

Hyper Spectral Image Data Classification Using Deep Learning

Gitanjali Pilankar, Dr. D. D. Doye

Submitted: 15/10/2024

Revised: 28/11/2024

Accepted: 08/12/2024

Abstract: This paper presents a novel framework for hyperspectral image classification, using deep learning techniques to achieve high classification accuracy. The proposed approach integrates convolutional neural networks (CNN) with improved clustering and feature fusion strategies, outperforming traditional methods. By integrating an optimized architecture and clustering strategy, the proposed method effectively addresses the challenges of integrating high-dimensional data and spectral-spatial features. Experiments performed on standard datasets show the superiority of the proposed model, achieving an overall accuracy (OA) of 99.10% for the Indian Pines dataset and 99.09% for the University of Pavia dataset, outperforming other state-of-the-art classifiers such as O-CNN, E-CNN and SVM. These results prove the effectiveness of the model in accurately capturing spatial-spectral features, making it suitable for hyperspectral data analysis tasks.

Keywords: capturing, presents, superiority, O-CNN, E-CNN, SVM

1. INTRODUCTION

Hyperspectral imagery (HSI) is widely used in remote sensing for its ability to capture hundreds of spectral channels in a single scene. However, accurate classification of HSI requires robust techniques to extract features from the image. This has been a challenging task due to factors such as the large amount of data, mixed pixels, and limited training samples. Over the years, various classification methods, including spectral domain classifiers like support vector machines (SVMs), random forest (RF), and multinomial logistic regression (MLR), have shown improvements in understanding HSI scenes.

Recent advancements in technology have introduced more promising approaches for HSI classification. Techniques such as morphological profiles (MPs), Markov random fields (MRFs), and sparsity signal-based methods have been introduced to leverage spatial and contextual properties for a better understanding of the image scenes. These methods aim to combine spectral and spatial information for classification. For instance, joint sparse models combine information from

neighboring pixels to enhance classification performance.

In the domain of computer vision, deep learning, particularly convolutional neural networks (CNNs), has gained significant interest for its outstanding performance in various tasks like face recognition, object detection, and video classification. CNNs can automatically learn feature representations from raw images through convolutional blocks, making them superior to traditional rule-based feature extraction methods. This feature extraction capability has led to the application of many CNN models for HSI classification. Researchers have employed CNNs to extract nonlinear and invariant deep features from raw HSI data and avoid overfitting through various strategies. Other works have combined CNNs with dimension reduction methods to extract spectral-spatial features, constructed hierarchical CNN structures for high-level deep feature learning, and designed specific CNN frameworks for learning both spectral and spatial features.

Although CNNs have shown success in feature extraction for HSI classification, there has been limited exploration of multiple feature learning for this purpose. Multiple feature learning aims to simultaneously learn multiple types of features to achieve more representative feature extraction for image

Research Scholar, SGGSIET
Supervisor, SGGSIET

processing tasks. In this paper, we propose an enhanced framework that combines CNN and multiple feature learning. We extract initial geometrical feature maps using four widely used attribute filters to capture spatial characteristics and local spatial correlations in the original image. These initial feature maps, along with the original image, are fed into the CNN with different inputs corresponding to different initial features. The CNN extracts representative features, which are then concatenated to form a joint feature map representing both spectral and contextual properties of HSI. This joint feature map is used in subsequent layers to determine the final labels of HSI pixels.

The proposed CNN consists of proper convolutional layers, a pooling layer for spatial invariance, a concatenating layer to exploit rich information, and a rectified linear unit (ReLU) function to accelerate convergence. As HSI often suffers from limited training samples, we designed a relatively shallow but effective network to avoid overfitting. The framework benefits from multiple inputs corresponding to various

image features and concurrently exploits spectral and spatial contextual information. Despite limited training samples, the proposed network remains robust and efficient.

The novelty of this work lies in the integration of multi-threading into a CNN-based framework, which allows the creation of common feature maps that capture rich spatial-spectral information. Using attribute profiles and a shallow but efficient CNN architecture, the proposed method addresses the challenges of limited training samples and achieves state-of-the-art classification performance on reference datasets.

The rest of the paper is organized as follows: Section 2 presents the detailed mechanism of the designed CNN and the proposed framework. In Section 3, we provide experimental results and discussions, investigating various factors influencing the results. Finally, Section 4 concludes the paper and offers remarks on the contributions of the proposed approach.

2. METHODS

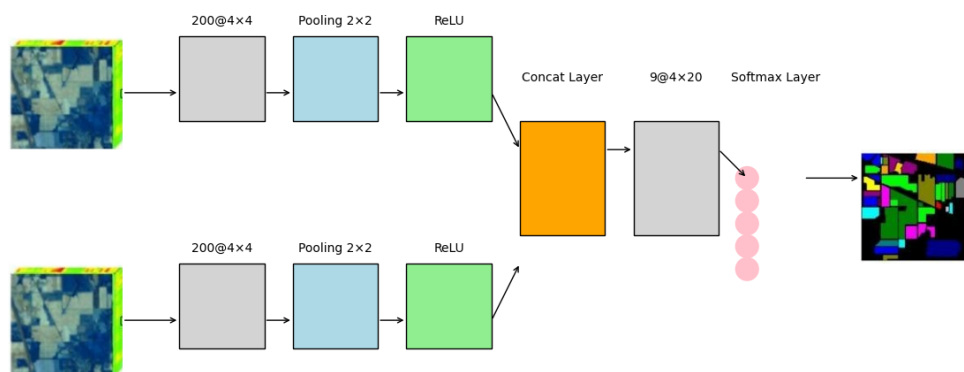


Fig 1. Describe the structure of the proposed framework. The initial step is to extract some HSI features, followed by some CNN blocks. For each set of features (total test set), individual CNN blocks are used to learn representative feature maps. These feature maps are then merged with a connection layer. The weights and biases of each block are refined by backpropagation in the network. The output of the network for each pixel is a vector containing class membership probabilities, with C units corresponding to the C classes defined in the hyperspectral dataset. The core principles of this proposed framework will be developed in more detail in the following sections.

