Dynamic Data Augmentation Method for Hyperspectral Image Classification Based on Siamese Structure

Hongmin Gao, Junpeng Zhang, Xueying Cao, Zhonghao Chen, Yiyan Zhang, Chenming Li

***Abstract*—At present, deep learning classification researches of hyperspectral usually focus on optimizing the classification model. In essence, most of them did not take special measures for the characteristics of the small sample and imbalanced category distribution of hyperspectral itself. Aiming at the problems of small samples and imbalanced category distribution, we propose a dynamic data selection algorithm. For one thing, this algorithm can dynamically select the samples that need data augmentation most. For another, it can be nested in Stochastic gradient descent (SGD) and can be easily implemented. Furthermore, there will be differences between the original sample and the transformed sample because of data augmentation transformation, which obstructs trained models' performance. Aiming at the difference between the augmented sample and the original sample, we define the similarity score and introduce the Siamese training structure to obtain the similarity score by which we reduce the difference through the SGD algorithm. Experiments show that the method proposed in this paper improves the classification results of the backbone training model when using data augmentation for training.**

***Index Terms*—Hyperspectral (HSI) classification, data augmentation (DA), Siamese structure, convolutional neural network (CNN).**

I. Introduction

HYPERSPECTRAL images (HSIs) have hundreds of almost continuous spectral bands, providing a wealth of spectral information. Since the spectra reflected by different ground objects have different characteristics, and HSIs happen to have much spectral information, researchers can use their spectral characteristics to classify ground objects. HSI classification is widely used in agricultural statistics, mineral reconnaissance, military surveillance, and other industries.

In the literature [14]-[22], many algorithms have been proposed for HSI preprocessing and classification, using supervised, semi-supervised and unsupervised methods. In the literature [32], augmented linear mixing model (LMM) is proposed to address spectral variability in inverse problems of hyperspectral unmixing. In the literature [33], nonconvex modeling has proven to be a feasible solution that reduces the gap between challenging HS vision tasks and currently advanced intelligent data processing models. In the literature [34], a new hyperspectral dimensionality reduction method called iterative multitask regression (IMR) is proposed to consider the labeled and unlabeled data.

In recent years, deep learning methods have been widely used in hyperspectral image classification. In the literature [1], a stacked autoencoder (SAE) is used to extract the spatial spectra features of HSI, and then the extracted features are input into a logistic regression to obtain the classification results. In the literature [2], a deep belief network (DBN) is used to extract the spectral features of a single pixel. In the literature [3], a convolutional neural network (CNN) is used to extract the spatial-spectral features of HSI. In the literature [4], the restricted Boltzmann machine (RBM) is used for HSI classification with a spatial-spectral combination. In the literature [5], the idea of recurrent neural network (RNN) is combined with CNN models to produce the R-CNN series models.

CNN is widely used in HSI feature extraction tasks among these deep learning structures due to local perception, weight sharing, and other features. In literature [6], two-dimensional CNN is used as the basic module. Combined with multi-task learning strategies, two data sets are input for one model training, so that the network itself has more diverse feature recognition capabilities. Reference [7] introduces the attention mechanism, uses three-dimensional CNN to extract the spatial-spectral features of the data, and then performs feature fusion. Literature [8] uses multi-scale features brought by convolution kernels of different sizes and dilated convolutions to classify HSI. All these above show the commonality and importance of CNN in the feature extraction process.

However, in many current HSI classification algorithms using deep learning, researches are often limited to the classifiers (deep learning models), ignoring the data distribution characteristics of HSI itself. First of all, HSI

This paragraph of the first footnote will contain the date on which you submitted your paper for review. This work was supported in part by the National Natural Science Foundation of China under Grant 62071168, in part by the National Key R&D Program of China under Grant 2018YFC1508106, in part by the Fundamental Research Funds for the Central Universities of China under Grant B200202183, and in part by the China Postdoctoral Science Foundation under Grant 2021M690885. *(Corresponding author: Chenming Li.)*

The authors are with the College of Computer and Information, Hohai University, Nanjing 211100, China (e-mail: gaohongmin@hhu.edu.cn, 201307020015@hhu.edu.cn, shary@hhu.edu.cn, chenzhonghao@hhu.edu.cn, zhangyiyan@hhu.edu.cn, lcm@hhu.edu.cn).

classification is different from general color picture classification. Color pictures are often easier to obtain and label, while the pixel of HSI data cannot be intuitively determined and is not easy to obtain. At the same time, in practical applications, the process of manually labeling training samples is often cumbersome and costly. As a result, only a limited number of training samples can be obtained, so it is necessary to perform proper data augmentation (DA) during classification.

Secondly, due to the randomness and unpredictability of the ground truth distribution, HSI data cannot guarantee the uniform distribution of the number in each category. For example, in the Indian pines (IP) data set, there are only 20 oats categories with the smallest samples, while there are 2455 soy mint mixed categories with the largest samples. In the Pavia University (PU) data set, the number of grass categories with the largest samples is more than 20 times the number of shadows with the smallest samples. This shows that the imbalanced distribution of HSI samples is a common phenomenon. For ordinary color image data sets, such as the CIFAR100 data set, the number of samples in each category is artificially set to be equal. The imbalance of HSI data will make the deep learning models fit each category differently, making it difficult for the models to identify categories with small numbers. Therefore, DA needs to be adjusted according to the actual situation of the target samples.

Furthermore, since DA uses certain transformations on the original sample to generate additional samples, and deep learning models are often sensitive to small changes, DA will inevitably create a certain "distance" between the original sample and the augmented sample, which will interfere with the training result. The transformed samples can indeed provide information with diversity. However, the transformed samples usually deviate too much from the raw samples, which is redundant. Under this circumstance, the training result will be disturbed. This paper tries to achieve a dynamic balance between the acquisition of diverse information and the deviation of transformed samples from raw samples. This point is often overlooked in many HSI classification algorithms.

In summary, the HSI classification algorithm's improvement should consider the classification model and take specific optimization measures based on the characteristics of the HSI itself.

In response to the above problems, this paper proposes a Siamese Structure dynamic data augmentation (SSDDA) method for HSI deep learning classification. This method considers the characteristics of small samples and imbalanced categories of HSI data. The specific innovations are summarized as follows:

1) Given the uneven distribution of HSI sample categories, this paper designs a dynamic sample selection algorithm, enabling the model to dynamically select the original samples that need to be augmented in each batch during training, and balances the model’s response to different categories, thereby improving the fitting degree of the model for some categories with a small number. The comprehensive classification results of the deep learning model are thus improved.

2) Aiming at the problem that the "distance" between the DA sample and the original sample will interfere with the model, this paper defines the similarity score to measure the degree of similarity between the DA sample and the original sample. At the same time, the Siamese structure is used to get the similarity score. Combined with Stochastic gradient descent (SGD), after training iterations, the difference between the DA sample and the original sample is reduced, and the interference of the DA sample on the model is weakened, making it easier for the model to fit the original sample.

3) To be able to perform DA more flexibly, this paper uses convolution operation to perform DA on the original sample. With the Siamese structure, the parameters of the convolution kernel can be dynamically updated, thereby generating DA samples that are more similar to the original sample, which further reduces the interference caused by difference.

The rest of this paper is summarized as follows. First, section II describes the proposed method. Then, section III analyzes the experimental results. Finally, the conclusion is drawn in Section IV.

II. Proposed Method

In this part, first, we define the similarity score to measure the difference between the augmented samples and the original samples. Then the concept of Siamese Structure is introduced. Based on the above, this paper proposes a dynamic data selection algorithm for data augmentation and a Siamese structure data augmentation method to reduce similarity scores. Part *C, D, E*, and SGD make up the whole process. The overall process is shown in Fig. 2.

A. Similarity Score

DA transforms the original data block to augment the number of samples, and the transformation inevitably leads to difference between the new samples and the original samples. The similarity score is used to measure the similarity between the original sample and corresponding augmented sample that belong to the same category.

Denote the original sample data block as $x \in R^{M \times N \times C}$, where M , N and C represent the height, width, and number of channels of the data block respectively. The new sample is recorded as $x' \in R^{M \times N \times C}$. Let the number of sample categories be P . Assuming that the classification model is a CNN, and the last layer is the soft-max layer corresponding to the number of sample categories, the classification model can be defined as a function: $y = f(x) \in R^{P \times 1}$, whose output is the soft-max result of the model. The final category can be obtained after an argmax layer. Let y be the soft-max result of the original sample, y' be the soft-max result of the new sample. Concerning category cross entropy, the similarity score of x and x' is defined as:

$$S_0 = -\sum_{i=1}^{P} y^{(i)} \log y'^{(i)} \qquad (1)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2021.3102610, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

The larger S_0 is, the greater the difference between y and $y^{'}$ is. For the soft-max results of a set of m original samples and their one-to-one corresponding new samples $Y, Y' \in R^{P \times m}$, the average similarity score of the two sets of data is defined as:

$$S_{average} = -\frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{P} Y_j^{(i)} \log Y_j^{'(i)} \qquad (2)$$

B. Siamese Structure

Siamese can be referred to as conjoined twins, which means that two humans live together, sharing the same body except for the head and lower body. Analogously, the Siamese network structure is realized by sharing weights between two neural networks.

As shown in Fig. 1, the overall network structure has two inputs. Neural Network 1 and Neural Network 2 share weight parameters. Due to different inputs, the outputs of the two sub-networks are also different, and loss is used to indicate the difference between the two results. Therefore, the Siamese network structure can be used to measure the similarity between two different image samples.

