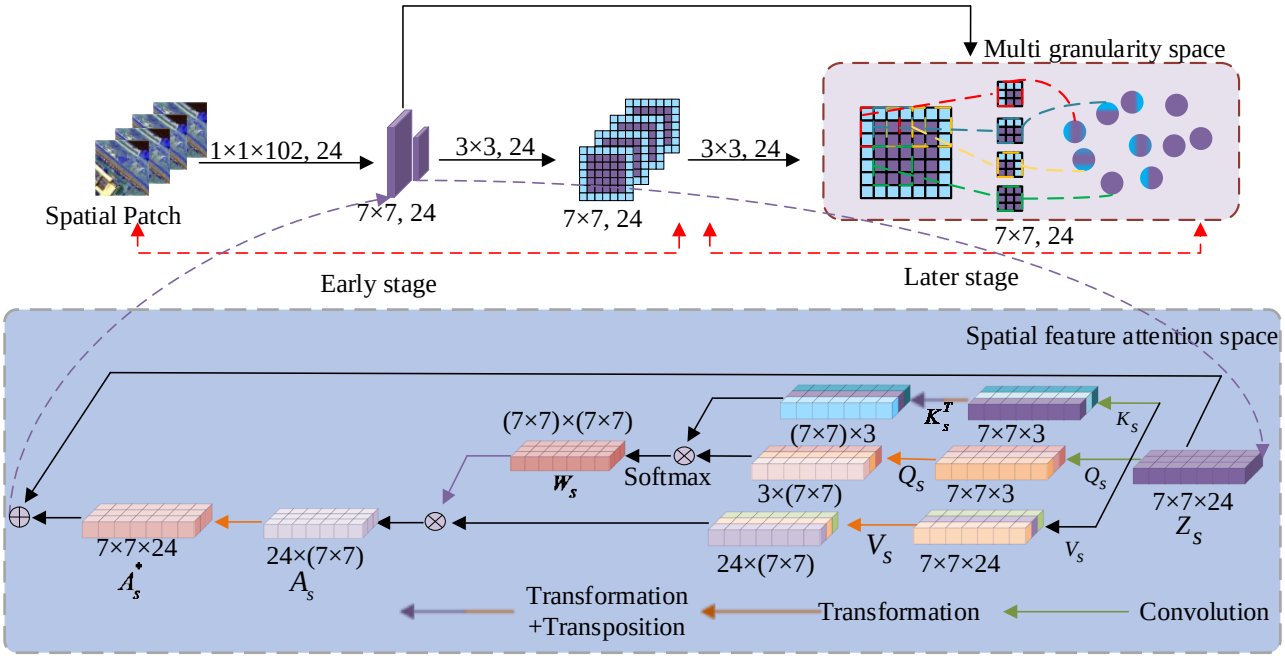


Figure 4. Multi-granularity residual spatial attention model.

In early training of the spatial dimensions, 3D convolution is first used to map hyperspectral data to a multi-granular residual spatial attention model with only one band, and the convolution kernel size is $1 \times 1 \times 102$. Then the coarse feature extraction is performed using the first layer of convolution, and the domain patch size of 7×7 is mapped to the attention space of spatial features. In order to better process spatial information, both the second and third convolutional layers use a 2D convolution size of 3×3 extracts of spatial features, and after each convolution, they are cascaded with a BatchNorm (BN) layer and a ReLU layer.

Firstly, a convolution operation is performed on input Z_s to obtain three feature maps, namely $\boldsymbol{K_s}$, $\boldsymbol{Q_s}$, and $\boldsymbol{V_s}$ from top to bottom, where the subscript s represents the weight of the convolutional layer for spatial attention modules W_K, W_Q, and W_V, respectively. B_K, B_Q, and B_V represent bias terms. The three feature maps are obtained as shown in Formula (5).

$$\begin{aligned} \boldsymbol{K_s} &= Conv(Z_s, W_K) + B_K \\ \boldsymbol{Q_s} &= Conv(Z_s, W_Q) + B_Q \\ \boldsymbol{V_s} &= Conv(Z_s, W_V) + B_V \end{aligned} \tag{5}$$

We transform three feature maps to obtain $\boldsymbol{K_s^T}$, $\boldsymbol{Q_s}$, and $\boldsymbol{V_s}$, and multiply $\boldsymbol{K_s^T}$ by $\boldsymbol{Q_s}$ to calculate the correlation of pixels in the spatial feature map, as shown in Formula (6).

$$f_s = \boldsymbol{K_s^T Q_s} \tag{6}$$

Then, Softmax is used to obtain the pixel weight matrix $\boldsymbol{W_s}$, where $W_s(i,j)$ represents the impact of pixel i on pixel j. Similarly, a larger weight value indicates a stronger correlation between spatial pixels, as shown in Equation (7).

$$W_S(i,j) = \frac{e^{f(i,j)}}{\sum_{j=1}^{p \times p} e^{f(i,j)}} \tag{7}$$

Subsequently, by multiplying $\boldsymbol{V_s}$ and $\boldsymbol{W_s}^T$ to obtain spatial attention features, the spatial features with significant weight are more helpful in improving classification results, as shown in Formula (8).

$$A_s = \boldsymbol{W_s^T V_s} \tag{8}$$

Finally, we transform the spatial attention feature into A_S^*, and add the spatial attention feature to the input until convergence, as shown in Formula (9).

$$Z_S^* = A_S^* + Z_S \tag{9}$$

During the space dimension post training, we use residual blocks to concatenate the output of the first layer network with the output Z_s^* of the spatial attention feature space in a spatial dimension of 7×7, which is divided into 3×3 regions in a multi-granularity space. The region forms different particles, with varying levels of information contained within each particle. The purpose of multi-granularity is to reduce the concentration of noisy sample features in a certain part of the feature map, thereby affecting the feature information of adjacent clean samples. The multi-granularity space is shown in Figure 5. After multi-granularity, the features exist in the form of particles, weakening the interference of other particles and re-extracting more significant feature information from each particle. The particle size is the size of the convolution kernel in deep convolution. Assuming that the feature map of the multi-granularity residual spectral attention model is $I' \in R^{w \times h \times c}$, deep convolution divides the feature map into several semantic markers of different granularity. w is the width of the feature map and h is the height of the feature map; c is the number of bands. T_i can be obtained by Formula (10).

$$T_i = DWConv2D_i(I'), \quad i = 1, 2, \ldots \tag{10}$$

i represents the i^{th} granularity branch, and DWConv2D represents a two-dimensional deep convolution operation. By setting the size of deep convolution, the granularity can be adjusted.

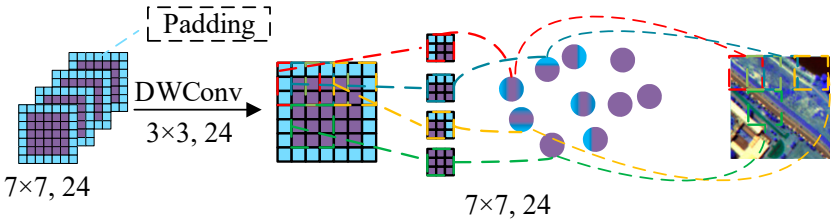


Figure 5. Multi-granularity space.

2.2. Spectral-Spatial Feature Extraction

At present, most algorithms for noise label classification only use a single network model for classification, which has lower binding force on noise labels and higher misjudgment rate compared to multiple network models. Based on the above reasons, after redesigning the architecture of spectral and spatial dimensions, this article also uses the MLP model to further obtain spectral-spatial features for the fused spectral-spatial features. The MLP model is cascaded with multiple sets of residual spectral attention models and multi-granularity residual spatial attention models to form a weakly supervised feature learning architecture, and noise labels are constrained by different models in different dimensions.

As a neural network with fewer constraints, MLP can eliminate the adverse effects of local connectivity, thus enabling the model to have strong discrimination ability for small differences in the local field of view, effectively extract deep features, achieve accurate acquisition of spectral-spatial structure information, and further reduce the interference of noisy samples in the model [31].Therefore, this paper introduces the MLP neural network as the final model for processing spectral-spatial dimensions.

In this section, first, the concat function is used to simultaneously integrate spectral and spatial information of the data, integrating dimension $128 \times 7 \times 7$ spectral information and dimension $24 \times 7 \times 7$ spatial information, resulting in dimension $152 \times 7 \times 7$ feature maps that combine spectral and spatial information, followed by the use of the average pooling layer size of 7×7 to reduce the size of feature maps while maintaining spatial information, thereby reducing the number of parameters that need to be optimized in the network, resulting in a vector size of 152×1. Finally, the vector is input to an MLP composed of the full connection layer, the GELU activation function, and the Dropout layer, and propagates forward to complete the final classification. Through the multi-layer perceptron classifier, the spectral-spatial dimension feature information can be further obtained, and the feature information of the noise label can be constrained to the greatest extent.

Next, MLP will be introduced as shown in Figure 6, which consists of three parts: full connection layer (FC), the GELU activation function, and the Dropout layer, in which the layers are fully connected. By introducing the GELU activation function to process data, when the input is negative, the input will be mapped to a non-zero value, so as to avoid the problem that some neurons of the ReLU activation function are invalid, and retain the characteristic information of the model in the negative signal, increasing the learning ability of MLP models for small differences within local features. In addition, by randomly discarding the values of 0.1% of neurons through the Dropout layer, overfitting of the model is avoided.