2.1. Extraction of Feature Maps

Morphological profiles (MPs) are utilized to characterize spatial contextual information, representing structural variability in images

[27]. However, features obtained from a specific MP cannot model other geometrical characteristics. To address this limitation and simultaneously model various geometrical

characteristics for feature extraction in HSI classification, attribute profiles (APs) were introduced in the work of [28]. APs offer interesting properties in HSI processing and can be used to generate an extended AP (EAP). APs serve as a generalized form of MPs and are obtained by applying a criterion T to an image through morphological attribute filters (AFs) [28]. AFs are connected operators that process the image by merging connected components instead of individual pixels. When the operators are applied to regions, the attribute results are compared to a pre-defined reference value. Based on whether the criterion is met (i.e., if the attribute results are larger than the reference value), the region is either preserved or removed from the image. In the case of removal, the values in the eliminated region are set to the closest grayscale value of the adjacent region. Additionally, thinning or thickening may be applied based on whether the merged region has a lower or greater gray level, respectively. Afterward, an AP can be constructed by directly applying a sequence of thinning and thickening AFs to the image, using a set of specific criteria. This can be achieved by utilizing n morphological thickening (ϕT) and n thinning (ϕT) operators, resulting in the construction of an AP from the image f as follows:

$$AP(f) = \phi T(f), \phi T(\phi T(f)), \dots, \phi T(\phi T(\phi T(f))), f, \phi T(f), \phi T(\phi T(f)), \dots, \phi T(\phi T(\phi T(f))), \phi T(f) \quad \text{Eq. 1.}$$

Typically, the operators (thickening or thinning) in image processing are associated with common criteria like area, volume, diagonal box, and standard deviation. Depending on the chosen operators, the image can be transformed into an extensive or anti-extensive form. For the purpose of this paper, our focus is on measuring the effectiveness of multiple feature learning using the proposed CNN, rather than achieving absolute performance maximization. Therefore, we only extract attribute profiles (APs) based on four different criteria (i.e., area, standard deviation, moment of inertia, and length of the diagonal) as the diverse feature maps for the classification tasks. Each AP feature is named according to its corresponding criterion. More information on the various APs can be found in [27].

2.2. Proposed CNN

CNNs are designed to extract representative features from different types of data through multiple non-linear transformation architectures [29]. The features learned by a CNN are generally more reliable and effective compared to rule-based features. In this paper, we focus on HSI classification using directed acyclic graphs (DAG), where the layers are not limited to sequential chaining. In HSI classification, a neural network can map input HSI pixels to output pixel labels, and this function is composed of a sequence of simple blocks called layers. The basic layers in a CNN are as follows:

Mathematically, an individual neuron is computed by taking a vector of inputs x and applying an operator with a weight filter f and bias b :

$$a = \sigma(fx + b) \quad \text{Eq. 2}$$

Here, $\sigma(\bullet)$ represents a nonlinear function known as an activation function. In a convolutional layer, each neuron is associated with a specific spatial location (i, j) relative to the input image. The output $a_{i,j}$, which corresponds to the given input, can be defined as follows:

$$a_{i,j} = \sigma((F \otimes X)_{i,j} + b) \quad \text{Eq. 3}$$

In a network, the kernel function F utilizes learned weights, while X represents either the input or the layer. The convolution operator $\otimes$ is applied. Normally, networks include at least one layer of the activation function. Two commonly used activation functions are the sigmoid function and the ReLU function. Comparatively, the ReLU function is regarded as more effective than the sigmoid function in facilitating the training procedure's convergence [29]. The ReLU function is defined as follows:

$$\sigma(x) = \max(0, x) \quad \text{Eq. 4}$$

Another crucial layer type is pooling, which acts as a down-sampling function. The two most common pooling methods are max-pooling and mean-pooling. The pooling function divides the input feature map into rectangular regions and produces the maximum or mean value for each of these sub-regions. Consequently, this helps reduce computational complexity.

In the top layer, a softmax function is typically applied to obtain a probability distribution as an output, where each unit represents the probability of class membership. Following this principle, this paper introduces different features from the raw image into their respective CNN blocks, and the network is fine-tuned using backpropagation.

2.3. CNN Architecture

HSI typically contains numerous spectral bands, and a HSI classifier usually takes the entire image as input, which sets it apart from common classification problems. Recognizing the significance of spatial contextual information for HSI classification, we opt for a three-dimensional structure of the HSI pixel as input to the CNN model we built. Given a HSI cube $X \in \mathbb{R}^{M \times N \times L}$, where $M \times N$ represents the image size and L indicates the number of spectral channels, we use a $K \times K \times B$ format structure as input for a test pixel x_i (where i is the index of the test pixel). Here, $K \times K$ is a fixed neighborhood size, and B represents the dimension of the input features. For instance, in the original image cube, B equals the number of spectral channels L . In this study, after extracting T attribute profile features (e.g., area, standard deviation, length of diagonal, and moment of inertia), each attribute can be represented as $A_t \in \mathbb{R}^{M \times N \times B_t}$, where $t = 1, 2, \dots, T$. A_t denotes the t th attribute of X , and B_t indicates the number of spectral channels of A_t . For each pixel in A_t , a $K \times K \times B_t$ neighborhood region patch is selected as input for the corresponding model.

Each convolutional layer utilizes a four-dimensional convolution of $W \times W \times B \times F$, where $W \times W$ represents the kernel size of the convolutional layer, B is the dimension of the input variable, and F denotes the number of kernels in each convolutional layer. For example, for a $2 \times 2 \times 200 \times 50$ convolutional layer with an input size of $5 \times 5 \times 200$, the output in the DAG will be in a format of $4 \times 4 \times 50$, which will then be used as the input for the next layer.

The proposed network's three-dimensional input format results in a high dimensionality, potentially reaching several hundreds ($K \times K \times B$). This increased dimensionality can pose an overfitting risk during the training process. To address this issue, the proposed network incorporates the ReLU activation function.

The ReLU used in this study is a straightforward nonlinear function that assigns 0 or 1 based on whether the neuron's input is positive or negative. It has been demonstrated that ReLU can significantly enhance network performance in various scenarios [30].