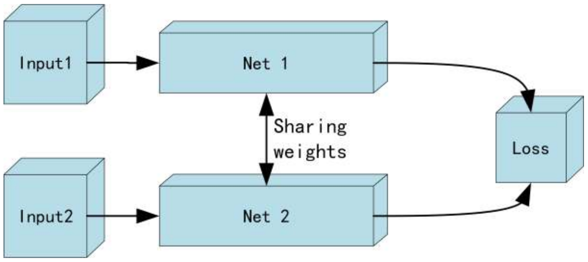

Fig 1. Illustration of the Siamese Structure. It comprises two inputs and two sub-networks sharing weights. The outputs of the sub-networks constitute the loss function.

C. Dynamic Augmentation Data Selection Method

When using SGD to train a model, the original data set is first divided into a training set and a testing set. Then, during an epoch of training, the program randomly selects a batch of data from the training set. However, due to the uneven distribution of the sample categories, the category distribution in this batch of training data is also uneven. Therefore, DA needs to be performed on the sample with the least number of occurrences. This paper designs an algorithm to dynamically select the samples that need DA most in a batch.

Define the sample in a batch as X_{batch}, its corresponding label is Y_{batch}. The number of planned DA samples is n, which satisfies $1 \le n \le m$. X_{DA} and Y_{DA} are empty lists which store chosen samples and their labels respectively. X_{new} and Y_{new} represents the lists merged by the augmented samples and the original samples. cnt is used to count the number of each category, and its initial subscript ($index$) is set to be 0. The first dimension of cnt refers to each category and its corresponding count. The second dimension of cnt refers to the total number of categories.

In summary, the dynamic augmentation data selection method gets the raw training batch as inputs. During the selection process, the samples in this batch which most need augmentation are selected and augmented. Finally, the outputs are the augmented samples and the merged new training batch.

In the standard HSI small sample classification method, when DA is used, DA samples are often artificially added to the original training set before training. For example, in experiment 4 of literature [9], the number of samples per category of the IP data set is pre-added to 150 or more. Such DA operation is not combined with the model for dynamic optimization, and the process is cumbersome. Algorithm 1 proposed in this paper can be nested in the batch processing process of the SGD algorithm and plug and play after being packaged as a python class, which is convenient to use. At the same time, the algorithm takes into account the characteristics of the imbalanced distribution of HSI data samples and preferentially selects samples with a small number for DA, which increases the comprehensive classification performance of the model.

Algorithm 1 Dynamic augmentation data selection method

```
Input: $X_{batch} \in R^{m \times M \times N \times C}$, $Y_{batch} \in R^{m \times 1}$, $n \in [1, m]$,
       $X_{DA} \in \varnothing, Y_{DA} \in \varnothing$, $cnt \in R^{2 \times P}$, $index = 0$.
Output: $X_{DA} \in R^{n \times M \times N \times C}$, $Y_{DA} \in R^{n \times 1}$, $X_{new} \in R^{(n+m) \times M \times N \times C}$,
        $Y_{new} \in R^{(n+m) \times 1}$.
1: According to the number of each category in $X_{batch}$, update the value in $cnt$.
2: Sort in ascending order according to the second value of each element in $cnt$.
3: Find the first element in $cnt$ in which the second value is not equal to 0. Update $index$.
4: Traverse the elements in $X_{DA}$ and $Y_{DA}$. If the tag corresponding to an element pair is equal to $cnt[index, 0]$, then add it to $X_{DA}$ and $Y_{DA}$, do $cnt[index, 1] = cnt[index, 1] - 1$.
5: Repeat steps 3 and 4 until the length of $X_{DA}$ and $Y_{DA}$ is equal to $n$.
6: Perform convolution transformation on $X_{DA}$.
7: Merge $X_{DA}$ and $X_{batch}$, $Y_{DA}$ and $Y_{batch}$ to get $X_{new}$, $Y_{new}$. The algorithm ends.
```

D. Siamese Structure Data Augmentation Method

As shown in Fig. 2, after utilizing algorithm 1, the training process uses a Siamese structure with three branches sharing weights. The first branch takes samples selected by algorithm 1 that need to be augmented. They are then inputting to the network of weight sharing. At the second branch, the augmented samples are generated after convolution transformation and input to the network model of weight sharing to produce the soft-max classification result. Through integrating branch one and branch two, the average similarity score is acquired using formula (2). At the same time, the raw batch samples and augmented samples are input into the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2021.3102610, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

network to produce the general category cross-entropy loss function. Finally, the similarity score and the loss function are linearly added to perform gradient descent optimization. The final loss function is defined as:

$$loss_{new} = \alpha \bullet S_{average} + (-\frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{P} Y_j^{true(i)} \log Y_j^{predict(i)}) \qquad (3)$$

Among this, Y_j^{true} refers to the one-hot array of true labels corresponding to X_{new}. $Y_j^{predict}$ refers to the predicted soft-max value of X_{new}. α is a constant, which is set to be 0.1 in this paper.

When predicting, as shown in Fig. 3, it is the same as the general situation. Only the backbone network structure that has been trained is used.

This method considers the difference between the new sample and the original sample and tries to reduce this difference through gradient descent, which weakens the interference of DA samples for model training and strengthens its positive effect on the model. In this way, the classification results of the backbone model are improved.

E. Convolutional Transformation for Data Augmentation

HSI data is often augmented by transformation methods such as rotate, flip, and noise. However, this paper utilizes the convolutional transformation to generate augmented samples. The main advantage of convolutional transformation is that the convolutional kernel can be updated throughout the training process and thus can generate augmented samples with more minor differences relative to the deep classifier. In this way, the DA can be flexible.

The convolutional transformation is depicted by formula (4)[28]:

$$v_{ij}^{xyz} = f(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}) \qquad (4)$$

Where m represents the feature map in layer $i-1$ connected to the current jth feature map. P_i and Q_i are the height and width of the space convolution kernel, respectively. R_i is the depth of the convolution kernel in the spectral dimension. w_{ijm}^{pqr} is the weight at coordinate (p,q,r) connected to the mth feature map. b_{ij} is the bias of the jth feature map at the ith layer. f is the activation function.

The input of convolution transformation is the samples selected from the training set, and the output is the augmented samples.

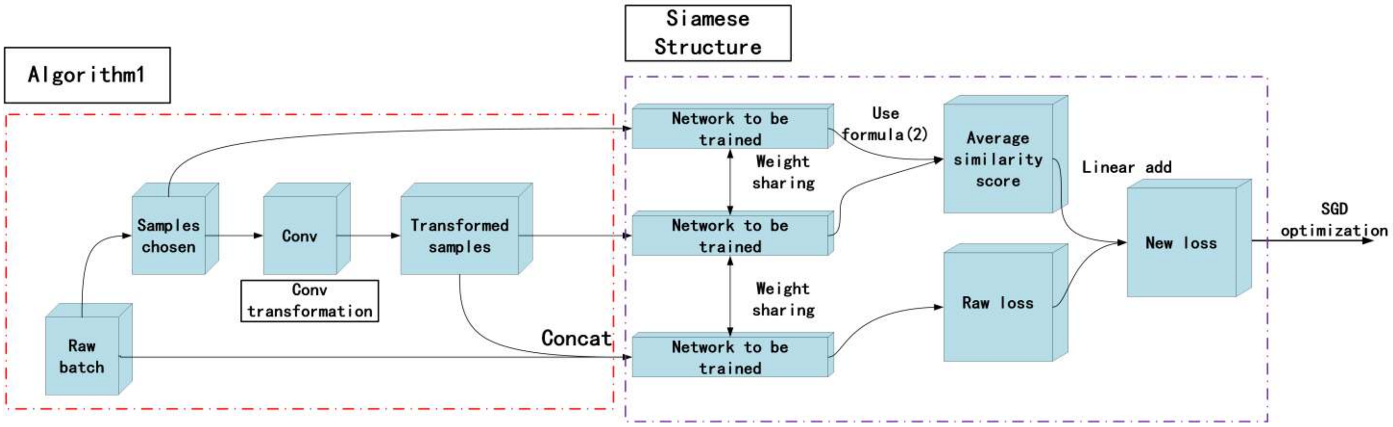

Fig 2. Siamese Structure dynamic data augmentation method when training. Algorithm1 is first utilized to generate augmented samples and the merged new training batch. Then three sub-networks sharing weights are employed to produce the new loss function for SGD optimization.

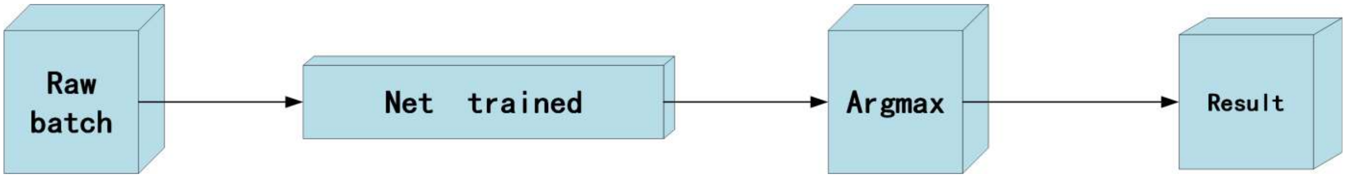

Fig 3. Siamese Structure dynamic data augmentation method when predicting. The testing process is no different to common testing process. The raw batch goes through the trained model to get the classification result.