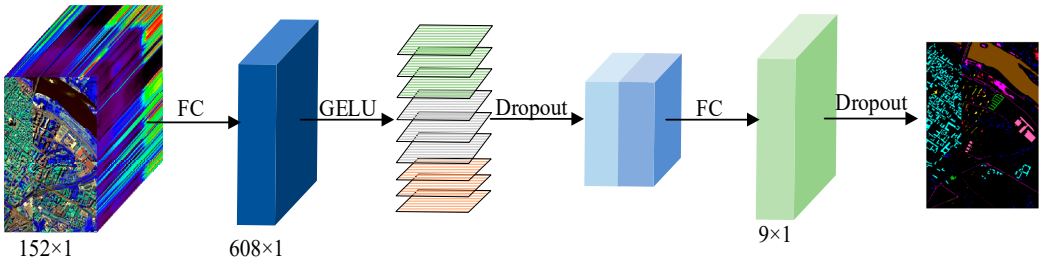


Figure 6. Multilayer perceptrons.

2.3. Lion Optimization

Hyperspectral data contain noise samples, which makes each batch of training have many confounding data points. If a small training batch is used, when the number of noise samples extracted from the batch is greater than the number of clean samples, the model will not be able to fully learn the features of clean samples. Therefore, we increase the number of batch sizes and the number of learnable samples per batch during each training phase. However, the currently popular AdamW optimizers often apply small batch sizes.

$$AdamW := \begin{cases} m_t = \beta_1 \times m_{t-1} + (1-\beta_1) \times g_t \\ v_t = \beta_2 \times v_{t-1} + (1-\beta_2) \times g_t^2 \\ \hat{m}_t = m_t / (1-\beta_1^t) \\ \hat{v}_t = v_t / (1-\beta_2^t) \\ u_t = \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon) + \lambda_t \theta_{t-1} \\ \theta_t = \theta_{t-1} - \eta_t \mu_t \end{cases} \tag{11}$$

Compared to AdamW and various adaptive optimizers that require both first-order and second-order moments to be saved simultaneously, Lion only requires momentum, reducing the additional memory footprint by half, which will be beneficial for training large models and batch sizes [32]. Therefore, this paper introduces the Lion optimizer to

simplify the process of parameter updating. Taking the t^{th} iteration of gradient descent as an example, the Lion optimizer process is shown in Formula (15).

$$Lion := \begin{cases} u_t = sign(\beta_1 \times m_{t-1} + (1-\beta_1) \times g_t) + \lambda_t \theta_{t-1} \\ \theta_t = \theta_{t-1} - \eta_t \mu_t \\ m_t = \beta_2 \times m_{t-1} + (1-\beta_2) \times g_t \end{cases} \tag{12}$$

When the input value is positive, sign is 1, and when the input value is negative, sign is -1. m_t and v_t are the first order momentum term and the second order momentum term, respectively, β_1 and β_2 are the default values of the hyperparameters with 0.9 and 0.99, the deviation correction values of m_t and v_t are $\hat{m}_t$ and $\hat{v}_t$, and g_t is the gradient of the loss function of the current sample.

3. Results

In order to verify the accuracy and efficiency of the proposed model, experiments were conducted on three datasets, and the model was evaluated using three evaluation criteria: overall accuracy (OA), average accuracy (AA), and Kappa coefficient. At the same time, this paper also studied the running time of each model to evaluate its efficiency.

3.1. The Description of Public HSI Datasets

In this paper, three widely used HSI datasets, including Pavia Center (PC) [33], WHU-Hi-LongKou (LK) [34], and HangZhou (HZ) [35], are employed in the experiments.

The Pavia Center dataset was captured by the ROSIS sensor during a flight campaign over Pavia, Northern Italy. It consists of 1906 × 715 pixels with a spatial resolution of 1.3m. After removing 13 bad bands, it has 102 bands(430~860nm). The ground truth contains nine classes representing a typical urban site. The WHU-Hi-LongKou dataset covers a simple agricultural area and was captured by an 8mm focal length steeple-wall Headwall Nano-HyperSpec sensor equipped with a receiver Matrix 600 Pro UAV platform with six kinds of crops. The image size was 550 × 400 pixels, with 270 bands ranging from 400 to 1000 nm. The Hangzhou dataset was obtained by the EO-1 Hyperion hyperspectral sensor, which kept 198 bands after removing 22 bad bands and 590 × 230 pixels. The false-color images and corresponding ground-truth maps of the three datasets can be seen in Tables 1–3.

Table 1. The Number of Samples of the PC Dataset.

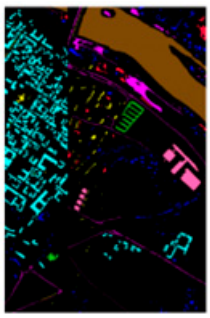
Class	Class Name	Samples	Color	False-Color Map	Ground-Truth Map
C1	Water	65,971			
C2	Trees	7598			
C3	Asphalt	3090			
C4	Self-blocking Bricks	2685			
C5	Bitumen	6584			
C6	Tiles	9248			
C7	Shadows	7287			
C8	Meadows	42,826			
C9	Bare soil	2863			
	Background				
	Total	148,152			

Table 2. The Number of Samples of the LK Dataset.

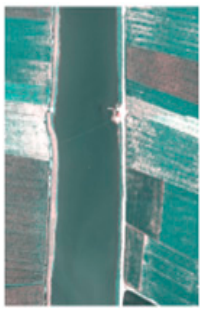
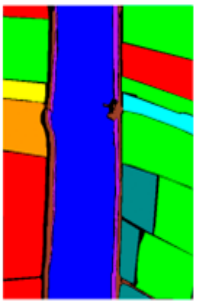
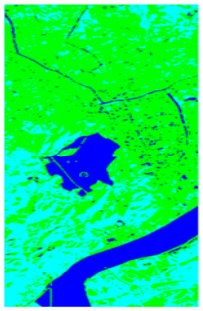
Class	Name	Samples	Color	False-Color Map	Ground-Truth Map
C1	Corn	34,511			
C2	Cotton	8374			
C3	Sesame	3031			
C4	Broad-leaf soybean	63,212			
C5	Narrow-leaf soybean	4151			
C6	Rice	11,854			
C7	Water	67,056			
C8	Roads and houses	7124			
C9	Mixed weed	5229			
	Background				
	Total	204,542			

Table 3. The Number of Samples of the HZ Dataset.

Class	Name	Samples	Color	False-Color Map	Ground-Truth Map
C1	Water	18,043			
C2	Land/building	77,450			
C3	Plants	40,207			
	Total	135,700			

3.2. Experimental Setting

The GPU server used in this article is manufactured by Finehoo Technology Co., Ltd., located in Shanghai, China. The Python version used is 3.7. The experimental environment was an Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz processor, 128 GB running memory (RAM), and NVIDIA GeForce RTX 2080Ti GPU. In addition, the deep learning framework was Pytorch and Tensorflow. The initial learning rate was set to 0.00012, the Lion optimization algorithm was used for training models, and the number of iterations was set to 6000. At the same time, in order to reduce the randomness brought by training samples, each experiment was repeated 10 times and the average accuracy is taken as the final experimental result. In order to evaluate the effectiveness of the model in this paper, the model in this paper is compared with five other algorithms, including Depthwise separable neural network (DSNN) [36], 2DCNN [37], 3DCNN [38], SSRN [39], and the advanced classification network with noisy samples, dual channel residual network (DCRN) [30]. The learning rate of DSNN, 2DCNN, 3DCNN, SSRN, and DCRN was set to 0.001, the optimizer was AdamW, the early stop method was used to train the network, and the epoch was set to 4000.

3.3. Classification Results of Different Methods

Tables 4–6, respectively, show the classification results of different methods in 24 clean samples of each class for PC, LK, and HZ data. It can be found that the OA of WSFL is

the highest, with 98.77%, 97.58%, and 80.14%, respectively. Taking HZ data as an example, WSFL increased by 0.62%, 0.95%, 3.74%, 5.63%, and 3.05% compared to DCRN, SSRN, 3DCNN, 2DCNN, and DSNN, respectively. In summary, it can be fully demonstrated that the WSFL model is still the best performing model without adding noise samples.

Table 4. The classification results of the PC dataset by different methods.

Class	DSNN	2DCNN	3DCNN	SSRN	DCRN	WSFL
C1	99.71 ± 0.09	99.60 ± 0.01	99.27 ± 0.49	99.73 ± 0.14	99.48 ± 0.10	99.81 ± 0.14
C2	90.19 ± 9.44	82.21 ± 3.70	92.25 ± 1.80	94.51 ± 4.90	92.51 ± 1.97	91.82 ± 0.95
C3	59.90 ± 12.25	95.21 ± 0.82	90.24 ± 3.44	98.19 ± 0.56	96.46 ± 0.35	97.92 ± 0.77
C4	73.76 ± 9.60	73.91 ± 9.52	86.52 ± 7.75	99.77 ± 0.09	98.82 ± 0.17	99.71 ± 0.06
C5	73.71 ± 10.33	86.72 ± 7.86	90.37 ± 2.94	85.84 ± 3.10	93.07 ± 1.83	97.59 ± 1.78
C6	67.40 ± 12.41	99.22 ± 0.06	90.95 ± 4.59	99.21 ± 0.20	99.86 ± 0.10	98.52 ± 0.39
C7	77.62 ± 5.93	67.45 ± 9.44	86.45 ± 3.79	97.36 ± 1.67	93.54 ± 5.03	95.39 ± 2.80
C8	96.33 ± 0.98	99.07 ± 0.06	96.56 ± 2.16	99.46 ± 0.34	99.74 ± 0.11	99.51 ± 0.16
C9	99.96 ± 0.09	100.00 ± 0.00	95.51 ± 0.98	100 ± 0.00	99.93 ± 0.01	98.42 ± 0.32
OA (%)	92.68 ± 1.81	95.87 ± 0.28	96.10 ± 0.43	98.60 ± 0.61	98.58 ± 0.27	98.84 ± 0.40
AA (%)	82.06 ± 3.26	89.26 ± 1.09	92.01 ± 0.54	97.11 ± 1.32	97.04 ± 0.66	97.75 ± 0.55
Kappa	89.56 ± 2.53	94.33 ± 0.39	94.54 ± 0.59	98.01 ± 0.86	98.01 ± 0.38	98.36 ± 0.32

Table 5. The classification results of the LK dataset by different methods.