For classification using the learned representative features, the proposed network employs the softmax operator in its top layer. Softmax is a probabilistic-based classification model that measures the correlation between an output value and a reference value through a probability score. It's worth noting that, in the CNN architecture, softmax can be applied across spectral channels for all spatial locations in a convolutional manner [31]. Given the three-dimensional input ($K \times K \times B$), the probability that the input belongs to class c is calculated as follows: [probability formula].

$$P(y = c) = \frac{e^{x_{mnc}}}{\sum_{b=1}^B \sum e^{x_{mnb}}} \quad \text{Eq. 5}$$

To achieve the necessary probability distribution using the softmax operator, the number of kernels in the last layer should be set equal to the number of classes defined in the HSI dataset. The entire training process of the network can be viewed as optimizing parameters to minimize a loss function, which measures the discrepancy between the network's outputs and the ground truth values for the training dataset. Let $y_i = 1, \dots, c, \dots, C$ represent the target ground truth value corresponding to the test pixel x_i , and $p(y_i)$ be the output class membership distribution with i as the index of the test pixel. The multi-class hinge loss employed in this study is expressed as follows:

$$L = \sum_{i=1}^N \sum_{c=1}^C \sum \max(0, 1 - p(y_i = c)) \quad \text{Eq. 6}$$

Ultimately, the prediction label is determined by selecting the argument that minimizes the value of the loss function:

$$\hat{y}_i = \underset{c}{\operatorname{argmin}} L_c \quad \text{Eq. 7}$$

3. RESULTS AND DISCUSSION

The proposed framework underwent testing using two benchmark HSI datasets. Section 3.1 introduces these datasets and provides details about their class information. In Section 3.2, the specific network architectures applied in this study and other relevant information for the experimental evaluation

are presented. The experimental results for all the classifiers are outlined in Section 3.3, while Section 3.4 focuses on additional experiments that influenced the classification outcomes.

In this research, both the original features and four attribute features derived from four attribute filters (area, moment of inertia, length

of diagonal, and standard deviation) are utilized as inputs to the proposed network. The parameters for each attribute profile criterion are set as defaults, as described in [28]. The attribute features extracted in this study have parameters set as described in [27, 31]. All the scripts were executed in Google Colab. Figure 2 depicts the architecture of the classifier.

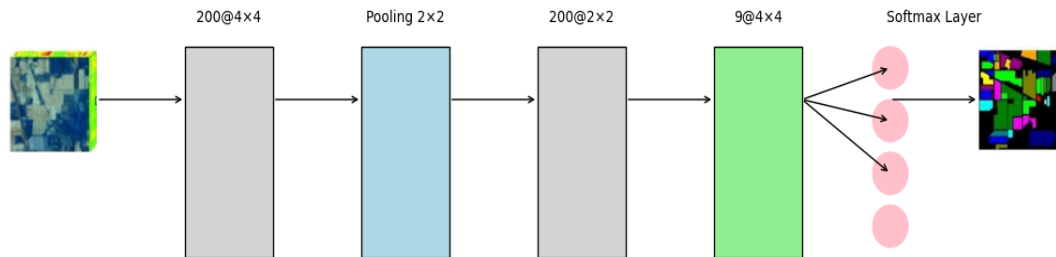


Fig 2. The architecture of classifier.

3.1 Description of the Data

In this paper, two benchmark datasets [32] are employed to assess the efficacy of the proposed framework.

1) The Indian Pines dataset was generated using the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, which captured imagery over a location in northwest Indiana, USA. The dataset consists of 220 spectral bands spanning wavelengths from 0.2 to 2.4 μm . Each band comprises 145 * 145 pixels with a spatial resolution of 20 meters. The dataset contains 16 labeled

classes, and during the experiments, 20 water absorption bands were excluded depicted in figure 3(a) and 3(b).

2) The University of Pavia, Italy, provided the Reflective Optics System Imaging Spectrometer (ROSIS) dataset. It was collected over the region in northern Italy and consists of 610 * 610 pixels with a spatial resolution of 1.3 meters per pixel. The dataset comprises nine labeled ground truths and includes 103 bands for the experiments after eliminating water absorption bands depicted in figure 3(c) and 3(d).

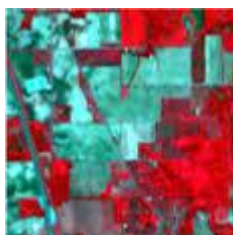


Fig.3.(a) False color composite of Indian Pines dataset



Fig.3.(b) groundtruth of Indian Pines dataset



Fig.3.(c) False color composite of Pavia dataset

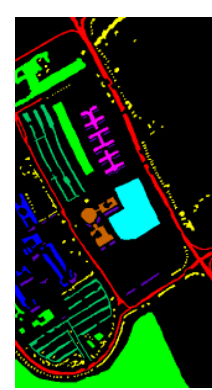


Fig.3.(d) False color composite of Pavia dataset

Each of the two datasets is divided into two subsets: a training set and a test set. The specifics regarding the number of subsets can

be found in Tables 1 and 2. During the training process of each Convolutional Neural Network (CNN) block, 90% of the training pixels are

utilized to learn the filter parameters for that block, while the remaining 10% form the validation set. The training set is responsible for adjusting the neural network's weights, while the validation set plays a crucial role in providing an unbiased evaluation of the

model's fit to the training data. This unbiased evaluation is especially important when tuning hyperparameters. The test set, on the other hand, is exclusively used to assess the performance of a fully-trained CNN model.

Table 1. Class Information for Indian Pines Data Set.

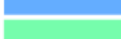
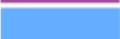
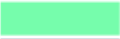
No.	Class	Color	Train	Test
1	Alfalfa		5	41
2	Corn-notill		143	1285
3	Corn-mintill		83	747
4	Corn		24	213
5	Grass-pasture		48	435
6	Grass-trees		73	657
7	Grass-pasture-mowed		3	25
8	Hay-windrowed		48	430
9	Oats		2	18
10	Soybean-notill		97	875
11	Soybean-mintill		245	2210
12	Soybean-clean		59	534
13	Wheat		20	185
14	Woods		126	1139
15	Buildings-Grass-Trees-Drives		39	347
16	Stone-Steel-Towers		9	84
Total			1024	9225

Table 2. Class Information for Pavia Data Set.

No.	Class	Color	Train	Test
1	Asphalt		199	6432
2	Meadows		559	18,090
3	Gravel		63	2036
4	Trees		92	2972
5	Metal Sheets		40	1305
6	Bare soil		151	4878
7	Bitumen		40	1290
8	Bricks		110	3572
9	Shadows		28	919
Total			1282	41,494

3.2. Experimental Setup and CNN Design

The CNN blocks were designed with a consistent architecture for different features. Each block consists of three convolutional layers, pooling layers, ReLU layers, and concatenating layers. Specific details about the network structure can be found in Tables 3 and

4. The input images are normalized to the range of $[-1, 1]$. In each convolutional layer, 200 kernels are empirically set. The input neighborhood size for each feature is $5 * 5$ for the Indian Pines dataset, and $7 * 7$ for the University of Pavia dataset.