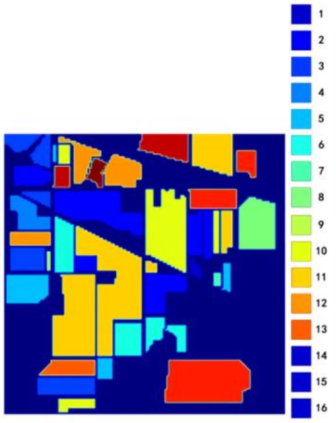

Fig 4. IP ground truth

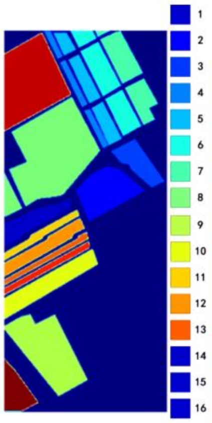

Fig5. SA ground truth

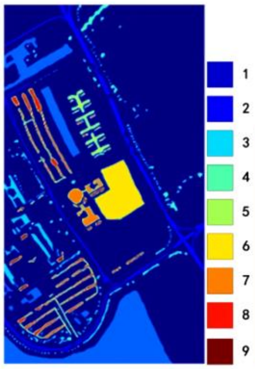

Fig6. PU ground truth

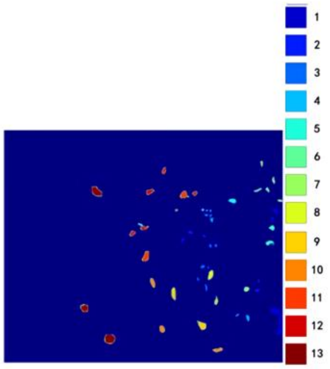

Fig7. KSC ground truth

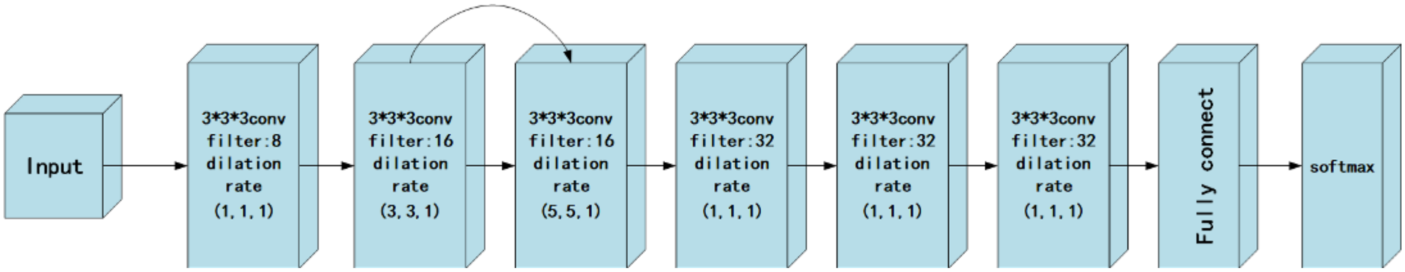

Fig8. Net to be trained. The net is composed of six 3-D convolution kernels with different dilation rates and one jump connection, a fully connect layer and a soft-max layer

III. Experiments and Discussion

In this part, we first introduce four hyperspectral data sets used to study the method's performance in this paper. Then the experiment arrangement and hyper-parameter settings are carried out. Finally, experiments are conducted, and the results are discussed.

A. *Dataset Description*

The IP (Indian Pines) data set was first applied to the study of hyperspectral data classification. The spatial domain size is 145×145, and the spatial resolution is about 20m. After processing, there are 200 bands as the research object of HSI classification. The IP data set has 21,025 pixels, but only 10,249 pixels are pixels of specific features, and the remaining 10,776 pixels are background pixels. In actual classification, only 10249 feature pixels are used as samples, and there are 16 categories.

The SA (Salinas) data set has a spatial domain size of 512×217 and a spatial resolution of about 3.7m. After processing, there are 204 bands as the research object of HSI classification. The SA data set has 111104 pixels, of which only 54129 pixels are pixels of specific features, and the remaining 56975 pixels are background pixels, a total of 16 categories.

The PU (Pavia University) data set has a spatial domain size of 610×340 and a spatial resolution of about 3.7m. After processing, there are 103 bands as the research object of HSI classification. The PU data set has a total of 207,400 pixels, of which only 42776 pixels are specific label pixels, and there are a total of 9 categories.

The KSC (Kennedy Space Center) data set has a spatial domain size of 512×614 and a spatial resolution of about 18m. After processing, there are 176 bands as the research object of HSI classification. The KSC data set has 314,368 pixels, of which only 5,211 pixels are used as classification pixels, and there are 13 categories in total.

The color map of the ground truth distribution of the four data sets is shown in Fig. 4-7, and the category data is shown in Table I-IV.

B. *Experiment Arrangement*

This paper uses the above four HSI data sets for experiments to verify the effectiveness of this method. The software environment of the system is python 3.7.1, tensorflow2.4.1. All experiments were performed on Google AI's Colaboratory platform, using GPU acceleration. All data sets were processed with mean-variance normalization and PCA (Principal component analysis) dimensionality reduction before the experiment, down to 30 bands. The backpropagation algorithm uses the SGD method. The classification evaluation index adopts OA, AA, and Kappa. All experiments were repeated 5 times and averaged.

First, a parameter study is performed. This paper studies the impact of different batch sizes, DA ratios, and the combination of L2 regularization on the classification results. Then, the adaptability of the SSDDA method is studied on five standard CNN models using the most unevenly distributed set: IP and

PU. Next, SSDDA is compared with some state of art classification and augmentation methods. In the ablation study, to verify the performance of the Dynamic augmentation data selection method in improving the classification results, this paper uses three data sets of IP, PU, and KSC to take different training ratios for comparison experiments. To verify the effectiveness of the Siamese structure, we also use three data sets of IP, PU, and KSC to take different training ratios for comparison experiments. Finally, to study the effectiveness of convolution as a DA transformation, we use the SA data set and conduct a comparative experiment.

The network to be trained in the parameter study and ablation study is a three-dimensional CNN network that uses dilated convolutions and a combination of multiple receptive fields. Its structure is shown in Fig. 8.

C. Hyperparameter Settings

Unless otherwise specified, the learning rate of this experiment is set to be 3×10^{-4}. The data block size of the input model is set to be $9\times9\times30$. α in the loss function is set to be 0.1. The batch size is set to be 16. The DA data ratio selected in the dynamic augmentation data selection method is set to be $\frac{1}{4}$. The experiment uses the L2 regularization method. The regularization parameter is set to be 2×10^{-3}.

TABLE I
NUMBER OF SAMPLES ON THE IP DATASET

Class	Name	Number	Total
1	Alfalfa	46	
2	Corn-notill	1428	
3	Corn-mintill	830	
4	Corn	237	
5	Grass-pasture	483	
6	Grass-trees	730	
7	Grass-pasture-moved	28	10249
8	Hay-windrowed	478	
9	Oats	20	
10	Soybean-nottill	972	
11	Soybean-minttill	2455	
12	Soybean-clean	593	
13	Wheat	205	
14	Woods	1265	
15	Buildings-Grass-Trees-Drives	386	
16	Stone-Steel-Towers	93	

TABLE II
NUMBER OF SAMPLES ON THE PU DATASET

Class	Name	Number	Total
1	Asphalt	6631	
2	Meadows	18649	
3	Gravel	2099	
4	Trees	3064	
5	Painted metal sheets	1345	42776
6	Bare Soil	5029	
7	Bitumen	1330	
8	Self-Blocking Bricks	3682	
9	Shadows	947	

TABLE III
NUMBER OF SAMPLES ON THE SA DATASET

Class	Name	Number	Total
1	Brocoli_green_weeds_1	2009	
2	Brocoli_green_weeds_22	3726	
3	Fallow	1976	
4	Fallow_rough_plow	1394	
5	Fallow_smooth	2678	
6	Stubble	3959	
7	Celery	3579	54129
8	Grapes_untrained	11217	
9	Soil_vinyard_develop	6203	
10	Corn_senesced_green_weeds	3278	
11	Lettuce_romaine_4wk	1068	
12	Lettuce_romaine_4wk	1927	
13	Lettuce_romaine_4wk	916	
14	Lettuce_romaine_4wk	1070	
15	Vinyard_untrained	7268	
16	Vinyard_vertical_trellis	1807	

TABLE IV
NUMBER OF SAMPLES ON THE KSC DATASET

Class	Name	Number	Total
1	Scrub	761	5211
2	Willow-swamp	243	
3	CP-hammock	256	
4	Slash-pine	252	
5	Oak/Broadleaf	161	
6	Hardwood	229	
7	Swap	105	
8	Graminoid-marsh	431	
9	Spartina-marsh	520	
10	Cattil-marsh	404	
11	Salt-marsh	419	
12	Mud-flats	503	
13	Water	927	

D. Parameter Study

1) Impact of Batch Size on Results

To better understand and explore the SSDDA method, this part of the experiment studies the impact of different batch sizes on the SSDDA training results. The experiment uses the IP data set, the training set ratio is 15%, and the batch size is set to be 8, 16, 32, and 64, respectively. The result is shown in Fig. 9.

Observation results show that the batch size should not be too large or small when the training utilizes SSDDA. A small batch size will make the model fit slowly. Large batch size will cause a sizeable fitting step size, and the ideal result can hardly be achieved. It can be seen that the best batch size is 16 or 32.

2) The Impact of Augmentation Data Ratio on the Results

In order to explore the impact of the augmentation data ratio on the SSDDA results, this part of the experiment takes the augmentation data ratio respectively $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, 1 on five different models for contrast experiments. Using IP and PU data set, the training proportion of IP is 10%. The training proportion of PU is 1%. The five different models are 3DCNN [10], Resnet [11], DCPN (Double Convolution and Pool Net) [12], MVN (Multi-View Net, noted in Fig. 8), DFFN (Deep Feature Fusion Net) [13]. The result is shown in Fig. 10 and Fig. 11. The results show that the SSDDA method combined with L2 regularization is better than the pure SSDDA method on the three metrics. It can be seen that SSDDA and L2 regularization have good combining performance.