Class	DSNN	2DCNN	3DCNN	SSRN	DCRN	WSFL
C1	96.97 ± 2.58	87.62 ± 6.14	95.94 ± 2.51	97.51 ± 1.02	98.94 ± 0.09	92.62 ± 0.35
C2	83.04 ± 10.08	68.48 ± 9.36	63.42 ± 8.84	94.97 ± 4.06	98.96 ± 0.92	99.27 ± 0.66
C3	68.86 ± 11.83	73.83 ± 7.90	85.10 ± 3.90	99.54 ± 0.10	99.84 ± 0.12	99.92 ± 0.06
C4	82.64 ± 10.30	70.87 ± 4.07	75.54 ± 1.52	86.71 ± 6.28	93.95 ± 2.23	98.12 ± 0.98
C5	55.38 ± 16.30	56.25 ± 10.30	74.73 ± 7.86	97.63 ± 2.06	99.37 ± 0.50	97.82 ± 1.16
C6	74.41 ± 6.93	90.11 ± 3.73	94.94 ± 4.08	99.07 ± 0.20	99.74 ± 0.08	99.13 ± 0.22
C7	99.93 ± 0.01	99.92 ± 0.02	99.84 ± 0.07	99.72 ± 0.18	99.41 ± 0.15	99.56 ± 0.30
C8	80.67 ± 12.56	89.15 ± 3.92	79.41 ± 4.48	95.38 ± 3.87	93.02 ± 2.92	96.97 ± 2.08
C9	55.37 ± 13.39	72.96 ± 8.30	83.30 ± 2.53	95.44 ± 2.08	96.51 ± 1.53	96.04 ± 1.25
OA (%)	88.79 ± 1.26	84.67 ± 2.72	88.05 ± 1.61	94.77 ± 1.50	97.34 ± 0.59	97.69 ± 0.26
AA (%)	77.47 ± 6.92	78.79 ± 6.62	83.58 ± 3.34	96.21 ± 2.36	97.74 ± 0.39	97.71 ± 0.51
Kappa	85.39 ± 1.47	80.70 ± 3.37	84.71 ± 2.06	93.25 ± 1.89	96.52 ± 0.77	96.86 ± 0.33

Table 6. The classification results of the HZ dataset by different methods.

Class	DSNN	2DCNN	3DCNN	SSRN	DCRN	WSFL
C1	81.10 ± 4.16	88.68 ± 2.65	93.92 ± 2.85	88.62 ± 1.73	89.30 ± 1.15	91.79 ± 1.29
C2	64.96 ± 10.71	71.94 ± 5.08	70.71 ± 2.67	74.35 ± 3.47	76.35 ± 4.63	77.25 ± 4.87
C3	98.65 ± 0.60	73.11 ± 10.52	79.57 ± 3.47	84.20 ± 1.09	81.32 ± 6.31	80.79 ± 2.98
OA (%)	77.07 ± 7.99	74.54 ± 4.80	76.41 ± 1.48	79.17 ± 2.00	79.52 ± 1.21	80.23 ± 1.82
AA (%)	81.57 ± 3.76	77.91 ± 1.39	81.40 ± 1.06	82.39 ± 1.01	82.32 ± 0.95	83.27 ± 0.80
Kappa	62.51 ± 10.40	59.19 ± 4.75	60.66 ± 2.32	64.95 ± 3.05	65.41 ± 1.77	66.51 ± 2.19

3.3.1. Results of PC Datasets with Different Numbers of Noise Samples

The results of using different methods to classify PC datasets are shown in Table 7. In the PC dataset, 24 samples were taken from each class of clean samples and four, eight, and 12 noisy samples were taken from each class to verify the processing ability of different deep learning models. It was found that WSFL had the best overall classification results, reaching 98.52%, 97.50%, and 96.77%, respectively. In addition, the number of training samples selected in this paper accounts for approximately 0.1944% of the total sample size. Compared with the approximately 3% training sample size required for other popular

depth models, the sample size required in this paper is greatly reduced, which can also prove that the model proposed has a reduced dependence on labeled samples.

Table 7. The classification results of the PC dataset with 24 clean + 4/8/12 noisy samples.

Class	The Number of Clean and Noisy Training Samples											
	24(clean) + 4(noisy)				24(clean) + 8(noisy)				24(clean) + 12(noisy)			
	3DCNN	**SSRN**	**DCRN**	**WSFL**	**3DCNN**	**SSRN**	**DCRN**	**WSFL**	**3DCNN**	**SSRN**	**DCRN**	**WSFL**
C1	98.33	98.88	98.35	99.87	94.07	97.32	97.49	99.53	86.15	94.56	98.14	98.23
Std	0.84	0.54	0.53	0.12	3.70	1.39	1.31	0.15	3.39	4.10	0.12	0.10
C2	82.22	94.75	93.13	96.00	70.41	91.17	92.51	91.07	54.34	89.30	91.00	92.83
Std	1.62	2.46	2.84	2.47	3.62	4.99	3.03	2.81	10.35	6.78	4.53	2.39
C3	82.15	89.77	91.70	90.74	74.38	87.84	94.12	94.22	62.42	63.44	93.89	86.32
Std	2.41	6.92	0.48	1.46	5.95	6.90	1.98	1.30	12.65	4.71	2.30	3.50
C4	79.01	95.62	96.28	98.81	52.66	88.30	91.01	97.99	44.74	91.00	94.50	92.34
Std	3.59	1.63	3.21	1.08	3.15	7.29	4.08	1.42	1.73	8.79	2.29	3.25
C5	75.53	92.55	96.29	96.91	62.45	92.59	89.34	91.02	47.53	85.03	92.03	92.04
Std	6.29	3.30	2.67	2.33	10.71	2.33	1.74	2.27	10.01	8.64	4.32	2.91
C6	81.60	94.42	96.04	98.21	62.99	89.92	98.96	98.28	50.35	81.70	95.73	98.33
Std	7.80	2.05	1.06	0.74	8.02	2.37	0.76	1.25	3.66	1.94	2.62	1.22
C7	72.25	94.70	94.78	92.42	61.85	91.19	93.62	90.81	52.28	77.83	93.59	91.84
Std	2.43	3.28	1.07	1.95	5.57	4.77	1.07	1.92	9.03	5.02	2.06	2.37
C8	83.54	91.24	97.06	98.77	68.04	91.69	96.48	97.54	56.75	90.38	93.28	97.49
Std	6.46	0.61	1.25	0.19	4.14	1.59	2.09	0.32	4.26	2.47	5.06	1.00
C9	91.95	97.60	98.39	98.67	79.55	95.94	99.57	99.62	75.85	87.68	91.16	96.36
Std	7.20	1.46	0.32	0.58	8.09	3.72	0.13	0.07	6.19	7.12	5.91	2.13
OA(%)	89.09	95.42	97.12	98.52	78.98	94.02	96.34	97.50	68.98	90.20	95.44	96.77
Std	2.79	0.28	0.80	0.45	3.93	0.95	1.95	0.12	2.00	2.56	3.82	0.71
AA(%)	82.95	94.39	95.78	96.71	69.60	91.77	94.79	95.57	58.93	84.55	93.70	93.97
Std	2.53	1.31	1.60	1.34	3.50	1.21	0.79	0.40	1.31	0.70	4.42	1.81
Kappa	84.82	93.69	95.82	97.91	71.44	91.68	94.87	96.48	58.64	86.40	93.62	95.48
Std	3.68	0.40	1.13	0.64	4.85	1.31	2.61	0.17	1.21	3.33	5.99	1.00

Table 7 shows that in 24 clean samples and four noisy samples, the OA of the proposed WSFL model reached 98.52%, which is 1.4%, 3.1%, and 9.43% higher than DCRN, SSRN, and 3DCNN, respectively. Among 24 clean samples and eight noisy samples, our OA reached 97.50%, which was 1.16%, 3.48%, and 17.52% higher than DCRN, SSRN, and 3DCNN, respectively. Among 24 clean samples and 12 noisy samples, our OA reached 96.77%, which was 1.33%, 6.57%, and 27.79% higher than DCRN, SSRN, and 3DCNN, respectively. In summary, it can be found that WSFL significantly improves OA under different noise sample sizes. Although the method proposed in this paper cannot achieve the best accuracy for each class, out of 24 clean samples and four noisy samples, seven classes are the best in this paper. Out of 24 clean samples and eight noise samples, six categories are the best category, and out of 24 clean samples and 12 noise samples, six categories are the best category. This can also prove that WSFL can better handle noisy samples compared to models such as DCRN, SSRN, and 3DCNN. As the number of noisy samples increases, the OA of WSFL decreases by only about 1%, which is acceptable as a multiple of the number of noisy samples. This fully demonstrates the effectiveness of WSFL in HSI classification tasks with noisy labels.