For the CNN models, a learning rate of 0.01 is used, and the training process is run for 100 epochs for the Indian Pines and University of Pavia datasets. The batch size is set to 10. To ensure the robustness of the proposed framework, the results are quantitatively

validated using overall accuracy (OA), average accuracy (AA), and the Kappa coefficient (k) as performance metrics. Each reported result is an average of ten repeated experiments with randomly selected training samples.

Table 3. Network Architecture for Indian Pines Data Set.

Convolution: 4*20*200*16	11*11*32
Batch Normalization	11*11*32
Relu	11*11*32
Max Pooling : 2*2	5*5*32

Table 4. Network Structure for University of Pavia Data Set.

Convolution: 4*20*200*9	5*5*64
BN	5*5*64
ReLU	5*5*64
Max Pooling : 2*2	2*2*64

3.3. Result and discussion

3.3.1. Indian Pines Data set results

Evaluation Metrics

The evaluation process employs the confusion matrix as the primary quantitative measure, including overall accuracy (OA), average accuracy (AA), and statistical Kappa (κ) coefficients, to gauge the performance of the

proposed network against alternative methods. OA quantifies the ratio of correctly classified test samples to the total number of test samples, while AA represents the mean accuracy across all classes. The kappa coefficient assesses the level of agreement between the classification maps generated by the model under consideration and the provided ground truth.

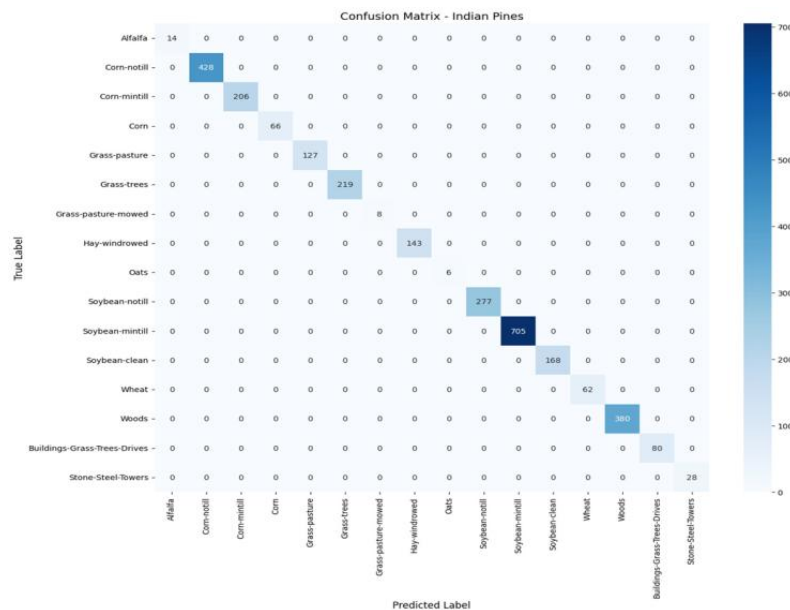


Fig.4 Confusion matrix for Indian Pines data

Table 5 displays the classification outcomes achieved by different classifiers for the Indian Pines dataset, and Figure 5 provides the

corresponding maps. It can be observed that all CNN-based models exhibit commendable

performance, with the proposed method showing notable improvements on this dataset.

Regarding O-CNN[41], the original image is used as input for the network. To validate the effectiveness of the proposed mechanism, spatial contextual features are extracted and combined as input for E-CNN. While E-CNN[41], achieves more accurate results than O-CNN, it falls short of surpassing the proposed method. The superior performance achieved by the proposed framework is likely attributed to the joint utilization of spatial-spectral information, resulting in reduced "salt-and-pepper" noise in the classification maps.

In comparison to O-CNN, the proposed method shows improvements of 2.3% in OA. Similar conclusions can be drawn when comparing the proposed method with E-CNN, particularly for classes with similar labels, as evident from Table 5. For example, the proposed method outperforms E-CNN by 5.18%, and as compared to SVM proposed method outperforms E-NN by 14.32% accuracy. Additionally, the proposed method excels at preserving discontinuities in the classification maps. Moreover, CNN methods do not require predefined parameters, unlike pixel-level extraction methods.

Table 5. Classification Results (%) of University of Indian Pine Data Set.

ClassNo.	O-CNN	E-CNN	SVM	Proposed
1	89.74 ± 0.01	94.63 ± 5.21	33.81 ± 0.05	100 ± 0.00
2	85.01 ± 0.12	89.90 ± 1.71	74.99 ± 0.03	96.76 ± 0.56
3	86.41 ± 0.26	89.97 ± 1.25	68.86 ± 0.02	99.57 ± 0.66
4	96.19 ± 0.02	97.65 ± 1.07	47.57 ± 0.04	99.15 ± 0.91
5	88.19 ± 0.01	97.06 ± 1.07	85.29 ± 0.03	99.86 ± 0.11
6	88.16 ± 0.02	99.33 ± 0.41	95.77 ± 0.03	99.51 ± 0.34
7		71.20 ± 60.00 ± 0.24		
	87.50 ± 0.04	10.24		100 ± 0.00
8	100 ± 0.00	99.86 ± 0.19	98.56 ± 0.01	99.90 ± 0.19
9		66.67 ± 30.00 ± 0.08		
	88.23 ± 0.16	11.11		93.34 ± 5.44
10	89.35 ± 0.01	92.78 ± 1.41	75.45 ± 0.02	99.43 ± 0.19
11	93.71 ± 0.03	95.70 ± 1.34	82.14 ± 0.01	99.38 ± 0.20
12	93.63 ± 0.23	80.56 ± 3.29	61.31 ± 0.01	98.58 ± 0.39
13	100 ± 0.00	99.68 ± 0.43	95.14 ± 0.02	99.89 ± 0.22
14	97.99 ± 0.04	98.53 ± 1.03	94.19 ± 0.02	99.84 ± 0.13
15	88.92 ± 0.01	90.72 ± 2.67	53.91 ± 0.04	100 ± 0.00
16	95.29 ± 0.01	91.90 ± 5.80	80.95 ± 0.05	97.14 ± 0.95
OA	91.62 ± 0.01	93.92 ± 0.91	79.60 ± 0.01	99.10 ± 0.06
AA	91.78 ± 0.01	91.01 ± 0.93	71.12 ± 0.01	98.90 ± 0.30
<i>Kappa</i>	90.45 ± 0.01	93.06 ± 1.04	76.66 ± 0.01	98.98 ± 0.0