It can be seen from the figures that OA changes with different DA ratios and models. In most cases, OA increases at first and then decreases as a whole. This is because as the proportion of DA samples increases, the number of samples selected by the dynamic selection algorithm is getting closer and closer to the number of the original batch, which weakens the algorithm's function to some extent so that the accuracy will decrease.

To conclude, the best DA ratio depends on the specific class distribution and model. The best da ratios for IP and PU datasets on five different models are shown in Table V, considering OA and stability.

TABLE V
BEST DA RATIO FOR DIFFERENT MODELS

Data Set	MODELS	Train Num	DA Num	DA Ratio
Indian pines	3DCNN	428	428	1
	RESNET		428	1
	DCPN		53	1/8
	MVN		107	1/4
	DFFN		107	1/4
Pavia University	3DCNN	1025	256	1/4
	RESNET		128	1/8
	DCPN		512	1/2
	MVN		512	1/2
	DFFN		256	1/4

3) The Effect of L2 Regularization on Training Results

In order to study the influence of the combination of the SSDDA algorithm and other deep learning methods on the training results, this part of the experiment studies SSDDA with L2 regularization and pure use of the SSDDA method. The experiment uses IP data set, and the proportion of the training set is 10%. The results are shown in Table VI.

The results show that the SSDDA method combined with L2 regularization is better than the pure SSDDA method on the three metrics. It can be seen that SSDDA and L2 regularization have good combining performance.

TABLE VI
L2 REGULARIZATION STUDY ON THE IP DATASET

Metrics	L2+SSDDA (%)	SSDDA (%)
OA	**95.32±0.80**	95.16±0.43
AA	**95.50±0.16**	94.83±1.20
Kappa	**94.67±0.91**	94.48±0.49
1	**100.00±0.00**	99.46±1.08
2	95.25±1.08	95.66±0.76
3	**94.19±1.28**	91.76±1.21
4	**96.41±1.49**	96.12±0.92
5	95.55±1.41	96.38±0.67
6	99.02±0.63	99.02±0.49
7	**98.46±1.88**	93.85±5.76
8	**100.00±0.00**	99.86±0.28
9	84.44±21.78	84.44±14.66
10	95.52±1.71	95.80±0.85
11	93.63±0.66	93.86±0.91
12	92.47±3.01	94.42±2.63
13	92.57±2.60	93.22±2.41
14	**96.42±0.92**	95.89±0.60
15	**96.16±1.41**	92.27±2.30
16	**97.91±1.71**	95.35±5.40

E. *Performance on Different Models*

This section evaluates the adaptability of SSDDA to different CNN models. We utilize five different CNN models, which are 3DCNN [10], Resnet [11], DCPN (Double Convolution and Pool Net) [12], MVN (Multi-View Net, noted in Fig. 8), DFFN (Deep Feature Fusion Net) [13]. Since PU and IP are the most unevenly distributed among the four datasets above, PU and IP have experimented in this part. IP dataset is used with a 10% training ratio. PU dataset is used with a 1% training ratio. Finally, the SSDDA results are compared with common practice. According to the third part of the parameter study, the most suitable DA ratio depends on specific data distribution and model. The ratio is set as the values in Table V.

The results of IP are shown in Table VII. Examples of classification maps are in Fig. 12-16. On 3DCNN, SSDDA has gained 1.82% OA compared with common practice. On Resnet, SSDDA has gained 7.83% OA compared with common practice. On DCPN, SSDDA has gained 0.14% OA compared with common practice. On MVN, SSDDA has gained 0.25% compared with common practice. Finally, on DFFN, SSDDA has gained 0.25% compared with common practice.

The results of PU are shown in Table VIII. Examples of classification maps are in Fig. 17-21. On 3DCNN, SSDDA has gained 2.41 % OA compared with common practice. On Resnet, SSDDA has gained 1.79% OA compared with common practice. On DCPN, SSDDA has gained 0.84% OA compared with common practice. On MVN, SSDDA has gained 1.47% compared with common practice. Finally, on DFFN, SSDDA has gained 1.09 % compared with common practice.

Fig. 12-16 and Fig. 17-21 demonstrate a significant improvement made by SSDDA compared with common practice on IP and PU.

To sum up, the SSDDA method has excellent adaptability to different CNN models on datasets unevenly distributed.