Finally, Figure 7 shows the pseudo-color images of the classification results of various classification methods in the PC dataset. False-color images, as a subjective evaluation indicator, can more intuitively display the classification effect. From Figure 7, it can be seen that WSFL has a significant improvement in classification performance compared to DCRN, SSRN, and 3DCNN. The area of misclassification is greatly reduced, and it is closer to the true distribution of ground objects.

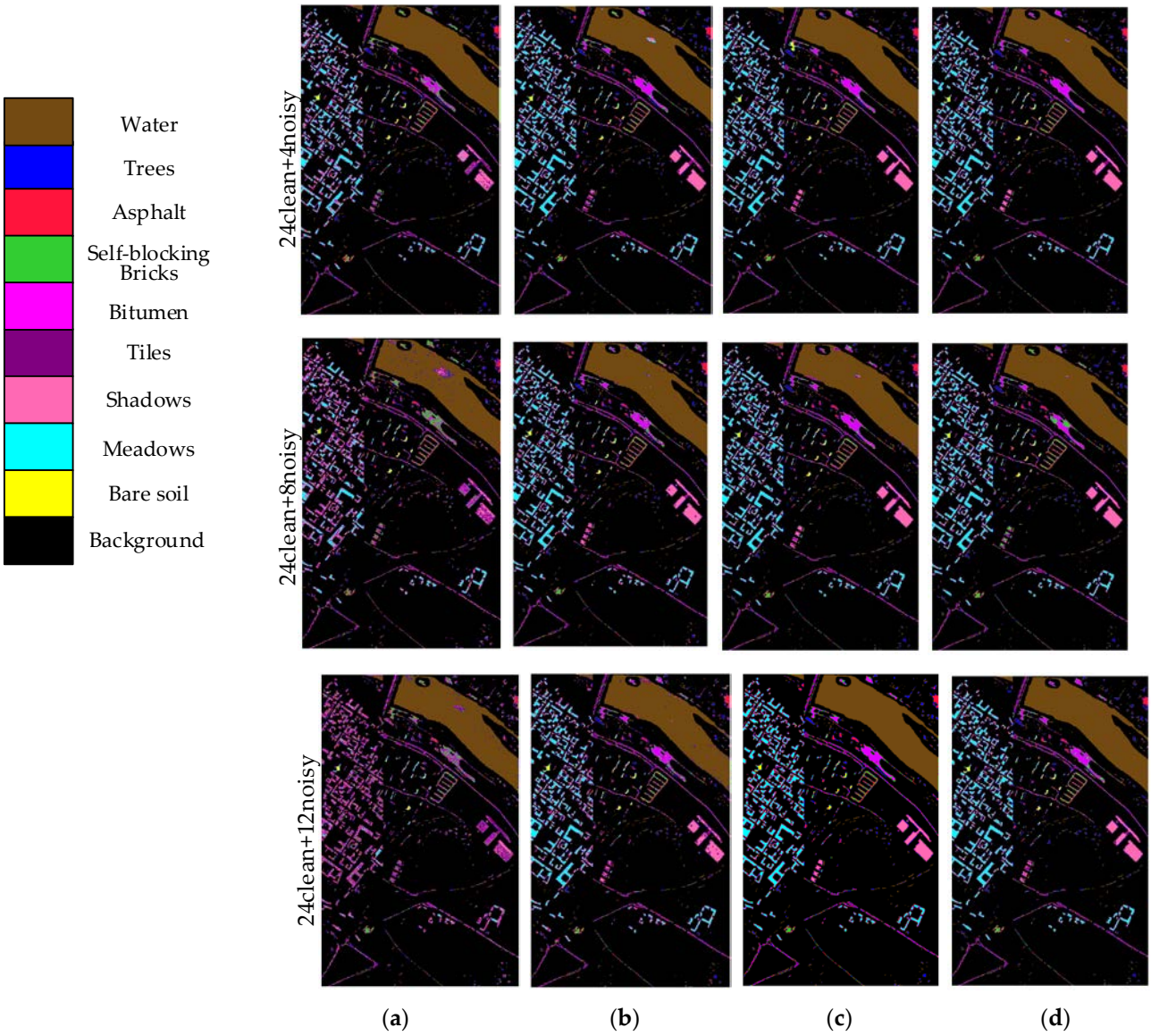


Figure 7. The classification results of PC dataset. (**a**) 3DCNN; (**b**) SSRN; (**c**) DCRN; (**d**) WSFL.

3.3.2. Results of LK Datasets with Different Numbers of Noise Samples

The classification results of LK dataset are shown in Table 8. There are abundant annotated samples in each category of the LK dataset, so all methods perform well. Compared to other models, the proposed WSFL model achieved the best classification performance, showing significant improvements in most categories. Although the method proposed in this paper cannot achieve the best accuracy for each class, out of 24 clean samples and four noisy samples, as well as out of 24 clean samples and eight noisy samples, all six classes are considered the best class. Among the 24 clean samples and 12 noisy samples, seven were the optimal categories, which also proves that WSFL can better handle noisy samples compared to models such as DCRN, SSRN, and 3DCNN, fully demonstrating the effectiveness of WSFL in HSI classification tasks with noisy labels.

Table 8. The classification results of the LK dataset with 24 clean + 4/8/12 noisy samples.

Class	The Number of Clean and Noisy Training Samples											
	24(clean) + 4(noisy)				24(clean) + 8(noisy)				24(clean) + 12(noisy)			
	3DCNN	**SSRN**	**DCRN**	**WSFL**	**3DCNN**	**SSRN**	**DCRN**	**WSFL**	**3DCNN**	**SSRN**	**DCRN**	**WSFL**
C1	79.28	94.03	95.79	94.97	60.90	88.49	90.38	94.18	50.76	80.23	87.94	92.46
Std	9.23	1.45	2.22	1.13	9.68	4.86	4.84	3.62	7.42	1.41	4.02	3.20
C2	47.53	92.53	89.40	92.86	44.77	79.26	90.30	81.54	46.53	74.26	86.31	82.57
Std	7.56	4.32	5.61	2.45	7.54	11.0	7.06	2.97	3.98	7.27	8.84	4.42
C3	67.64	91.49	98.63	97.73	55.33	90.50	89.52	90.73	50.34	83.88	86.63	91.29
Std	8.18	1.60	0.81	1.93	10.02	5.67	5.90	3.41	7.55	4.73	9.83	5.49
C4	61.17	89.50	91.42	93.33	48.21	83.85	81.86	82.05	42.85	79.11	77.52	76.35
Std	6.26	5.24	4.13	5.07	6.79	5.38	6.45	4.49	3.18	5.57	3.02	2.35
C5	64.15	89.74	89.20	94.42	44.40	83.49	86.42	91.07	45.92	75.83	81.88	88.31
Std	9.81	3.60	4.23	1.27	6.80	8.73	6.79	3.85	10.05	7.76	8.45	7.01
C6	82.62	96.51	98.55	94.13	62.05	92.57	91.83	93.86	57.35	89.79	80.19	82.72
Std	7.59	1.60	1.06	1.20	6.20	5.34	3.24	1.17	8.55	5.14	6.15	4.55
C7	98.63	98.20	98.38	99.70	96.11	97.11	98.23	99.57	92.59	95.01	96.64	99.32
Std	0.49	1.06	0.66	0.02	0.83	1.98	0.55	0.18	1.20	4.96	2.31	0.29
C8	71.44	87.52	91.45	92.34	58.44	77.55	82.76	83.93	52.18	70.39	76.85	78.51
Std	3.38	4.91	2.54	1.91	7.05	7.59	7.60	3.60	2.46	8.34	11.07	8.09
C9	57.81	88.09	89.36	89.52	47.52	77.32	82.75	80.71	42.77	70.33	73.01	75.50
Std	10.65	3.83	1.06	2.71	6.75	3.51	4.12	2.50	6.20	5.61	6.31	2.55
OA(%)	77.63	93.58	94.78	95.68	67.09	89.01	89.85	90.85	61.99	84.42	86.15	87.74
Std	3.92	2.00	1.42	0.71	3.80	2.85	3.01	1.58	2.87	2.06	3.73	1.85
AA(%)	70.03	91.96	93.58	94.33	57.52	85.57	88.23	88.63	53.48	79.87	83.00	85.23
Std	4.46	1.07	1.25	1.06	2.50	2.82	3.87	2.26	4.25	0.32	4.42	2.78
Kappa	71.73	91.67	93.22	94.36	58.94	85.82	86.89	88.18	53.07	79.98	82.23	84.28
Std	4.72	2.53	1.82	0.88	4.47	3.62	3.78	2.01	3.48	2.49	4.64	2.39

Finally, Figure 8 shows the pseudo-color images of various classification methods on LK dataset. Taking 24 clean + four noise as an example, the broad-leaf soybeans in the middle part have severe classification confusion. Compared with the DCRN, SSRN, and 3DCNN, the model proposed in this paper has fewer misclassification phenomena and is closer to the true distribution of ground objects.

3.3.3. Results of HZ Datasets with Different Numbers of Noise Samples

The HZ dataset has the characteristics of small inter-class differences and large intra-class differences. Therefore, the accuracy rates of various methods are relatively low, among which the indicators of WSFL are at the best, and OA reaches 79.44%, 72.90%, and 63.57%, respectively. The classification results of the HZ dataset are shown in Table 9.