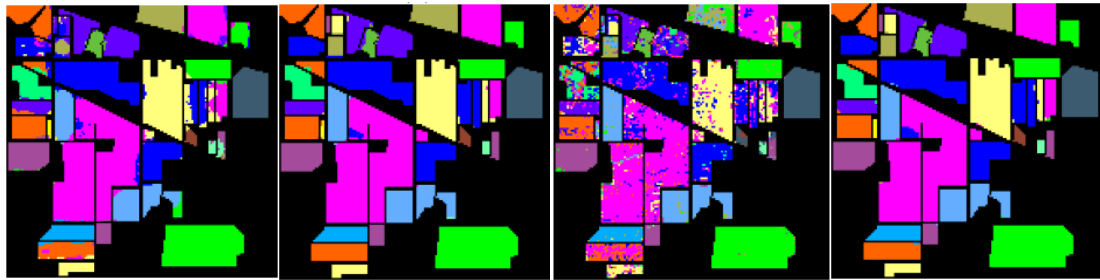


Fig. 5 (a) Classification maps of the Indian Pines dataset O-CNN(OA=91.62)

Fig. 5 (b) Classification maps of the Indian Pines dataset O-ENN(OA=93.92)

Fig. 5 (c) Classification maps of the Indian Pines dataset SVM (OA=79.60)

Fig. 5 (d) Classification maps of the Indian Pines dataset Proposed (OA=99.10)

3.3.2. Pavia Data set classification

Evaluation Metrics

The evaluation process employs the confusion matrix as the primary quantitative measure, including overall accuracy (OA), average accuracy (AA), and statistical Kappa (κ) coefficients, to gauge the performance of the proposed network against alternative methods.

OA quantifies the ratio of correctly classified test samples to the total number of test samples, while AA represents the mean accuracy across all classes. The kappa coefficient assesses the level of agreement between the classification maps generated by the model under consideration and the provided ground truth.

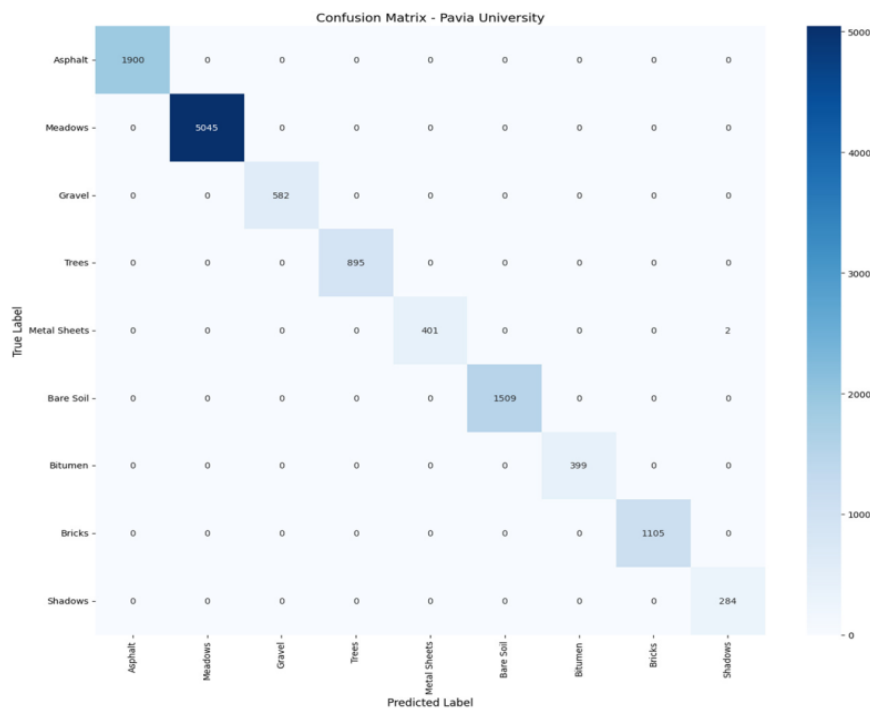


Fig.6 Confusion matrix for University of Pavia data

Table 6 presents the class-specific classification accuracies for the University of Pavia image, while Figure 7 displays the representative classification maps. The results clearly demonstrate the superiority of the

proposed method over other algorithms in terms of OA, AA, and Kappa. When evaluated using the University of Pavia dataset, the proposed method achieves remarkable accuracy improvements. Inspecting the

classification maps, it is evident that O-CNN and E-CNN exhibit more scattered noisy points compared to the proposed method. In contrast, the proposed method effectively eliminates these noisy artifacts and delivers smoother classification results while preserving clear boundaries without blurring depicted in figure 7.

In comparison to E-CNN, the proposed method shows improvements of 1.12% in OA. Similar conclusions can be drawn when

comparing the proposed method with E-CNN, particularly for classes with similar labels, as evident from Table 6. For example, the proposed method outperforms SVM by 5.15%, and as compared to O-CNN proposed method outperforms by 1% accuracy. Additionally, the proposed method excels at preserving discontinuities in the classification maps. Moreover, CNN methods do not require predefined parameters, unlike pixel-level extraction methods.

Table 6. Classification Results (%) of University of Pavia Data Set.

ClassNo.	O-CNN	E-CNN	SVM	Proposed
1	98.75 $\pm$ 0.01	99.36 $\pm$ 0.47	94.65 $\pm$ 0.01	99.29 $\pm$ 0.56
2	99.35 $\pm$ 0.01	99.96 $\pm$ 0.03	98.12 $\pm$ 0.01	99.97 $\pm$ 0.02
3	91.83 $\pm$ 0.03	97.89 $\pm$ 1.05	76.84 $\pm$ 0.04	91.68 $\pm$ 2.31
4	93.00 $\pm$ 0.02	90.51 $\pm$ 3.52	92.91 $\pm$ 0.03	98.48 $\pm$ 0.64
5	98.57 $\pm$ 0.01	96.88 $\pm$ 3.45	99.30 $\pm$ 0.01	100 $\pm$ 0.00
6	99.68 $\pm$ 0.01	99.40 $\pm$ 0.45	87.84 $\pm$ 0.02	99.79 $\pm$ 0.27
7	99.70 $\pm$ 0.01	99.18 $\pm$ 0.76	85.92 $\pm$ 0.02	98.60 $\pm$ 1.08
8	96.56 $\pm$ 0.02	97.99 $\pm$ 1.72	89.92 $\pm$ 0.01	97.87 $\pm$ 0.63
9	93.65 $\pm$ 0.03	78.24 $\pm$ 12.68	99.76 $\pm$ 0.01	99.41 $\pm$ 0.47
OA	98.09 $\pm$ 0.01	98.25 $\pm$ 0.42	93.94 $\pm$ 0.01	99.09 $\pm$ 0.03
AA	96.79 $\pm$ 0.01	95.49 $\pm$ 1.44	91.69 $\pm$ 0.01	98.34 $\pm$ 0.18
Kappa	97.47 $\pm$ 0.01	97.68 $\pm$ 0.57	91.93 $\pm$ 0.01	98.80 $\pm$ 0.05