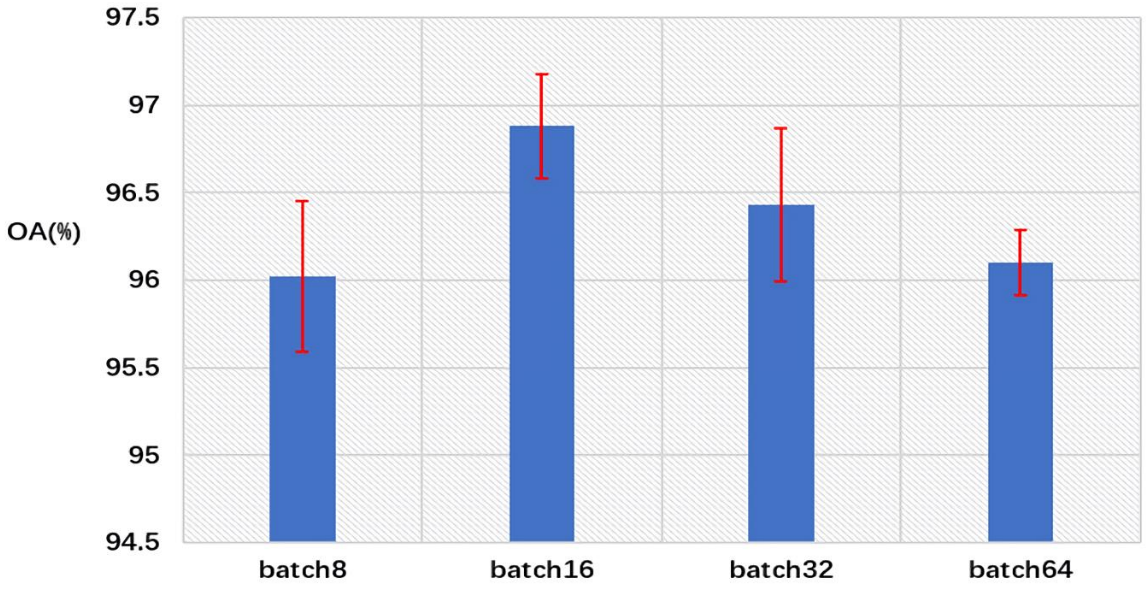

Fig. 9. Result of different batch size on the IP dataset

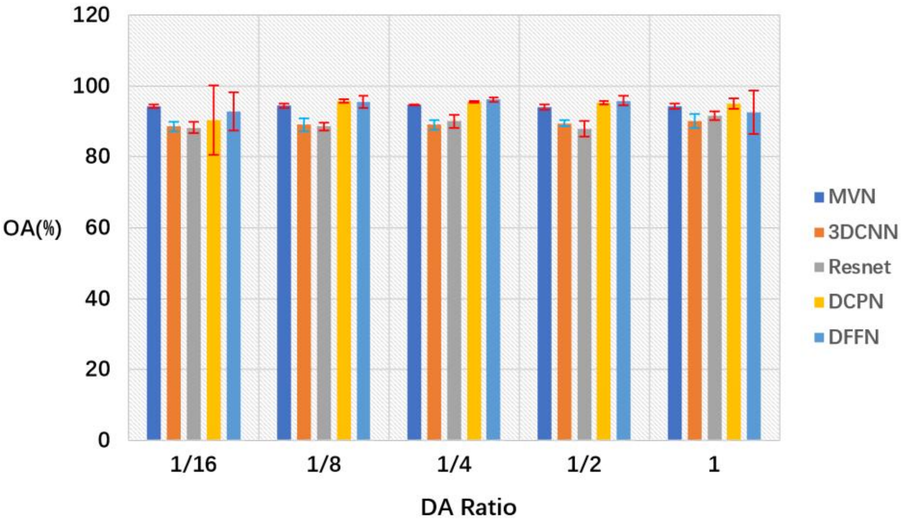

Fig. 10. Result of different DA ratio on the IP dataset

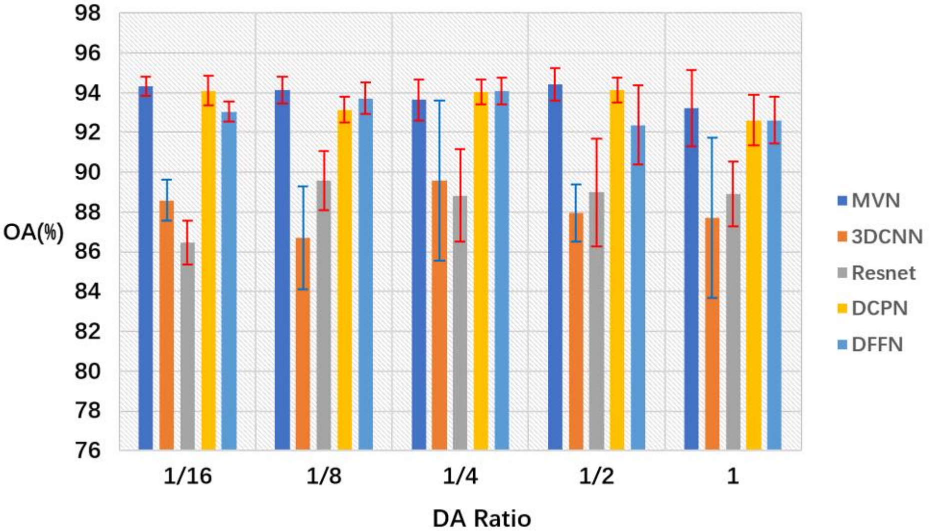

Fig. 11. Result of different DA ratio on the PU dataset

TABLE VII
THE PERFORMANCE OF SSDDA METHOD ON FIVE CNN MODELS ON IP DATASET

	3DCNN		Resnet		DCPN		MVN		DFFN	
Condition	Common	SSDDA	Common	SSDDA	Common	SSDDA	Common	SSDDA	Common	SSDDA
OA	86.45±2.66	**88.27±3.57**	84.48±3.40	**92.31±0.87**	93.18±1.26	**93.32±0.50**	95.39±0.37	**95.64±0.39**	94.91±0.73	**95.16±0.50**
AA	80.12±5.96	**83.67±4.21**	78.73±3.18	**89.54±0.99**	87.01±2.89	**88.05±1.23**	96.02±0.61	**96.11±0.39**	90.12±1.28	**92.13±1.38**
kappa	84.45±3.06	**86.55±4.11**	82.23±3.91	**91.23±0.99**	92.23±1.44	**92.39±0.58**	94.74±0.42	**95.02±0.44**	94.19±0.83	**94.48±0.57**
1	80.00±15.33	**87.00±4.85**	65.50±16.23	**97.00±4.85**	68.37±9.14	**72.56±7.56**	99.49±1.03	98.97±1.26	81.90±7.76	**83.33±3.98**
2	83.25±4.40	**87.02±3.65**	77.09±5.11	**89.80±1.59**	91.21±1.79	**92.29±1.51**	94.02±1.11	**94.54±0.96**	96.08±0.52	95.49±0.61
3	81.73±2.50	**84.34±3.49**	80.43±6.77	**92.86±2.36**	89.32±2.65	87.91±2.76	91.14±2.82	**91.58±0.86**	92.42±3.80	**93.99±0.22**
4	63.03±11.16	**67.58±7.44**	59.73±5.05	**74.57±3.27**	80.19±5.83	**80.86±1.91**	92.19±1.90	**95.05±1.49**	88.13±7.56	85.33±3.14
5	88.04±1.97	**89.00±1.30**	85.92±4.16	**95.67±1.36**	92.55±0.62	90.95±2.99	96.43±1.17	95.82±1.35	89.17±1.71	88.76±3.78
6	98.18±1.26	**98.81±0.79**	98.20±0.83	**99.18±0.46**	98.92±0.56	**98.95±0.26**	98.91±0.47	**99.45±0.21**	96.56±3.48	**98.57±0.47**
7	70.43±22.58	67.83±8.95	81.74±9.68	**96.52±3.25**	63.85±15.50	**79.23±8.63**	99.13±1.74	**100.00±0.00**	90.00±4.25	**96.67±4.86**
8	99.02±1.11	98.74±0.70	95.91±3.47	**99.53±0.25**	99.95±0.09	99.73±0.27	99.86±0.28	99.72±0.37	98.79±1.04	98.70±1.16
9	33.33±12.67	**54.44±10.77**	15.79±7.44	**37.89±11.24**	48.42±19.52	48.42±8.42	100.00±0.00	100.00±0.00	41.05±1.58	**62.11±10.73**
10	79.77±3.59	**79.82±4.05**	76.77±5.75	**86.45±1.49**	89.30±2.73	**92.01±2.01**	94.12±0.62	**94.80±0.53**	91.15±0.96	90.26±1.45
11	91.54±2.38	**92.63±3.54**	87.86±2.45	**91.64±1.61**	94.48±1.27	94.43±0.41	96.67±0.65	96.34±1.06	97.01±0.57	**97.11±1.10**
12	53.08±8.76	**62.69±12.87**	76.13±6.03	**86.98±3.23**	87.12±3.19	**87.27±2.35**	91.49±2.59	**91.98±1.45**	89.08±4.20	**90.00±2.40**
13	99.33±0.65	**99.44±0.61**	99.45±0.70	**100.00±0.00**	95.48±1.30	95.05±1.10	99.88±0.23	99.77±0.28	98.59±1.12	**99.78±0.27**
14	96.70±1.84	95.12±3.03	93.10±3.75	**98.42±1.02**	99.40±0.30	98.87±0.74	96.63±0.63	**97.01±0.80**	97.67±1.02	**97.95±0.67**
15	86.26±7.50	**89.53±4.57**	73.71±9.14	**89.97±2.84**	94.23±2.48	92.74±1.09	92.14±1.80	**93.72±2.64**	98.09±1.67	**99.01±1.21**
16	78.18±17.42	**84.77±11.95**	92.41±4.08	**96.20±3.30**	99.32±0.91	97.50±1.96	94.19±5.93	89.07±6.44	96.28±3.71	**96.98±3.79**

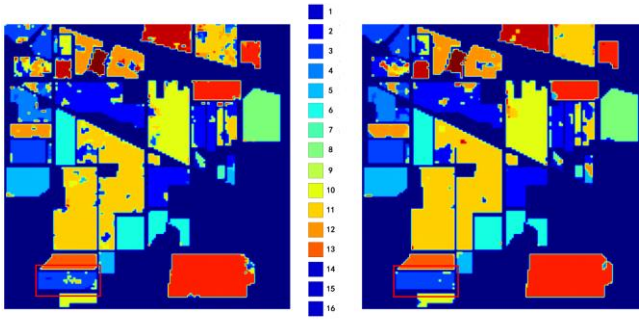

Fig. 12. Classification map of 3DCNN using common practice(left) or SSDDA(right) on IP dataset

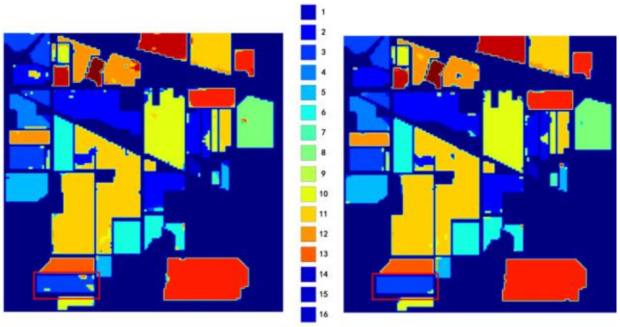

Fig. 13. Classification map of Resnet using common practice(left) or SSDDA(right) on IP dataset

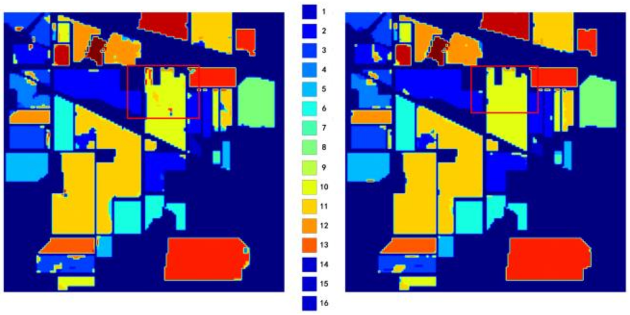

Fig. 14. Classification map of DCPN using common practice(left) or SSDDA(right) on IP dataset

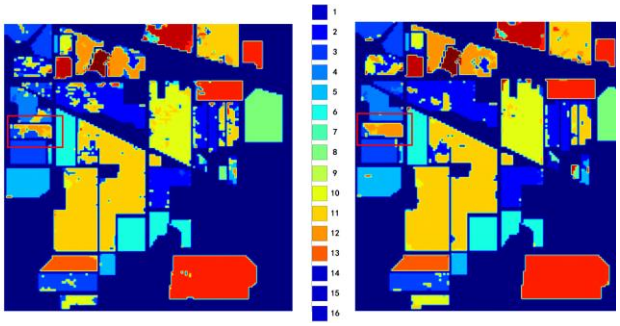

Fig. 15. Classification map of MVN using common practice(left) or SSDDA(right) on IP dataset

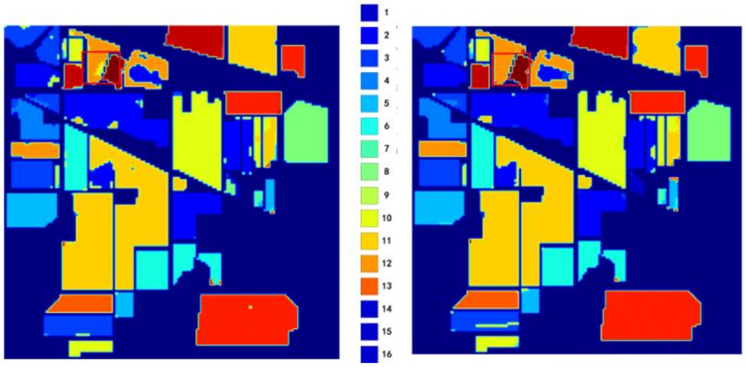

Fig. 16. Classification map of DFFN using common practice(left) or SSDDA(right) on IP dataset