Table 9 shows that in 24 clean samples and four noisy samples, the OA of this article reached 79.44%, which is 1.93%, 2.37%, and 4.72% higher than DCRN, SSRN, and 3DCNN, respectively. Among 24 clean samples and 12 noisy samples, our OA reached 63.57%, which was 1.26%, 2.39%, and 4.38% higher than DCRN, SSRN, and 3DCNN, respectively. For WSFL, out of 24 clean samples and four noisy samples, one class is the best class in this article. Among 24 clean samples and eight noise samples, as well as 24 clean samples and 12 noise samples, there are two optimal categories.

Finally, Figure 9 shows the pseudo-color images of the classification results on the HZ dataset. Compared to DCRN, SSRN, and 3DCNN, WSFL has the smallest staggered area of water.

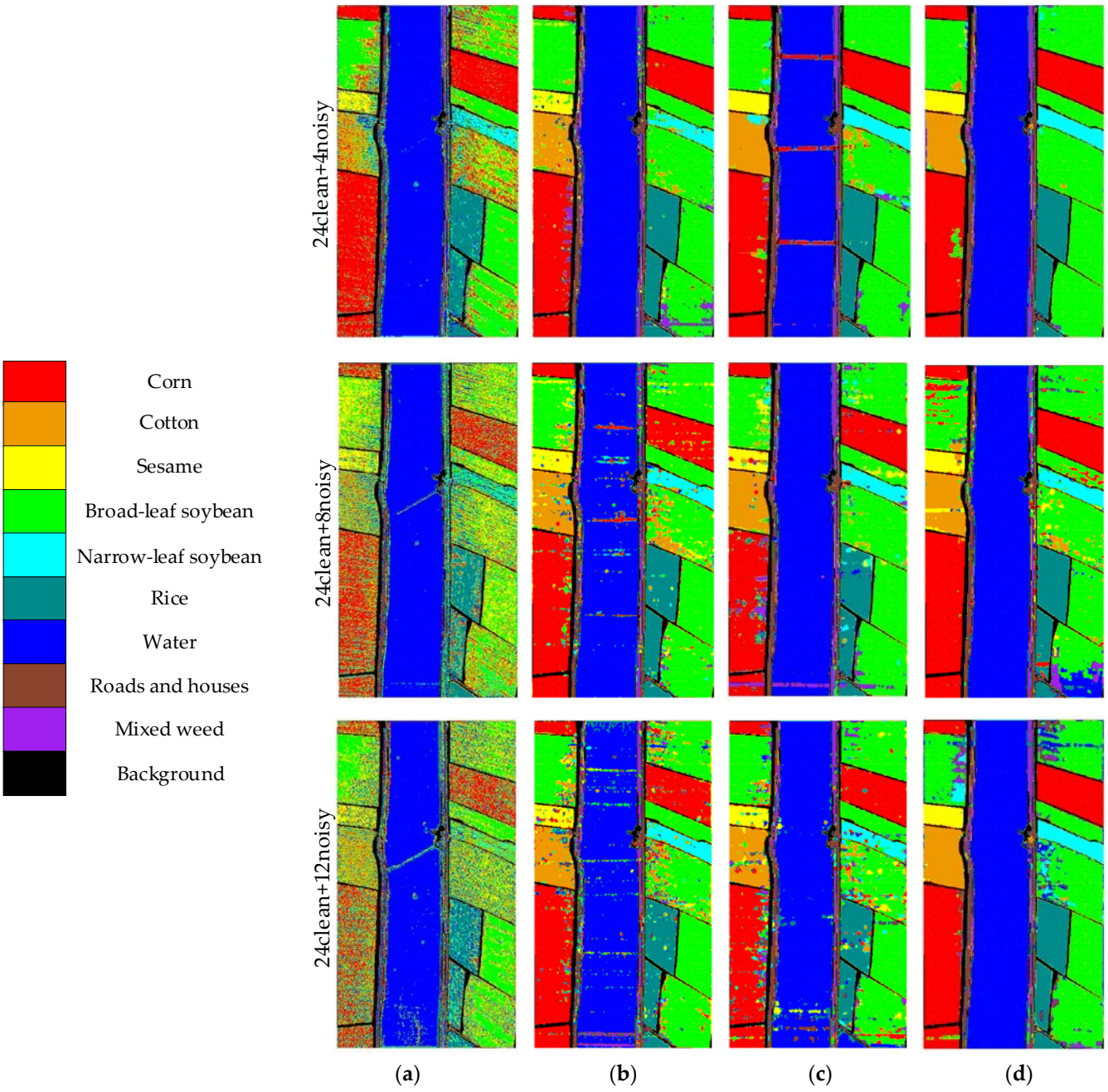


Figure 8. The classification results of the LK dataset. (**a**) 3DCNN; (**b**) SSRN; (**c**) DCRN; (**d**) WSFL.

3.4. The Numbers of Clean and Noisy Samples

In the classification with noisy labels, the number of clean samples is crucial, and even when there are few available clean samples, the proposed WSFL framework can perform well in classification. As shown in Tables 10–12, when the number of clean samples in each category is limited, the proposed WSFL still has relatively good performance. Compared with DCRN, SSRN, and 3DCNN, the WSFL model has a more robust network structure, resulting in a significant improvement in performance. In addition, as the number of noise samples increases, WSFL exhibits a slow performance decline. Therefore, compared to other methods, WSFL has higher stability.

Table 9. The classification results of the HZ dataset with 24 clean + 4/8/12 noisy samples.

Class	The Number of Clean and Noisy Training Samples											
	24(clean) + 4(noisy)				24(clean) + 8(noisy)				24(clean) + 12(noisy)			
	3DCNN	**SSRN**	**DCRN**	**WSFL**	**3DCNN**	**SSRN**	**DCRN**	**WSFL**	**3DCNN**	**SSRN**	**DCRN**	**WSFL**
C1	86.43	81.07	81.43	88.37	77.53	71.54	73.39	83.05	77.06	75.88	69.43	81.33
Std	4.55	9.46	4.13	3.86	5.09	8.98	5.81	3.34	4.86	8.70	7.41	3.20
C2	71.20	79.61	76.85	78.97	60.06	58.05	64.80	73.07	51.85	57.98	59.52	62.45
Std	5.98	5.96	5.35	2.24	8.90	5.60	3.49	2.39	8.49	9.17	8.70	4.57
C3	76.26	70.40	77.03	76.35	62.12	75.63	67.95	68.03	65.33	60.75	64.51	57.76
Std	2.13	8.89	7.67	3.26	7.36	5.30	7.68	3.82	8.46	9.60	8.10	4.37
OA(%)	74.72	77.07	77.51	79.44	62.99	65.05	66.88	72.90	59.19	61.18	62.31	63.57
Std	3.90	5.51	2.87	1.19	6.17	3.31	2.58	1.96	6.37	3.62	3.42	3.28
AA(%)	77.96	77.03	78.44	81.23	66.57	68.41	68.71	74.72	64.75	64.87	64.49	67.18
Std	2.78	6.04	2.90	1.86	4.28	1.70	1.40	0.94	4.90	1.52	2.04	2.02
Kappa	57.61	60.30	61.49	64.84	40.81	44.08	68.71	54.69	36.00	37.42	39.46	40.32
Std	5.95	8.62	4.39	2.29	8.50	4.15	3.52	2.39	8.62	2.94	3.90	3.01