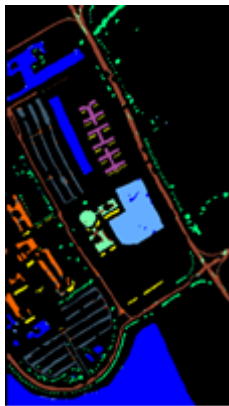


Fig. 7 (a) Classification maps of the Indian Pines dataset O-CNN(OA=98.09)



Fig. 7 (b) Classification maps of the Indian Pines dataset O-ENN(OA=98.25)

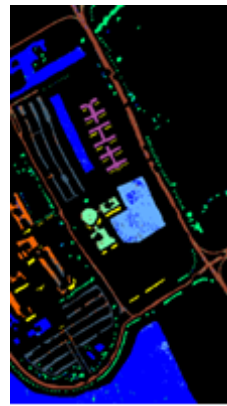


Fig. 7 (c) Classification maps of the Indian Pines dataset SVM (OA=93.94)

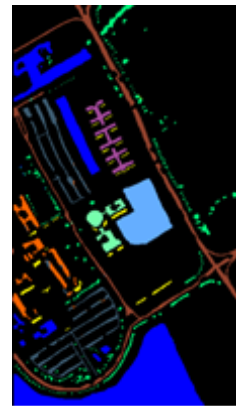


Fig.7 (d) Classification maps of the Indian Pines dataset Proposed (OA=99.09)

3.3.3 The Influence of the Number of Training Iterations

The number of training epochs plays a crucial role in CNN-based methods. Figure 8 illustrates how the training error changes with

varying epochs across all three datasets. During the network training process, backpropagation is used to minimize the training objective= $-\sum_{i=1}^{N_t} \sum \log(p_{ic})$, which

is computed as the sum of logarithms of prediction probabilities for each class.

The "error" term is also calculated during training by $error = \sum_{i=1}^{N_t} \sum P_{ic}(\text{argmax}_i \cong c)$, where, training samples denoted by N_t and

P_{ic} representing the sum of prediction probabilities for pixels x_i belonging to their respective c_{th} classes. This assessment proves to be helpful in evaluating the model. ReLU's presence can expedite network convergence and enhance overall training efficiency [29].

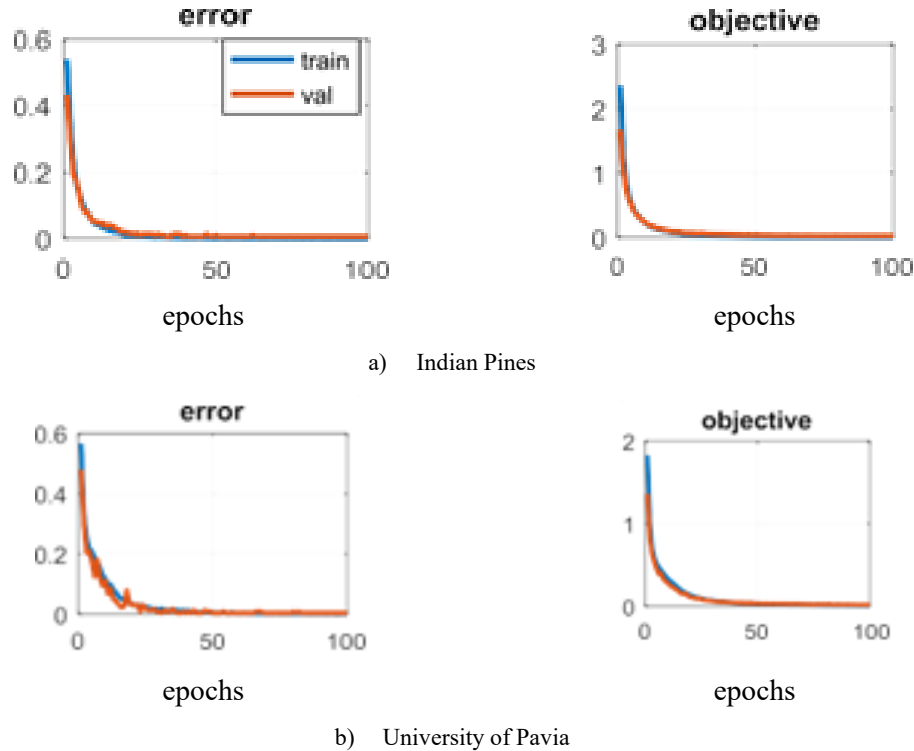


Fig. 8 The training error of the proposed framework was evaluated on two different datasets.

3.3.4. The influence of the number of training samples

The number of training samples plays a crucial role in training a CNN. It is well-known that CNNs require an abundance of training samples to effectively extract features. However, in the case of hyperspectral imagery (HSI), obtaining a large number of training samples is not common. Therefore, it is essential to develop a network that is robust and efficient for classification tasks in such scenarios.

In this paper, the impact of the number of training samples on the accuracies of three datasets is investigated. For the Indian Pines scene, training pixels are randomly selected from 5% to 50% of the samples, with the remaining pixels used for testing. For the University of Pavia and Salinas images, 50 to

500 pixels per class are randomly chosen as training samples, while the rest are used for testing.

Figure 9 displays the overall accuracy (OA) of various methods with different numbers of training samples. It is evident that all methods show improved performance as the number of training samples increases, particularly for the Indian Pines dataset, where the proposed method outperforms the others. Notably, the proposed method achieves an accuracy higher than 95% with less than 10% of the training samples. As the number of training samples further increases, the accuracies tend to stabilize for all three methods.