TABLE VIII
THE PERFORMANCE OF SSDDA METHOD ON FIVE CNN MODELS ON PU DATASET

	3DCNN		Resnet		DCPN		MVN		DFFN	
Condition	Common	SSDDA	Common	SSDDA	Common	SSDDA	Common	SSDDA	Common	SSDDA
OA	82.44±2.34	**84.85±2.44**	86.45±2.90	**88.53±1.42**	94.33±0.57	**95.17±0.54**	92.67±0.60	**94.14±0.60**	92.13±0.99	**93.22±0.68**
AA	66.85±2.75	**71.27±3.90**	76.43±5.29	**78.56±3.21**	91.21±0.69	**92.62±0.63**	88.80±1.05	**91.01±1.21**	86.54±1.33	**88.85±1.75**
kappa	76.06±3.33	**79.68±3.27**	81.65±4.02	**84.30±2.91**	92.47±0.77	**93.59±0.71**	90.26±0.81	**92.23±0.79**	89.52±1.33	**91.02±0.91**
1	82.53±2.86	**85.99±2.53**	94.36±2.87	93.89±2.44	93.57±0.59	**95.29±1.25**	91.05±1.42	**92.99±2.66**	90.68±3.78	**91.71±1.42**
2	97.67±0.95	96.53±1.71	97.31±7.71	97.18±0.34	99.25±0.38	98.74±0.59	98.25±0.28	**98.64±0.62**	99.07±0.38	98.15±0.33
3	49.07±6.48	**54.41±8.98**	59.70±8.09	**68.65±6.09**	83.46±2.16	81.54±2.43	79.56±2.88	**80.59±1.06**	81.90±7.19	**86.64±4.08**
4	69.23±3.63	**78.03±3.46**	84.79±3.48	**88.65±1.79**	92.03±2.18	**92.89±1.76**	88.62±5.03	**88.69±0.87**	85.47±4.88	**86.22±2.39**
5	97.67±1.15	**97.76±2.25**	96.63±5.04	**99.22±0.85**	98.97±1.31	98.25±1.38	99.65±0.32	99.56±0.46	99.76±0.38	**99.92±0.08**
6	73.51±6.73	**79.18±4.12**	78.36±6.14	**88.04±2.39**	93.75±2.26	**95.60±1.50**	93.38±1.78	**94.71±1.36**	88.77±4.84	**95.55±1.30**
7	53.67±7.30	**65.15±16.02**	55.46±1.76	**70.09±8.15**	90.72±2.28	**92.18±1.87**	90.80±3.61	**96.01±1.89**	83.30±5.29	**87.61±4.20**
8	71.94±9.02	**76.98±3.78**	58.57±8.19	**60.41±7.77**	80.84±3.14	**86.49±1.98**	78.82±3.13	**85.28±2.71**	82.34±4.59	**82.70±3.00**
9	63.84±5.65	**73.59±5.19**	62.73±19.34	47.32±20.90	88.30±9.53	**92.64±1.45**	79.10±6.09	**82.58±9.43**	67.52±7.14	**71.18±9.38**

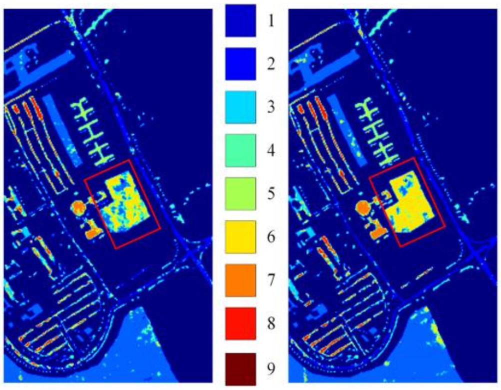

Fig. 17. Classification map of 3DCNN using common practice(left) or SSDDA(right) on PU dataset

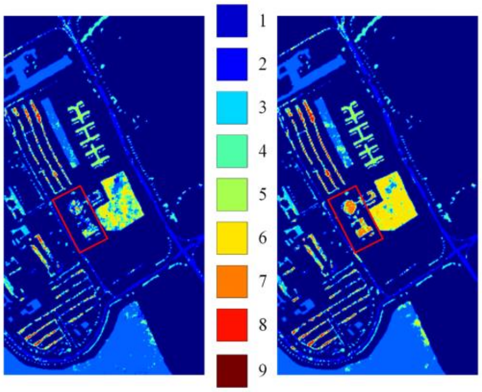

Fig. 18. Classification map of Resnet using common practice(left) or SSDDA(right) on PU dataset

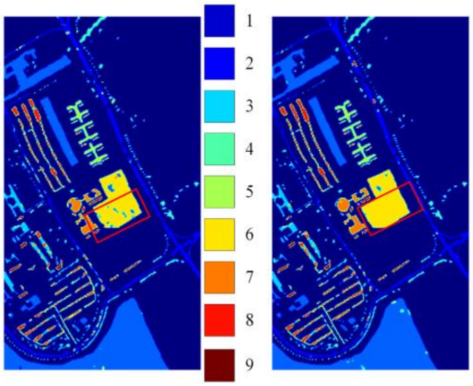

Fig. 19. Classification map of DCPN using common practice(left) or SSDDA(right) on PU dataset

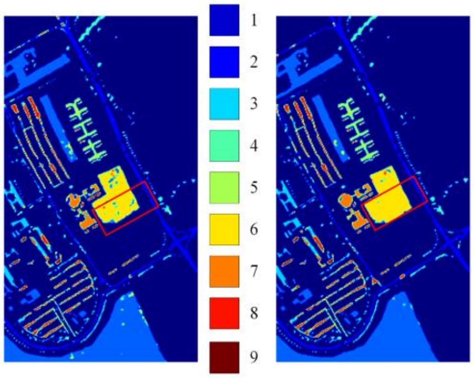

Fig. 20. Classification map of MVN using common practice(left) or SSDDA(right) on PU dataset

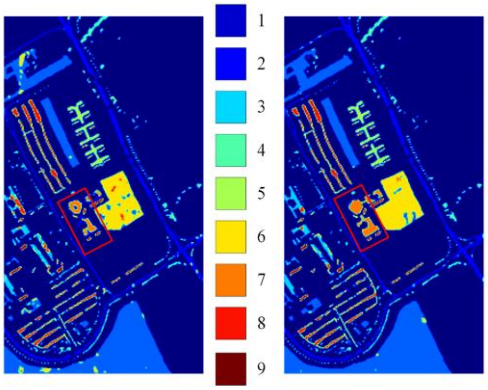

Fig. 21. Classification map of DFFN using common practice(left) or SSDDA(right) on PU dataset

TABLE IX

COMPARISON WITH STATE OF ART METHODS ON IP DATASET

Methods	SVM-GC	SVM-LR	SAE	CNN-MRF	CNN-AL-MRF	CA-GAN	DGCN	RSSAN	SSDDA
OA(%)	62.34±3.34	69.34±3.56	67.35±2.30	75.31±1.25	89.79±1.82	89.22±0.78	88.92±1.45	89.84±0.23	**90.06±0.76**
AA(%)	81.26±2.45	88.61±2.05	83.12±1.75	86.05±1.12	**94.28±0.31**	88.34±0.43	89.34±0.55	89.44±1.12	88.65±1.39

Methods	FLIP	ROTATE	NOISE	OCCLUSION	SSDDA
OA(%)	83.16±0.95	86.40±1.51	92.14±2.75	93.68±1.84	**94.14±0.60**
AA(%)	74.97±1.83	82.90±2.15	80.88±3.10	89.48±3.44	**91.01±1.21**

F. Comparison with State of Art Methods

1）Comparison with Classification Methods

In this part, the IP dataset has been experimented with. SVM-GC [23], SVM-LR [24], SAE [25], CNN-MRF [26], CNN-AL-MRF [27], CA-GAN [29], DGCN [30], RSSAN [31] are compared with SSDDA. The result is in Table IX. The training ratio is set to be 5%.

The table above demonstrates that SSDDA acquires a competitive result compared with state-of-the-art methods regarding samples with a small number and uneven distribution like IP dataset.

2）Comparison with Data Augmentation Methods

In this part, the IP dataset has been experimented with. Data augmentation methods Flip, Rotate, Noise, Occlusion are compared with SSDDA. The result is in Table IX. The training ratio is set to be 10%.

The table above demonstrates that SSDDA acquires a better result than the four state-of-the-art data augmentation methods.

G. Ablation Study

1）Performance Verification of Dynamic Augmentation Data Selection Method

This part uses three data sets: IP, PU, and KSC. Experiments are performed under the conditions of not using the dynamic augmentation data selection method and using the dynamic augmentation data selection method. The training set ratio is set differently. The results are shown in Table X, Table XI, and Table XII.

It can be seen from the classification results of the three data sets below that, in terms of accuracy, the result of using the dynamic augmentation data selection algorithm is better than the unused result. Furthermore, in terms of the stability of accuracy, except for a few cases, the result of the dynamic augmentation data selection algorithm is significantly more stable than the unused result.