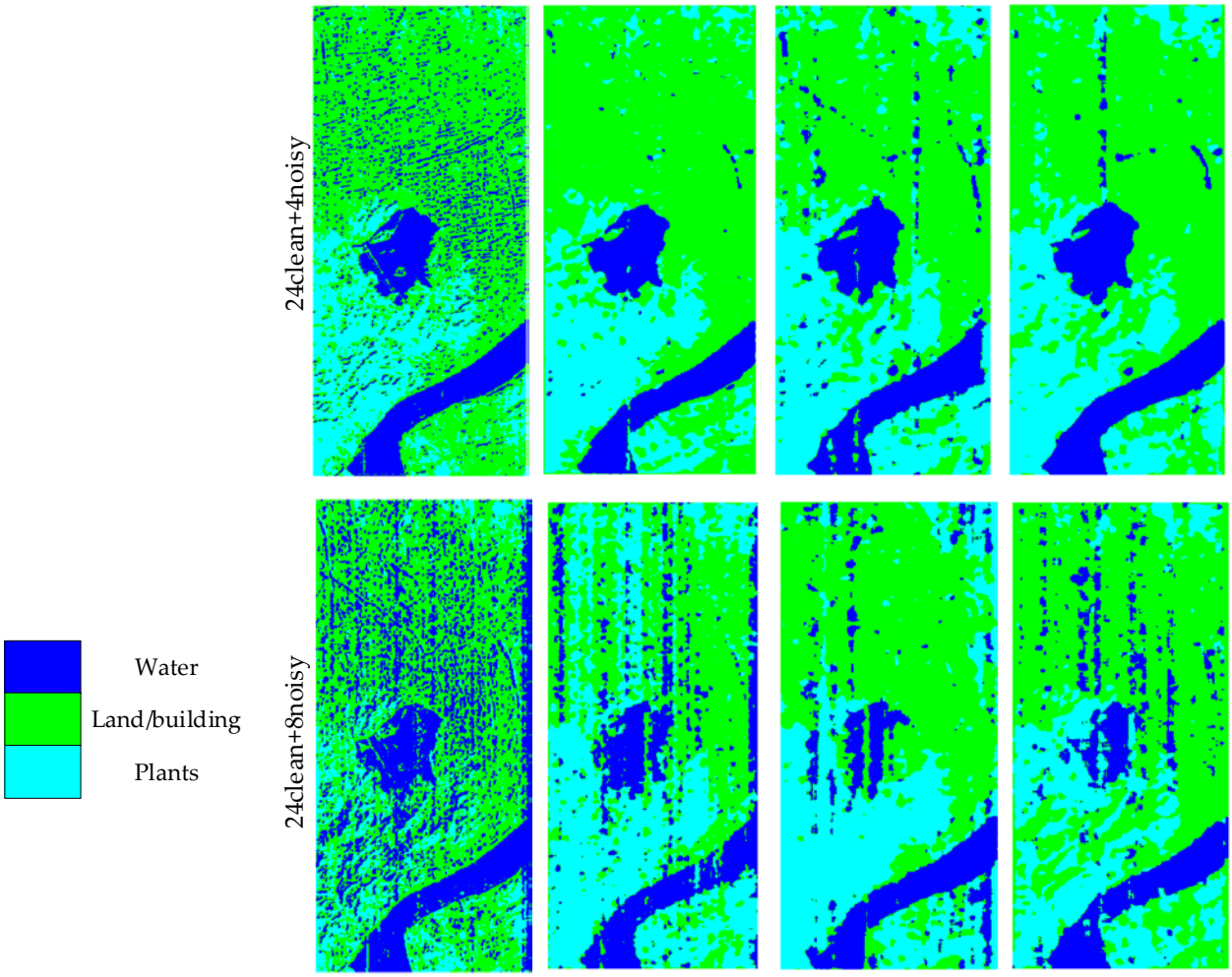


Figure 9. *Cont.*

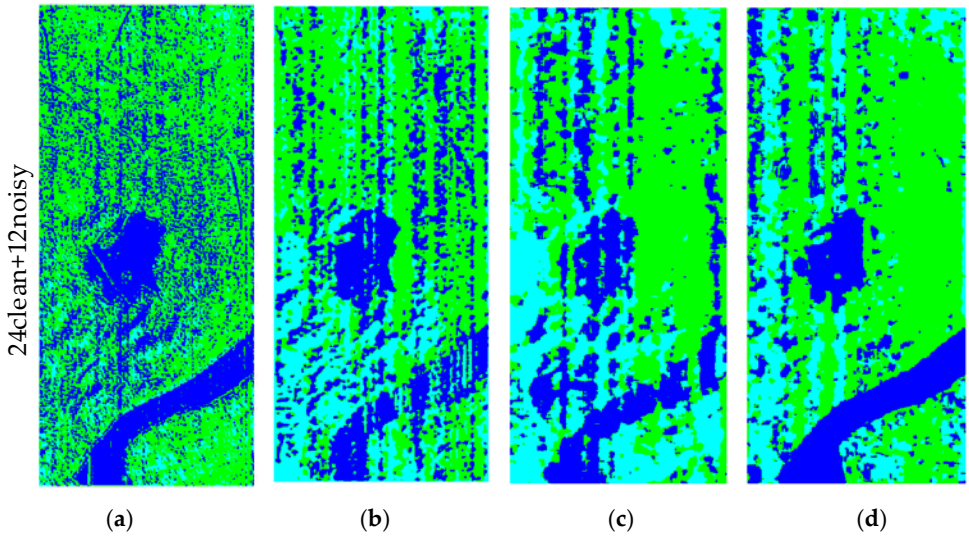


Figure 9. The classification results of the HZ dataset. (**a**) 3DCNN; (**b**) SSRN; (**c**) DCRN; (**d**) WSFL.

Table 10. The classification results of the PC dataset with different numbers of clean samples and noisy samples.

Class	The Numbers of Clean and Noisy Training Samples											
	8(clean) + 4(noisy)				8 (clean) + 8(noisy)				8(clean) + 12(noisy)			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	67.60	84.09	90.96	94.02	52.78	79.55	82.05	85.92	41.50	61.68	67.08	71.35
Std	3.91	4.80	4.61	3.03	4.96	3.04	5.41	4.61	5.11	4.97	8.72	7.16
AA(%)	57.81	79.92	84.36	86.78	42.80	70.72	67.39	85.26	32.65	55.81	57.90	64.29
Std	7.67	7.97	3.06	3.21	5.97	3.57	3.63	2.79	5.08	1.63	6.96	6.02
Kappa	57.05	79.01	87.35	91.53	40.60	72.31	75.21	80.74	27.63	50.26	56.60	62.51
Std	5.15	6.47	6.21	4.05	4.71	3.36	7.14	5.14	5.22	5.45	10.37	8.15
	20(clean) + 4(noisy)				20(clean) + 8(noisy)				20(clean) + 12(noisy)			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	82.16	94.41	96.76	97.61	73.48	93.36	93.92	96.63	62.58	86.14	91.82	96.38
Std	2.50	0.81	0.61	0.94	0.46	1.55	3.77	2.05	3.92	1.06	6.25	2.43
AA(%)	73.83	92.68	95.17	94.04	63.30	89.16	89.89	94.65	54.09	79.85	88.01	93.49
Std	2.56	1.85	0.86	1.22	1.01	1.44	3.49	1.77	1.67	1.88	6.64	3.01
Kappa	75.58	92.15	95.43	96.61	64.24	90.69	91.51	95.25	51.74	81.00	88.69	94.89
Std	3.44	1.15	0.87	1.32	0.30	2.10	5.12	2.80	4.43	1.44	8.54	3.39
	24(clean) + 4(noisy)				24(clean) + 8(noisy)				24(clean) + 12(noisy)			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	89.09	95.42	97.12	98.52	78.98	94.02	96.34	97.50	68.98	90.20	95.44	96.77
Std	2.79	0.28	0.80	0.45	3.93	0.95	1.95	0.12	2.00	2.56	3.82	0.71
AA(%)	82.95	94.39	95.78	96.71	69.60	91.77	94.79	95.57	58.93	84.55	93.70	93.97
Std	2.53	1.31	1.60	1.34	3.50	1.21	0.79	0.40	1.31	0.70	4.42	1.81
Kappa	84.82	93.69	95.82	97.91	71.44	91.68	94.87	96.48	58.64	86.40	93.62	95.48
Std	3.68	0.40	1.13	0.64	4.85	1.31	2.61	0.17	1.21	3.33	5.99	1.00

Table 11. The classification results of the LK dataset with different numbers of clean samples and noisy samples.

Metrics	The Numbers of Clean and Noisy Training Samples											
	8(clean) + 4(noisy)				8 (clean) + 8(noisy)				8(clean) + 12(noisy)			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	58.65	80.42	80.58	84.63	51.42	66.12	66.40	68.17	45.25	55.71	56.87	60.93
Std	2.84	3.72	5.35	3.39	5.28	7.23	4.74	3.79	4.54	5.47	3.71	3.54
AA(%)	46.53	74.66	78.51	78.45	36.50	60.84	65.69	66.81	32.91	47.92	51.97	52.57
Std	2.73	2.33	1.67	2.86	4.83	4.15	3.18	4.05	2.73	4.81	3.09	2.62
Kappa	49.13	75.17	75.55	80.39	40.69	58.16	58.63	60.50	34.25	46.18	47.02	52.10
Std	2.96	4.38	6.37	4.02	6.01	8.04	5.56	4.02	4.61	5.94	3.93	3.74
Metrics	**20(clean) + 4(noisy)**				**20(clean) + 8(noisy)**				**20(clean) + 12(noisy)**			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	69.08	87.52	92.12	94.83	66.98	79.70	87.63	89.54	55.47	74.68	81.10	85.21
Std	2.20	4.74	2.60	1.74	3.09	3.76	6.29	2.79	5.70	6.44	5.51	2.52
AA(%)	61.24	86.80	91.58	92.19	55.51	77.96	85.63	88.54	45.17	70.92	79.17	80.63
Std	3.14	1.20	1.19	1.06	1.02	3.49	4.33	2.15	5.82	3.57	6.98	2.97
Kappa	61.44	84.03	89.82	93.27	58.50	74.53	84.10	86.41	45.17	68.13	76.02	81.07
Std	2.56	5.58	3.31	2.22	3.37	4.69	7.87	3.57	6.63	7.64	7.04	3.06
Metrics	**24(clean) + 4(noisy)**				**24(clean) + 8(noisy)**				**24(clean) + 12(noisy)**			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	77.63	93.58	94.78	95.68	67.09	89.01	89.85	90.85	61.99	84.42	86.15	87.74
Std	3.92	2.00	1.42	0.71	3.80	2.85	3.01	1.58	2.87	2.06	3.73	1.85
AA(%)	70.03	91.96	93.58	94.33	57.52	85.57	88.23	88.63	53.48	79.87	83.00	85.23
Std	4.46	1.07	1.25	1.06	2.50	2.82	3.87	2.26	4.25	0.32	4.42	2.78
Kappa	71.73	91.67	93.22	94.36	58.94	85.82	86.89	88.18	53.07	79.98	82.23	84.28
Std	4.72	2.53	1.82	0.88	4.47	3.62	3.78	2.01	3.48	2.49	4.64	2.39

Tables 10–12 show the impact of different numbers of clean and noisy samples on the PC, LK, and HZ datasets. Taking the PC dataset as an example, the OA of the WSFL model reached 98.52%, 97.50%, and 96.77%, respectively. Although the OA of WSFL is also decreasing, compared to other models this model has higher performance. When the number of noise samples is fixed, the value of OA continuously increases as the number of clean samples increases. By comparing different clean sample numbers and noise sample numbers, it can be found that when the clean sample number is 24 and the noise sample number is four, WSFL has the optimal indicator. In summary, the model in this paper has better processing ability when dealing with noisy samples, which can also prove that WSFL has a more robust network structure and stronger feature learning ability.