For the University of Pavia dataset, the classification accuracies of the CNN-based methods reach approximately 100% with more training samples, especially for the proposed method, which achieves an accuracy of over 96% with 50 samples per class.

Throughout all two datasets, the CNN-based classifiers demonstrate sensitivity to the number of training samples, with accuracy increasing as more training samples are used. Moreover, these CNN-based approaches show

competitive performance even with a large number of training samples, with the proposed method displaying enhanced robustness across different sample sizes depicted in figure 7.

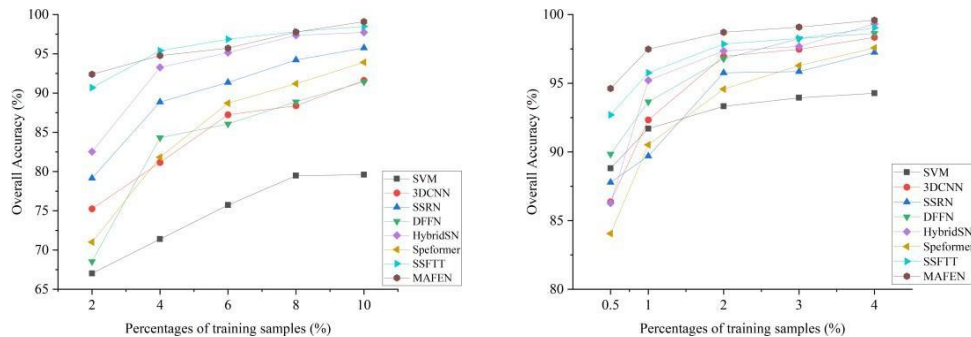


Fig.9 Model performance across various proportions of training samples.(a)Indian Pines; (b)Pavia University

4. Conclusion

This study introduced a novel convolutional neural network (CNN) framework designed for hyperspectral image (HSI) classification. The proposed method combines spectral and spatial features to create a highly efficient and powerful classification model. By using multiple CNN blocks and a connection layer, the architecture efficiently captures various spatial-spectral information, thereby improving classification performance. Experimental results on standard datasets, including the Indian Pine and the University of Pavia, emphasize the superiority of the proposed method over traditional and existing CNN-based approaches. The model achieved an overall accuracy (OA) of 99.10% and 99.09% on the Indian Pines and University of Pavia datasets, respectively. These results outperformed alternative methods such as O-CNN, E-CNN and SVM, especially in preserving spatial discontinuities and minimizing noise artifacts in classification maps. For example, the proposed framework was shown to improve by 5.18% over E-CNN and 14.0% over SVM. 32% improvement over SVM for specific classes. The inclusion of ReLU activation and pooling

layers, along with strategic use of multiple input feature maps, mitigates overfitting and accelerates convergence, even with limited training samples. In particular, the model maintained strong performance with fewer training samples, achieving more than 95% accuracy on Indian pine trees with less than 10% training samples. This adaptation for small The datasets highlight the practicality of the proposed approach in real-world scenarios. In summary, the proposed CNN-based framework shows significant advances in HSI classification, providing a shallow but effective network that integrates multiple feature learning strategies. The results highlight the potential of leveraging spatial information and deep learning to improve classification accuracy, paving the way for future advances in hyperspectral data analysis.

Author Contributions

We present a novel framework for hyperspectral image (HSI) classification that integrates convolutional neural networks (CNNs) with a multi-feature learning approach. Their primary contribution lies in designing a unique CNN architecture capable of efficiently capturing both spatial and spectral features through multiple individual CNN blocks, each extracting distinct features from the HSI data. By concatenating these

feature maps into a joint feature map, the model can leverage a combination of spatial and spectral information, leading to improved classification accuracy. The framework also addresses the issue of limited training samples in HSI by using multiple input feature maps and ReLU activation, which helps reduce overfitting and accelerate convergence. Additionally, the authors identified an optimal three-layer CNN structure with a specific neighborhood and pooling size, balancing efficiency and complexity. Extensive experiments on three benchmark datasets confirm the framework's effectiveness, showcasing significant performance improvements over traditional classifiers and CNN-based single feature methods.

Conflicts of Interest

"The authors declare no conflicts of interest regarding the publication of this paper."

References

- [1] Melgani, Farid, and Lorenzo Bruzzone. "Classification of hyperspectral remote sensing images with support vector machines." *IEEE Transactions on geoscience and remote sensing* 42, no. 8 (2004): 1778-1790.
- [2] Pal, Mahesh, and Giles M. Foody. "Feature selection for classification of hyperspectral data by SVM." *IEEE Transactions on Geoscience and Remote Sensing* 48, no. 5 (2010): 2297-2307.
- [3] Ham, Jisoo, Yangchi Chen, Melba M. Crawford, and Joydeep Ghosh. "Investigation of the random forest framework for classification of hyperspectral data." *IEEE Transactions on Geoscience and Remote Sensing* 43, no. 3 (2005): 492-501.
- [4] Li, Jun, José M. Bioucas-Dias, and Antonio Plaza. "Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression." *IEEE Geoscience and Remote Sensing Letters* 10, no. 2 (2012): 318-322.
- [5] Benediktsson, JónAtli, JónAevarPalmason, and Johannes R. Sveinsson. "Classification of hyperspectral data from urban areas based on extended morphological profiles." *IEEE Transactions on Geoscience and Remote Sensing* 43, no. 3 (2005): 480-491.
- [6] Fauvel, Mathieu, JónAtliBenediktsson, Jocelyn Chanussot, and Johannes R. Sveinsson. "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles." *IEEE Transactions on Geoscience and Remote Sensing* 46, no. 11 (2008): 3804-3814.
- [7] Li, Jun, José M. Bioucas-Dias, and Antonio Plaza. "Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields." *IEEE Transactions on Geoscience and Remote Sensing* 50, no. 3 (2011): 809-823.
- [8] Zhang, Bing, Shanshan Li, XiupingJia, Lianru Gao, and Man Peng. "Adaptive Markov random field approach for classification of hyperspectral imagery." *IEEE Geoscience and Remote Sensing Letters* 8, no. 5 (2011): 973-977.
- [9] Chen, Yi, Nasser M. Nasrabadi, and Trac D. Tran. "Hyperspectral image classification using dictionary-based sparse representation." *IEEE transactions on geoscience and remote sensing* 49, no. 10 (2011): 3973-3985.
- [10] Parkhi, Omkar, Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition." In *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015.
- [11] Lawrence, Steve, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back. "Face recognition: A convolutional neural-network approach." *IEEE transactions on neural networks* 8, no. 1 (1997): 98-113.
- [12] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).
- [13] Karpathy, Andrej, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725-1732. 2014.
- [14] Chen, Yushi, Hanlu Jiang, Chunyang Li, XiupingJia, and PedramGhamisi. "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks." *IEEE transactions on geoscience and remote sensing* 54, no. 10 (2016): 6232-6251.