TABLE X

DATA SELECTION METHOD VERIFICATION ON THE IP DATASET

Train ratio	Metrics	Selection (%)	No Selection (%)
5%	OA	**90.06±0.76**	88.31±1.18
	AA	**88.65±1.39**	86.64±2.13
	Kappa	**88.68±0.85**	86.68±1.33
10%	OA	**94.75±0.54**	94.12±0.38
	AA	**96.12±0.21**	95.55±0.75
	Kappa	**94.03±0.61**	93.31±0.48
20%	OA	97.69±0.11	97.69±0.17
	AA	**96.33±0.81**	96.23±1.33
	Kappa	97.37±0.13	97.37±0.20

TABLE XI

DATA SELECTION METHOD VERIFICATION ON THE PU DATASET

Train ratio	Metrics	Selection (%)	No Selection (%)
1%	OA	**92.31±1.57**	91.86±0.68
	AA	**86.60±2.96**	86.11±1.90
	Kappa	87.79±2.09	89.17±0.91
2%	OA	**96.34±0.29**	96.27±0.58
	AA	**94.45±0.65**	94.16±1.04
	Kappa	**95.15±0.39**	95.05±0.77
4%	OA	**98.22±0.19**	98.11±0.18
	AA	**96.75±0.31**	96.53±0.25
	Kappa	**97.63±0.25**	97.49±0.24
7%	OA	**98.86±0.17**	98.79±0.09
	AA	**98.22±0.25**	98.16±0.15
	Kappa	**98.48±0.23**	98.39±0.11

TABLE XII

DATA SELECTION METHOD VERIFICATION ON THE KSC DATASET

Train ratio	Metrics	Selection (%)	No Selection (%)
5%	OA	**93.30±1.41**	93.27±1.13
	AA	**88.94±1.67**	88.90±1.31
	Kappa	**92.54±1.57**	92.51±1.26
10%	OA	**95.94±0.64**	95.36±1.03
	AA	**93.15±0.94**	92.13±1.47
	Kappa	**95.48±0.71**	94.84±1.15
15%	OA	**97.28±0.31**	97.17±0.47
	AA	**95.60±0.42**	95.37±0.78
	Kappa	**96.97±0.34**	96.85±0.52
20%	OA	**98.46±0.23**	98.13±0.34
	AA	**97.59±0.45**	96.96±0.45
	Kappa	**98.28±0.26**	97.92±0.39
25%	OA	**98.61±0.43**	98.43±0.31
	AA	**97.70±0.62**	97.31±0.51
	Kappa	**98.45±0.48**	98.26±0.34

In summary, the dynamic expansion data augmentation algorithm can indeed effectively improve the comprehensive classification performance of the model. By dynamically selecting the data in the HSI samples that need to be augmented most, the model's fitting for each type of sample has a comprehensive improvement.

2）Performance Verification of Siamese Structure Data Augmentation Method

This part uses three data sets: IP, PU, and KSC. Experiments were performed under the conditions of using the Siamese structure and not using the Siamese structure during training. Different proportions of the training set were taken. The results are shown in Table XIII, Table XIV, and Table XV.

TABLE XIII

SIAMESE STRUCTURE VERIFICATION ON THE IP DATASET

Train ratio	Metrics	Siamese (%)	No Siamese (%)
5%	OA	**88.41±1.46**	87.82±1.55
	AA	**85.90±3.00**	83.71±3.79
	Kappa	**86.80±1.67**	86.12±1.77
10%	OA	**95.18±0.52**	93.58±0.48
	AA	**90.94±2.24**	89.96±2.02
	Kappa	**94.50±0.59**	92.68±0.55
20%	OA	**98.00±0.15**	96.80±1.64
	AA	**97.07±2.42**	96.36±1.61
	Kappa	**97.73±0.17**	96.36±1.87
35%	OA	**99.08±0.19**	98.94±0.14
	AA	**98.63±0.37**	98.15±0.62
	Kappa	**98.95±0.22**	98.79±0.16

TABLE XIV
SIAMESE STRUCTURE VERIFICATION ON THE PU DATASET

Train ratio	Metrics	Siamese (%)	No Siamese (%)
1%	OA	**93.59±1.35**	91.44±0.73
	AA	**90.61±1.47**	87.84±0.71
	Kappa	**91.48±1.80**	88.60±0.96
2%	OA	**96.17±0.59**	95.24±0.96
	AA	**93.84±0.97**	92.36±1.26
	Kappa	**94.92±0.78**	93.68±1.28
4%	OA	**98.11±0.31**	97.82±0.45
	AA	**96.80±0.43**	96.40±0.58
	Kappa	**97.50±0.41**	97.10±0.60
7%	OA	**98.89±0.06**	97.83±1.40
	AA	**98.06±0.09**	96.35±2.37
	Kappa	**98.52±0.07**	97.13±1.85

TABLE XV
SIAMESE STRUCTURE VERIFICATION ON THE KSC DATASET

Train ratio	Metrics	Siamese (%)	No Siamese (%)
5%	OA	**92.41±0.74**	92.32±0.70
	AA	**87.75±0.86**	87.45±0.86
	Kappa	**91.56±0.83**	91.46±0.78
10%	OA	**96.45±0.59**	95.77±0.90
	AA	**93.82±1.18**	92.91±1.23
	Kappa	**96.04±0.66**	95.29±1.00
15%	OA	**97.06±0.48**	97.02±1.01
	AA	**94.62±0.53**	94.37±1.34
	Kappa	**96.74±0.54**	96.69±1.12
20%	OA	**98.10±0.21**	97.79±0.45
	AA	**96.98±0.33**	96.10±0.62
	Kappa	**97.89±0.23**	97.53±0.51
50%	OA	98.38±0.55	98.43±0.35
	AA	97.11±1.04	97.33±0.61
	Kappa	98.20±0.61	98.25±0.38

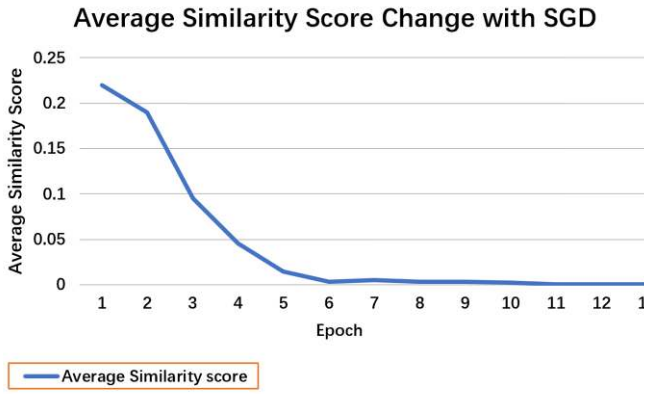

Fig. 22. Average similarity score decreases with SGD

According to the three tables above, it can be seen that the classification results using the Siamese structure is better than the classification results not using the Siamese structure. In addition, in most cases, the stability of the results produced by using the Siamese structure is significantly better than the results produced by not using the Siamese structure. However, when the KSC data training set accounted for 50%, an abnormal situation occurred. This is because the number of samples in each category has reached a satisfactory value, and the Siamese structure is targeted at small number of training set samples and unevenly distributed situation.

Fig. 22 shows the average similarity score change with SGD iterations. With the increase of train epoch, the average similarity score is reduced, which demonstrates that Siamese structure data augmentation method narrows the difference between raw samples and new samples.

In summary, the Siamese structure considers the difference between the new sample and the original sample and manages to reduce this difference through stochastic gradient descent, weakening the interference of DA samples with model training, and indeed strengthening its positive effect on the model.

3）Performance Study of Convolution Transformation for Data Augmentation

This paper uses convolution to transform the data for augmentation. In order to analyze the influence of convolution on the classification results, this experiment compares it with the data augmentation transformation that directly adds noise. The added noise is Gaussian noise with a variance value of 0.8 and a mean value of 0. The data set experimented is the SA data set. The results are shown in Table XVI.

TABLE XVI
CONVOLUTION STUDY ON THE SA DATASET

Train ratio	Metrics	Conv (%)	No Conv (%)
1%	OA	**95.70±0.42**	95.38±0.75
	AA	**96.41±0.30**	95.39±2.46
	Kappa	**95.21±0.46**	94.85±0.83
2%	OA	**97.64±0.47**	97.58±0.14
	AA	**98.61±0.25**	98.62±0.17
	Kappa	**97.37±0.52**	97.30±0.16
4%	OA	**98.83±0.20**	98.40±1.07
	AA	**99.24±0.09**	99.18±0.27
	Kappa	**98.69±0.23**	98.22±1.18
7%	OA	**99.38±0.16**	99.26±0.14
	AA	**99.56±0.09**	99.49±0.09
	Kappa	**99.31±0.18**	99.18±0.15
10%	OA	99.56±0.05	99.58±0.08
	AA	99.66±0.05	99.68±0.04
	Kappa	99.51±0.06	98.53±0.09

According to the above table, it can be seen that the convolution transformation has played a relatively better role for the SA data set. This is because convolution transformation can cooperate with the backbone model to perform SGD iterative optimization, and it is easier to reduce the difference between DA data and original data.

Based on the above experiments, it can be concluded that the SSDDA method proposed in this paper considers the small sample and imbalanced category distribution characteristics of HSI. Using dynamic augmentation data selection algorithm, the classification performance of the backbone model is comprehensively improved. At the same time, the Siamese structure introduced in this paper fully considers the difference between the DA sample and the original sample, and reduces the difference, weakening the interference caused by the difference. Therefore, classification ability of the backbone model is improved.

IV. Conclusion

This paper proposes a dynamic data augmentation method based on a Siamese structure for HSI deep learning classification. This method takes into account the small sample and imbalanced distribution characteristics of HSI. As a result, the dynamic augmentation data selection method compensates for the model's under-fitting for categories with a small number. At the same time, this paper narrows the difference between the DA sample and the original sample to achieve a dynamic balance between the acquisition of diverse information and the deviation of transformed samples from raw samples by introducing the Siamese structure. Experiments on typical HSI data sets show that SSDDA significantly improves the classification results of the backbone model with DA. Furthermore, the SSDDA method and L2 regularization have an excellent performance of combination. Finally, the SSDDA method is proved to have great adaptability to different CNN models on datasets unevenly distributed and get competitive results compared with many states of art classification methods and data augmentation methods.

The performance improvement of the convolutional data augmentation transformation in the SSDDA method on data sets such as IP is not apparent, and subsequent research will consider optimizing the convolutional data augmentation transformation.