3.5. Investigation of Running Time

Table 13 gave the computation time comparison for three HSI datasets. In addition, compared to complex models such as DCRN and SSRN, WSFL is 4.47s faster than DCRN in PC datasets and 27.4s slower than SSRN. In summary, compared to single model neural networks such as DSNN, 2DCNN, 3DCNN, etc., WSFL has a significant improvement in performance despite being slower in time, and the model has stronger anti-interference ability against noise labels. Secondly, compared to complex models such as DCRN and SSRN, it ranks second on the PC and LK datasets and third on the HZ dataset. Considering the balance between accuracy and efficiency, WSFL as proposed in this paper is optimal.

Table 12. The classification results of the HZ dataset with different numbers of clean samples and noisy samples.

Metrics	The Numbers of Clean and Noisy Training Samples											
	8(clean) + 4(noisy)				8(clean) + 8(noisy)				8(clean) + 12(noisy)			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	58.03	62.36	62.82	66.26	41.66	44.89	48.66	54.97	35.53	35.68	36.60	43.34
Std	4.59	5.29	7.45	5.06	5.94	4.76	8.29	3.30	8.79	4.51	6.92	5.26
AA(%)	63.96	62.26	67.65	65.43	46.75	47.44	50.83	63.48	38.35	36.71	40.64	40.23
Std	3.85	3.84	7.73	5.74	6.77	7.91	10.2	3.62	6.02	5.41	7.52	4.79
Kappa	34.77	38.11	42.32	44.69	10.06	16.30	21.04	32.64	5.82	6.44	7.36	11.18
Std	6.43	6.51	10.5	6.12	8.20	7.94	11.3	4.89	4.46	4.95	8.21	4.47
Metrics	**20(clean) + 4(noisy)**				**20(clean) + 8(noisy)**				**20(clean) + 12(noisy)**			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	68.35	71.09	73.21	77.64	58.25	64.82	65.64	66.62	58.03	59.73	61.70	61.96
Std	4.25	6.25	5.03	4.71	3.25	4.67	5.48	4.94	3.72	4.73	4.47	3.27
AA(%)	71.75	75.48	75.65	80.65	63.30	68.48	70.13	70.96	62.53	61.21	62.39	65.68
Std	2.43	3.69	2.70	2.55	3.40	2.11	3.96	3.09	2.70	2.43	4.54	3.97
Kappa	49.23	52.84	55.43	62.26	33.69	43.07	70.13	45.84	33.84	34.69	37.66	38.13
Std	6.00	7.93	6.27	5.71	4.90	5.91	7.71	5.49	5.36	5.33	6.36	4.82
Metrics	**24(clean) + 4(noisy)**				**24(clean) + 8(noisy)**				**24(clean) + 12(noisy)**			
	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL	3DCNN	SSRN	DCRN	WSFL
OA(%)	74.72	77.07	77.51	79.44	62.99	65.05	66.88	72.90	59.19	61.18	62.31	63.57
Std	3.90	5.51	2.87	1.19	6.17	3.31	2.58	1.96	6.37	3.62	3.42	3.28
AA(%)	77.96	77.03	78.44	81.23	66.57	68.41	68.71	74.72	64.75	64.87	64.49	67.18
Std	2.78	6.04	2.90	1.86	4.28	1.70	1.40	0.94	4.90	1.52	2.04	2.02
Kappa	57.61	60.30	61.49	64.84	40.81	44.08	68.71	54.69	36.00	37.42	39.46	40.32
Std	5.95	8.62	4.39	2.29	8.50	4.15	3.52	2.39	8.62	2.94	3.90	2.39

Table 13. Computation time comparison for three HSI datasets(s).

Dataset \ Algorithm	DSNN	2DCNN	3DCNN	SSRN	DCRN	WSFL
Pavia Center	203.60	207.69	161.21	241.27	273.14	268.67
WHU-Hi-LongKou	89.74	84.97	298.79	390.17	330.76	375.81
HangZhou	56.06	55.83	118.26	151.64	151.92	158.27

4. Discussion

4.1. Effectiveness of the Attention Model

In order to verify the effectiveness of MGRSAM and MRSAM on WSFL, this paper conducted ablation experiments and compared the OA, AA, and Kappa coefficients of MGRSAM, MRSAM, and MGRSAM + MRSAM as shown in Table 14. It can be seen that the simultaneous presence of MGRSAM and MRSAM has indeed improved OA, AA, and Kappa on the three datasets. However, on the WHU-Hi-LongKou dataset, the AA value of MRSAM is 0.26% higher than that of MGRSAM and MRSAM, as the accuracy of a certain class of MRSAM is slightly higher than that of the final method. Although the AA value of MRSAM has increased on the WHU-Hi-LongKou dataset, compared by OA, Kappa, and overall, the coexistence of MGRSAM and MRSAM is superior.

Table 14. The classification effectiveness of different attention models.

Dataset \ Algorithm	Index	MGRSAM	MRSAM	MGRSAM + MRSAM
Pavia Center	OA	98.27	98.26	98.52
	AA	96.47	96.54	96.71
	Kappa	97.55	97.54	97.91
WHU-Hi-LongKou	OA	95.34	95.28	95.68
	AA	91.02	94.59	94.33
	Kappa	93.93	93.86	94.36
HangZhou	OA	77.78	78.91	79.44
	AA	79.14	80.13	81.23
	Kappa	62.01	81.23	64.84

4.2. Effectiveness of the MLP Model

In order to verify the effectiveness of the MLP model in the model proposed in this paper, ablation experiments were conducted as shown in Table 15. It can be observed that after adding the MLP model, OA, AA and Kappa show a significant increase.

Table 15. The effectiveness of the MLP model on classification results.

Dataset \ Algorithm	Index	Without MLP	With MLP
Pavia Center	OA	98.23	98.52
	AA	96.34	96.71
	Kappa	97.50	97.91
WHU-Hi-LongKou	OA	95.01	95.68
	AA	94.11	94.33
	Kappa	93.21	94.36
HangZhou	OA	78.48	79.44
	AA	78.38	81.23
	Kappa	62.06	64.84

4.3. Effectiveness of the Number of Groups on the Model

In this section, in order to verify the impact of the number of groups in MGRSAM on the WSFL model, this paper selects two, three, four, six, and eight groups for comparison. From Figure 10, it can be seen that when the number of groups is three, OA, AA, and Kappa reach their highest values. Although the AA of three groups in the PC dataset is slightly lower than the AA of two groups, the improvement in OA, Kappa, and the value of three in the LK and HZ data are all in the optimal solution, which is acceptable. In addition, when the group value is changed from three to six, the overall indicator shows a decline phenomenon, which is because the features are too scattered in the later training of the spectral dimension. Although the model's ability to fit noisy labels is significantly reduced, it also reduces the ability to fit clean samples. Therefore, this paper selects three groups to train our model, in order to reduce the fitting of noisy samples while retaining the ability to fit clean samples.

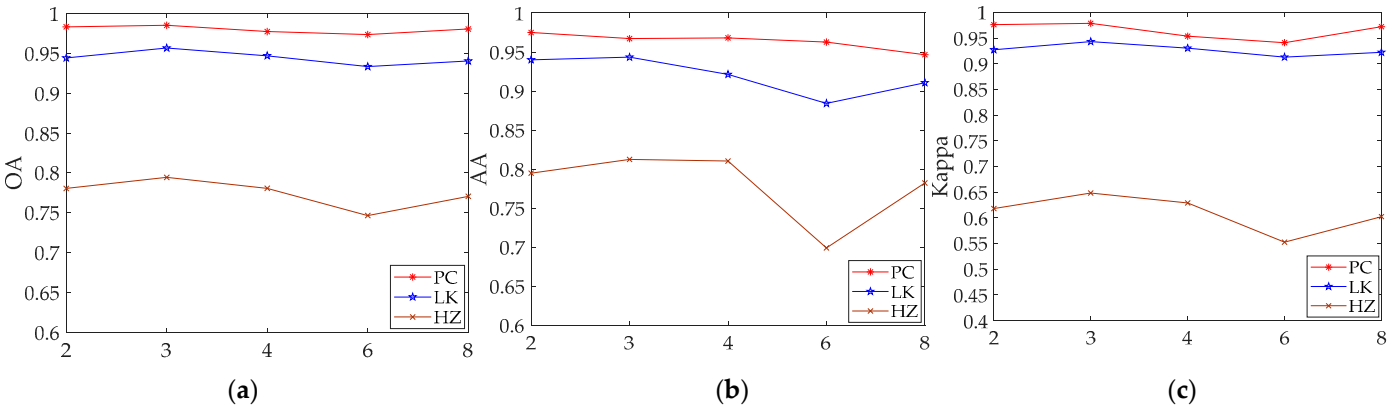


Figure 10. The effect of group on classification accuracy. (**a**) OA; (**b**) AA; (**c**) Kappa.