- [15] Zhao, Wenzhi, and Shihong Du. "Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach." *IEEE Transactions on Geoscience and Remote Sensing* 54, no. 8 (2016): 4544-4554.
- [16] Makantasis, Konstantinos, Konstantinos Karantzalos, Anastasios Doulamis, and Nikolaos Doulamis. "Deep supervised learning for hyperspectral data classification through convolutional neural networks." In *2015 IEEE international geoscience and remote sensing symposium (IGARSS)*, pp. 4959-4962. IEEE, 2015.
- [17] Yue, Jun, Wenzhi Zhao, Shanjun Mao, and Hui Liu. "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks." *Remote Sensing Letters* 6, no. 6 (2015): 468-477.
- [18] Kürüm, Ulas, Peter R. Wiecha, Rebecca French, and Otto L. Muskens. "Deep learning enabled real time speckle recognition and hyperspectral imaging using a multimode fiber array." *Optics express* 27, no. 15 (2019): 20965-20979.
- [19] Zhao, Wenzhi, Zhou Guo, Jun Yue, Xiuyuan Zhang, and Liqun Luo. "On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery." *International Journal of Remote Sensing* 36, no. 13 (2015): 3368-3379.
- [20] Luus, Francois PS, Brian P. Salmon, Frans Van den Bergh, and BodhaswarTikanathJugpershadMaharaj. "Multiview deep learning for land-use classification." *IEEE Geoscience and Remote Sensing Letters* 12, no. 12 (2015): 2448-2452.
- [21] Zhang, Lu, Zhenwei Shi, and Jun Wu. "A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8, no. 10 (2015): 4895-4909.
- [22] Zhang, Fan, Bo Du, and Liangpei Zhang. "Scene classification via a gradient boosting random convolutional network framework." *IEEE Transactions on Geoscience and Remote Sensing* 54, no. 3 (2015): 1793-1802.
- [23] Wang, Jun, Jingwei Song, Mingquan Chen, and Zhi Yang. "Road network extraction: A neural-dynamic framework based on deep learning and a finite state machine." *International Journal of Remote Sensing* 36, no. 12 (2015): 3144-3169.
- [24] Ding, Changxing, Chang Xu, and Dacheng Tao. "Multi-task pose-invariant face recognition." *IEEE Transactions on Image Processing* 24, no. 3 (2015): 980-993.
- [25] Zhu, Chao, and Yuxin Peng. "A boosted multi-task model for pedestrian detection with occlusion handling." *IEEE transactions on image processing* 24, no. 12 (2015): 5619-5629.
- [26] Liu, Wu, Tao Mei, Yongdong Zhang, Cherry Che, and Jiebo Luo. "Multi-task deep visual-semantic embedding for video thumbnail selection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3707-3715. 2015.
- [27] Dalla Mura, Mauro, Alberto Villa, Jon AtliBenediktsson, Jocelyn Chanussot, and Lorenzo Bruzzone. "Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis." *IEEE Geoscience and Remote Sensing Letters* 8, no. 3 (2010): 542-546.
- [28] Dalla Mura, Mauro, JónAtliBenediktsson, BjörnWaske, and Lorenzo Bruzzone. "Morphological attribute profiles for the analysis of very high resolution images." *IEEE Transactions on Geoscience and Remote Sensing* 48, no. 10 (2010): 3747-3762.
- [29] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).
- [30] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807-814. 2010.
- [31] Vedaldi, Andrea, and Karel Lenc. "Matconvnet: Convolutional neural networks for matlab." In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 689-692. 2015.
- [32] Gao, Qishuo, Samsung Lim, and XiupingJia. "Hyperspectral image classification using convolutional neural networks and multiple feature learning." *Remote Sensing* 10, no. 2 (2018): 299.
- [33] Zhang, Xiao, Liangyun Liu, Xidong Chen, Yuan Gao, and Mihang Jiang. "Automatically monitoring impervious

- surfaces using spectral generalization and time series Landsat imagery from 1985 to 2020 in the Yangtze River Delta." *Journal of Remote Sensing* 2021 (2021).
- [34] Yang, Xiguang, and Ying Yu. "Estimating soil salinity under various moisture conditions: An experimental study." *IEEE Transactions on Geoscience and Remote Sensing* 55, no. 5 (2017): 2525-2533.
- [35] Avtar, Ram, NetranandaSahu, Ashwani Kumar Aggarwal, Shamik Chakraborty, Ali Kharrazi, Ali P. Yunus, Jie Dou, and TonniAgustionoKurniawan. "Exploring renewable energy resources using remote sensing and GIS—A review." *Resources* 8, no. 3 (2019): 149.
- [36] Li, Jun, José M. Bioucas-Dias, and Antonio Plaza. "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields." *IEEE Transactions on Geoscience and Remote Sensing* 50, no. 3 (2011): 809-823.
- [37] Wenju, Li, Chu Wanghui, Cui Liu, and Zhang Gan. "A graph attention feature pyramid network for 3D object detection in point clouds." In *2022 7th International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS)*, vol. 7, pp. 94-98. IEEE, 2022.
- [38] Mei, Shaohui, Xingang Li, Xiao Liu, HuiminCai, and Qian Du. "Hyperspectral image classification using attention-based bidirectional long short-term memory network." *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021): 1-12.
- [39] Yu, Haoyang, Hao Zhang, Yao Liu, Ke Zheng, Zhen Xu, and Chenchao Xiao. "Dual-channel convolution network with image-based global learning framework for hyperspectral image classification." *IEEE Geoscience and Remote Sensing Letters* 19 (2021): 1-5.
- [40] Song, Weiwei, Shutao Li, Leyuan Fang, and Ting Lu. "Hyperspectral image classification with deep feature fusion network." *IEEE Transactions on Geoscience and Remote Sensing* 56, no. 6 (2018): 3173-3184.
- [41] Gao, Q., Lim, S., &Jia, X. (2018). Hyperspectral image classification using convolutional neural networks and multiple feature learning. *Remote Sensing*, 10(2), 299.