References

[1] Y. Chen, Z. Lin, X. Zhao, G. Wang and Y. Gu, "Deep Learning-Based Classification of Hyperspectral Data," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 6, pp. 2094-2107, June 2014, doi: 10.1109/JSTARS.2014.2329330.

[2] Y. Chen, X. Zhao and X. Jia, "Spectral–Spatial Classification of Hyperspectral Data Based on Deep Belief Network," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2381-2392, June 2015, doi: 10.1109/JSTARS.2015.2388577.

[3] Y. Chen, H. Jiang, C. Li, X. Jia and P. Ghamisi, "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks," in IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232-6251, Oct. 2016, doi: 10.1109/TGRS.2016.2584107.

[4] A. Sellami and I. R. Farah, "Spectra-spatial Graph-based Deep Restricted Boltzmann Networks for Hyperspectral Image Classification," 2019 PhotonIcs & Electromagnetics Research Symposium - Spring (PIERS-Spring), Rome, Italy, 2019, pp. 1055-1062, doi: 10.1109/PIERS-Spring46901.2019.9017309.

[5] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang and X. Huang, "Hyperspectral Image Classification With Deep Learning Models," in IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 9, pp. 5408-5423, Sept. 2018, doi: 10.1109/TGRS.2018.2815613.

[6] S. Liu and Q. Shi, "Multitask Deep Learning With Spectral Knowledge for Hyperspectral Image Classification," in IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 12, pp. 2110-2114, Dec. 2020, doi: 10.1109/LGRS.2019.2962768.

[7] Y. Cai, Z. Dong, Z. Cai, X. Liu and G. Wang, "Discriminative Spectral-Spatial Attention-Aware Residual Network For Hyperspectral Image Classification," 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, Netherlands, 2019, pp. 1-5, doi: 10.1109/WHISPERS.2019.8921022.

[8] K. Pooja, R. R. Nidamanuri and D. Mishra, "Multi-Scale Dilated Residual Convolutional Neural Network for Hyperspectral Image Classification," 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, Netherlands, 2019, pp. 1-5, doi: 10.1109/WHISPERS.2019.8921284.

[9] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza and F. Pla, "Deep Pyramidal Residual Networks for Spectral–Spatial Hyperspectral Image Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 2, pp. 740-754, Feb. 2019, doi: 10.1109/TGRS.2018.2860125.

[10] Li. Y, H. Zhang, and Q. Shen. "Spectral-spatial Classification of Hyperspectral Imagery with 3d Convolutional Neural Network, " in Remote Sensing, vol. 9, no. 1, pp.67, Jan. 2017, doi:10.3390/rs9010067.

[11] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition," in arXiv:1408.5093, 2014.

[12] G. Li, C. Zhang, F. Gao, X. Zhang. "3DCNN hyperspectral remote sensing image classification method based on double convolution pooling structure, " in Chinese Journal of image and graphics, vol. 24, no. 04, pp.639-654, 2019, doi: CNKI:SUN:ZGTB.0.2019-04-014.

[13] W. Song, S. Li, L. Fang and T. Lu, "Hyperspectral Image Classification With Deep Feature Fusion Network," in IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 6, pp. 3173-3184, June 2018, doi: 10.1109/TGRS.2018.2794326.

[14] L. Gao, D. Hong, J. Yao, B. Zhang, P. Gamba and J. Chanussot, "Spectral Super resolution of Multispectral Imagery With Joint Sparse and Low-Rank Learning," in IEEE Transactions on Geoscience and Remote

Sensing, vol. 59, no. 3, pp. 2269-2280, March 2021, doi: 10.1109/TGRS.2020.3000684.

[15] B. Zhao, L. Gao and B. Zhang, "An optimized method of kernel minimum noise fraction for dimensionality reduction of hyperspectral imagery," 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 2016, pp. 48-51, doi: 10.1109/IGARSS.2016.7729003.

[16] L. Gao, Dan Yao, Q. Li, L. Zhuang, B. Zhang, J. Bioucas-Dias. " A new low-rank representation based hyperspectral image denoising method for mineral mapping, " Remote Sensing, 2017, 9(11), 1145.

[17] L. Gao et al., "Subspace-Based Support Vector Machines for Hyperspectral Image Classification," in IEEE Geoscience and Remote Sensing Letters, vol. 12, no. 2, pp. 349-353, Feb. 2015, doi: 10.1109/LGRS.2014.2341044.

[18] D. Hong et al., "More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification," in IEEE Transactions on Geoscience and Remote Sensing, doi: 10.1109/TGRS.2020.3016820.

[19] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza and J. Chanussot, "Graph Convolutional Networks for Hyperspectral Image Classification," in IEEE Transactions on Geoscience and Remote Sensing, doi: 10.1109/TGRS.2020.3015157.

[20] K. Zheng et al., "Coupled Convolutional Neural Network With Adaptive Response Function Learning for Unsupervised Hyperspectral Super Resolution," in IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 3, pp. 2487-2502, March 2021, doi: 10.1109/TGRS.2020.3006534.

[21] L. Zhuang, L. Gao, B. Zhang, X. Fu and J. M. Bioucas-Dias, "Hyperspectral Image Denoising and Anomaly Detection Based on Low-Rank and Sparse Representations," in IEEE Transactions on Geoscience and Remote Sensing, doi: 10.1109/TGRS.2020.3040221.

[22] X. Sun et al., "Target Detection Through Tree-Structured Encoding for Hyperspectral Images," in IEEE Transactions on Geoscience and Remote Sensing, doi: 10.1109/TGRS.2020.3024852.

[23] Y. Tarabalka, M. Fauvel, J. Chanussot and J. A. Benediktsson, "SVM- and MRF-Based Method for Accurate Classification of Hyperspectral Images," in IEEE Geoscience and Remote Sensing Letters, vol. 7, no. 4, pp. 736-740, Oct. 2010, doi: 10.1109/LGRS.2010.2047711.

[24] Y. Xu, Z. Wu and Z. Wei, "Spectral–Spatial Classification of Hyperspectral Image Based on Low-Rank Decomposition," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2370-2380, June 2015, doi: 10.1109/JSTARS.2015.2434997.

[25] Zhouhan Lin, Yushi Chen, Xing Zhao and Gang Wang, "Spectral-spatial classification of hyperspectral image using autoencoders," 2013 9th International Conference on Information, Communications & Signal Processing, Tainan, Taiwan, 2013, pp. 1-5, doi: 10.1109/ICICS.2013.6782778.

[26] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu and J. Paisley, "Hyperspectral Image Classification With Markov Random Fields and a Convolutional Neural Network," in IEEE Transactions on Image Processing, vol. 27, no. 5, pp. 2354-2367, May 2018, doi: 10.1109/TIP.2018.2799324.

[27] X. Cao, J. Yao, Z. Xu and D. Meng, "Hyperspectral Image Classification With Convolutional Neural Network and Active Learning," in IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 7, pp. 4604-4616, July 2020, doi: 10.1109/TGRS.2020.296462

[28] C. Wang, N. Ma, Y. Ming, Q. Wang and J. Xia, "Classification of hyperspectral imagery with a 3d convolutional neural network and j-m distance," in Advances in Space Research, doi:10.1016/j.asr.2019.05.005.

[29] J. Feng, X. Feng, J. Cheng etc., "Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification, " in Remote Sensing 12.7 (2020): 1149.

[30] X. He, Y. Chen and P. Ghamisi, "Dual Graph Convolutional Network for Hyperspectral Image Classification With Limited Training Samples," in IEEE Transactions on Geoscience and Remote Sensing, doi: 10.1109/TGRS.2021.3061088.

[31] M. Zhu, L. Jiao, F. Liu, S. Yang and J. Wang, "Residual Spectral–Spatial Attention Network for Hyperspectral Image Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 1, pp. 449-462, Jan. 2021, doi: 10.1109/TGRS.2020.2994057.

[32] D. Hong, N. Yokoya, J. Chanussot and X. X. Zhu, "An Augmented Linear Mixing Model to Address Spectral Variability for Hyperspectral Unmixing," in IEEE Transactions on Image Processing, vol. 28, no. 4, pp. 1923-1938, April 2019, doi: 10.1109/TIP.2018.2878958.

[33] D. Hong et al., "Interpretable Hyperspectral Artificial Intelligence: When nonconvex modeling meets hyperspectral remote sensing," in IEEE Geoscience and Remote Sensing Magazine, vol. 9, no. 2, pp. 52-87, June 2021, doi: 10.1109/MGRS.2021.3064051.

[34] D. Hong et al., "Learning to propagate labels on graphs: An iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction," in ISPRS Journal of Photogrammetry and Remote Sensing, vol. 158, pp.35-49, December 2019, doi: 10.1016/j.isprsjprs.2019.09.008.

Hongmin Gao received the Ph.D. degree from Hohai University, Nanjing, China, in 2014.

He is a Professor with the College of Computer and Information, Hohai University. His research interests include deep learning, information fusion, and image processing in remote sensing.

Junpeng Zhang received the B.S. degree in communication engineering from Hohai University, Nanjing, China, in 2020.

He is a Graduate Student with the College of Computer and Information, Hohai University. His research interests include deep learning and image processing.

Chenming Li received the B.S., M.S., and Ph.D. degrees in computer application technology from Hohai University, Nanjing, China, in 1993, 2003, and 2010, respectively.

He is a Professor and the Deputy Dean of the College of Computer and Information, Hohai University. His research interests include information processing systems and applications, system modeling and simulation, multisensor systems, and information processing.

Dr. Li is a Senior Member of the China Computer Federation and the Chinese Institute of Electronics.