5. Conclusions

In this article, we propose WSFL, a novel weakly supervised feature learning architecture with the core goal of exploring the robustness of the model to different noise levels. The uniqueness of WSFL lies in its specific feature learning strategy for noisy labels, where it can adaptively learn features through multi-model attention adaptive feature learning without removing noisy samples. This preserves the diversity of features and reduces the influence of noisy samples on the model.

In addition, different architectures have been designed based on the characteristics of hyperspectral data in spectral, spatial, and spectral-spatial dimensions. Compared with other methods, WSFL can effectively capture information in hyperspectral data and transform it into discriminative feature representations. Specifically, multiple sets of residual spectral attention models were carefully designed in the spectral dimension, which differentiated features through multiple sets of spectral feature spaces to avoid excessive concentration of single layer spectral features and memory of noisy samples. Secondly, more clean spectral features were learned in the spectral attention space. In addition, a multi-granularity residual spatial attention model has been carefully designed in the spatial dimension. In the spatial feature attention space, the similarity between samples is calculated to reduce the weight of noisy samples and improve the influence of clean samples. Then, the spatial features are refined in the multi-granularity space to obtain more discriminative spatial features, improving the quality of capturing spatial features and enhancing the model's constraint on noise samples. Finally, the MLP model is introduced to eliminate the adverse effects of local connectivity in the model, obtaining more spatial structure information from the HSI dataset.

A large number of experimental results indicate that the framework proposed in this article surpasses state-of-the-art algorithms and can still achieve good accuracy even in the presence of a large number of noisy samples. Therefore, the architecture of this article is more suitable for HSI classification with noisy labels. The future work direction of this article is to apply the proposed framework to other hyperspectral images, rather than just processing the aforementioned open-source datasets, in order to enhance the universality of the model in practice.

Author Contributions: Conceptualization, C.L., H.W. and L.Z.; methodology, software, validation, C.L.; writing—review and editing, H.W., L.Z. and C.L.; supervision, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the High-end Foreign Experts Introduction Program (G2022012010L) and Heilongjiang Natural Science Foundation Project (LH2023F034) and Reserved Leaders of Heilongjiang Provincial Leading Talent Echelon (2021).

Data Availability Statement: https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 30 December 2020); http://rsidea.whu.edu.cn/resource_WHUHi_sharing.

htm (accessed on 28 December 2018); https://github.com/szubing/ED-DMM-UDA (accessed on 31 October 2019).

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

References

1. Paoletti, M.E.; Moreno-Álvarez, S.; Xue, Y.; Haut, J.M.; Plaza, A. AAtt-CNN: Automatic Attention-Based Convolutional Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5511118. [CrossRef]
2. Qi, W.; Huang, C.; Wang, Y.; Zhang, X.; Sun, W.; Zhang, L. Global–Local 3-D Convolutional Transformer Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5510820. [CrossRef]
3. Tu, B.; Ren, Q.; Li, Q.; He, W.; He, W. Hyperspectral Image Classification Using a Superpixel–Pixel–Subpixel Multilevel Network. *IEEE Trans. Geosci. Remote Sens.* **2023**, *72*, 5013616. [CrossRef]
4. Weber, C.; Aguejdad, R.; Briottet, X.; Avala, J.; Fabre, S.; Demuynck, J.; Zenou, E.; Deville, Y.; Karoui, M.S.; Benhalouche, F.Z.; et al. Hyperspectral Imagery for Environmental Urban Planning. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 1628–1631.
5. Yang, X.; Yu, Y. Estimating Soil Salinity Under Various Moisture Conditions: An Experimental Study. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2525–2533. [CrossRef]
6. Jiao, Q.; Zhang, B.; Liu, J.; Liu, L. A novel two-step method for winter wheat-leaf chlorophyll content estimation using a hyperspectral vegetation index. *Int. J. Remote Sens.* **2014**, *35*, 7363–7375. [CrossRef]
7. Liang, L.; Di, L.; Zhang, L.; Deng, M.; Qin, Z.; Zhao, S.; Lin, H. Estimation of crop LAI using hyperspectral vegetation indices and a hybrid inversion method. *Remote Sens. Environ.* **2015**, *165*, 123–134. [CrossRef]
8. Yue, J.; Fang, L.; Ghamisi, P.; Xie, W.; Li, J.; Chanussot, J.; Plaza, A. Optical Remote Sensing Image Understanding with Weak Supervision: Concepts, methods, and perspectives. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 250–269. [CrossRef]
9. Pal, M.; Foody, G.M. Feature Selection for Classification of Hyperspectral Data by SVM. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2297–2307. [CrossRef]
10. Cui, M.; Prasad, S. Class-Dependent Sparse Representation Classifier for Robust Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2683–2695. [CrossRef]
11. Zheng, J.; Feng, Y.; Bai, C.; Zhang, J. Hyperspectral Image Classification Using Mixed Convolutions and Covariance Pooling. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 522–534. [CrossRef]
12. Gu, Y.; Liu, T.; Jia, X.; Benediktsson, J.A.; Chanussot, J. Nonlinear Multiple Kernel Learning With Multiple-Structure-Element Extended Morphological Profiles for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3235–3247. [CrossRef]
13. Liu, T.; Gu, Y.; Chanussot, J.; Dalla Mura, M. Multimorphological Superpixel Model for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 6950–6963. [CrossRef]
14. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep Convolutional Neural Networks for Hyperspectral Image Classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
15. 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]
16. 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]
17. Fang, L.; Liu, G.; Li, S.; Ghamisi, P.; Benediktsson, J.A. Hyperspectral image classification with squeeze multibias network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1291–1301. [CrossRef]
18. Feng, J.; Chen, J.; Sun, Q.; Shang, R.; Cao, X.; Zhang, X.; Jiao, L. Convolutional Neural Network Based on Bandwise-Independent Convolution and Hard Thresholding for Hyperspectral Band Selection. *IEEE Trans. Cybern.* **2021**, *51*, 4414–4428. [CrossRef]
19. Kanthi, M.; Sarma, T.H.; Bindu, C.S. A 3d-Deep CNN Based Feature Extraction and Hyperspectral Image Classification. In Proceedings of the 2020 IEEE India Geoscience and Remote Sensing Symposium (InGARSS), Virtual, 1–4 December 2020; pp. 229–232.
20. Wang, L.; Zhu, T.; Kumar, N.; Li, Z.; Wu, C.; Zhang, P. Attentive-Adaptive Network for Hyperspectral Images Classification With Noisy Labels. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5505514. [CrossRef]
21. Li, Y.F.; Guo, L.Z.; Zhou, Z.H. Towards Safe Weakly Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 334–346. [CrossRef]
22. Kang, X.; Duan, P.; Xiang, X.; Li, S.; Benediktsson, J.A. Detection and correction of mislabeled training samples for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5673–5686. [CrossRef]
23. Tu, B.; Zhang, X.; Kang, X.; Wang, J.; Benediktsson, J.A. Spatial density peak clustering for hyperspectral image classification with noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5085–5097. [CrossRef]
24. Tu, B.; Zhou, C.; He, D.; Huang, S.; Plaza, A. Hyperspectral classification with noisy label detection via superpixel-to-pixel weighting distance. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4116–4131. [CrossRef]
25. Fang, Z.; Yang, Y.; Li, Z.; Li, W.; Chen, Y.; Ma, L.; Du, Q. Confident Learning-Based Domain Adaptation for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5527116. [CrossRef]

26. Algan, G.; Ulusoy, I. Image classification with deep learning in the presence of noisy labels: A survey. *Knowl.-Based Syst.* **2021**, *215*, 106771–106790. [CrossRef]
27. Roy, S.K.; Hong, D.; Kar, P.; Wu, X.; Liu, X.; Zhao, D. Lightweight heterogeneous kernel convolution for hyperspectral image classification with noisy labels. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 5509705. [CrossRef]
28. Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; Fergus, R. Training convolutional networks with noisy labels. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1406–1427.
29. Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; Li, F.-F. Mentornet:Learning data-driven curriculum for very deep neural networks on corrupted labels. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 2304–2313.
30. Xu, Y.; Li, Z.; Li, W.; Du, Q.; Liu, C.; Fang, Z.; Zhai, L. Dual-Channel Residual Network for Hyperspectral Image Classification with Noisy Labels. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5502511. [CrossRef]
31. Potghan, S.; Rajamenakshi, R.; Bhise, A. Multi-Layer Perceptron Based Lung Tumor Classification. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; pp. 499–502.
32. Chen, X.; Liang, C.; Huang, D.; Real, E.; Wang, K.; Liu, Y.; Pham, H.; Dong, X.; Luong, T.; Hsieh, C.; et al. Symbolic Discovery of Optimization Algorithms. *arXiv* **2023**, arXiv:2302.06675v4.
33. Li, Z.; Liu, M.; Chen, Y.; Xu, Y.; Li, W.; Du, Q. Deep Cross-Domain Few-Shot Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5501618. [CrossRef]
34. Zhong, Y.; Hu, X.; Luo, C.; Wang, X.; Zhao, J.; Zhang, L. WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H^2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF. *Remote Sens. Environ.* **2020**, *250*, 112012. [CrossRef]
35. Deng, B.; Jia, S.; Shi, D. Deep Metric Learning-Based Feature Embedding for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 1422–1435. [CrossRef]
36. Gao, H.; Yang, Y.; Li, C.; Gao, L.; Zhang, B. Multiscale Residual Network With Mixed Depthwise Convolution for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 3396–3408. [CrossRef]
37. 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]
38. Ying, L.; Haokui, Z.; Qiang, S. Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote Sens.* **2017**, *9*, 67.
39. 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]