Global Prototypical Network for Few-Shot Hyperspectral Image Classification

Chengye Zhang, Jun Yue, and Qiming Qin

***Abstract*—This article proposes a global prototypical network (GPN) to solve the problem of hyperspectral image classification using limited supervised samples (i.e., few-shot problem). In the proposed method, a strategy of global representation learning is adopted to train a network (f_θ) to transfer the samples from the original data space to an embedding-feature space. In the new feature space, a vector called global prototypical representation for each class is learned. In terms of the network (f_θ), we designed an architecture of a deep network consisting of a dense convolutional network and the spectral–spatial attention network. For the classification, the similarities between the unclassified samples and the global prototypical representation of each class are evaluated and the classification is finished by nearest neighbor classifier. Several public hyperspectral images were utilized to verify the proposed GPN. The results showed that the proposed GPN obtained the better overall accuracy compared with existing methods. In addition, the time expenditure of the proposed GPN was similar with several existing popular methods. In conclusion, the proposed GPN in this article is state-of-the-art for solving the problem of hyperspectral image classification using limited supervised samples.**

***Index Terms*—Deep Learning, dense convolution, global representations, hyperspectral image classification, small number of samples, spectral–spatial attention.**

I. Introduction

COMBINING both spatial and spectral information, hyperspectral remote sensing has been widely used in many areas, e.g., agriculture, geology, environment, and ecology [1], [2]. Hyperspectral image classification is to identify the target type of each pixel, which is usually a key step in the applications of hyperspectral remote sensing in many areas.

Machine learning provides an important way for hyperspectral image classification automatically. In the past several decades, many machine learning approaches were proposed to solve the problem of hyperspectral image classification, such as support vector machine (SVM) [3], [4], neural network [5], and random forest (RF) [6], [7]. Since the deep learning was proposed [8], it has achieved great success to solve many important problems, such as face recognition, speech recognition, image identification, and automatic translation. After the application of deep learning on hyperspectral image classification [9], the methods for hyperspectral image classification based on deep learning attracted the attention of many scholars. A series of deep networks were successively proposed and the classifying accuracy of hyperspectral image was gradually improved [10]–[16]. To make full use of the spatial–spectral information provided by the hyperspectral image, some deep-learning models fused both the spatial and spectral features, and extracted the comprehensive features, which also improved the classification accuracy [17]–[23].

However, the hyperspectral image classification has not been solved very well using deep learning. One of the most important reasons is that a mass of labeled samples are required by deep learning for a satisfying accuracy. In fact, it is usually very difficult to acquire enough labeled samples to meet the requirement of hyperspectral image classification using deep learning. In other words, the limited number of training samples hinders the scholars from improving the accuracy of hyperspectral image classification.

In this situation, a kind of methods called few-shot learning was introduced into solving the problem of hyperspectral image classification with limited samples, and obtained better accuracy when compared with previous machine learning methods [24], [25]. Few-shot learning is an important branch of machine learning, which is designed for solving the problems with only a few training samples, such as object identification, face recognition, image classification, etc. [26]–[30].

For hyperspectral image classification, several methods have been proposed to solve the problem of limited supervised samples. A pixel-pair method was proposed to construct a new data pair combination, which increased the amount of input data for training [31]. Limited to the number of training samples, an unsupervised method called self-taught feature learning was proposed to finish hyperspectral image classification [32]. Sensor-specific models were trained and directly applied on

Manuscript received June 8, 2020; revised July 16, 2020, August 4, 2020, and August 10, 2020; accepted August 13, 2020. Date of publication August 18, 2020; date of current version August 28, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 41901291, in part by the Aeronautical Observation System in the High Resolution Earth Observation System of the National Science and Technology Major Project of China under Grant 30-H30C01-9004-19/21, in part by the National Key Research and Development Program under Grant 2018YFC1800102, in part by the Open Fund of State Key Laboratory of Coal Resources and Safe Mining under Grant SKLCRSM19KFA04, in part by the National Facilities and Information Infrastructure for Science and Technology under Grant Y719H71006, and in part by the Fundamental Research Funds for the Central Universities under Grant 2019QD06. *(Corresponding author: Jun Yue.)*

Chengye Zhang is with the State Key Laboratory of Coal Resources and Safe Mining and the College of Geoscience and Surveying Engineering, China University of Mining and Technology, Beijing 100083, China (e-mail: czhang@cumtb.edu.cn).

Jun Yue is with the School of Traffic and Transportation Engineering, Changsha University of Science and Technology, Changsha 410114, China (e-mail: jyue@pku.edu.cn).

Qiming Qin is with the School of Earth and Space Sciences, Peking University, Beijing 100871, China (e-mail: qmqin@pku.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.3017544

target datasets when tuned by only a few training samples [33]. A semisupervised convolutional neural network (CNN) and a supervised deep feature extraction method were proposed to realize hyperspectral image classification using 200 training samples in each class [34], [35].

In few-shot image classification, there are a group of base classes and a group of new classes. The common strategy of few-shot learning is to train the model based on enough labeled samples in the base classes. Then, the trained model is generalized to the new classes, and is used to realize the classification based on limited supervised samples in the new classes. In other words, the classification for "new classes" with limited supervised samples is the task, whereas the "base classes" with enough labeled samples are utilized to help training the model. A deep few-shot learning (DFSL) method and a spatial–spectral prototypical network were proposed, respectively, [24], [36] for the few-shot hyperspectral image classification. The common basic idea of these two methods is to learn a generalized feature space using large amount of labeled samples and then be applied to new datasets with limited supervised samples, which is similar to this article. However, the differences with this article are in the method of feature extraction. This article utilizes the global prototypical learning with hallucinating new samples in the new classes and realizes classification based on global prototypical representations, and also designs the dense CNN and the spectral–spatial attention network (SSAN) to alleviate the vanishing of gradient information and adaptively adjust the receptive field size. In fact, there are two kinds of few-shot learning: First, the model is trained using base classes and then generalized to the new classes; second, the model is trained using both base classes and new classes, and then the trained model is used to realize the classification of new classes. However, a serious problem is present in these methods. That is the disproportion of the number of labeled samples between the base classes and new classes. In this situation, the classification method is vulnerable to over fit to the base classes [37]. To alleviate this problem, it is a strategy to hallucinate new samples in the new classes and then to learn the global representation for each class.

This article proposes a global prototypical network (GPN) to solve the problem of hyperspectral image classification with limited supervised samples. The proposed method combines the learning strategy for global prototypical representations and a novel deep network.

The main contributions of this article are listed as follows.

1) A method of global representation learning is proposed to train a network, which is to learn a global prototypical representation for each class in a new feature space. The learning procedure includes hallucinating new samples, generating episodic representations, and updating global prototypical representations.
2) A novel architecture of deep CNN is designed, including two branches: a dense convolutional network and a spectral–spatial attention network. The dense convolutional network is to alleviate the vanishing of gradient information after passing through many layers, whereas the SSAN is to adaptively adjust the receptive field size.
3) The experiments show that the overall accuracy (OA) of the proposed method is better than that of existing methods. The ablation study demonstrates the effectiveness of the global representation learning, the dense convolutional network, and the spectral–spatial attention network, for improving the classification accuracy.

This article is organized as follows. In Section II, the proposed method is explained in detail, including the global representation learning and the two-branch deep CNN. In Section III, the experimental data and parameters are described. In Section IV, the results are showed and discussed, and the ablation study and time consumption are analyzed. In Section V, the conclusions of this article are summarized.

II. Method

A. Procedure of the Proposed GPN

The procedure of the proposed GPN is shown in Fig. 1, including training the network and testing the network. In training, the base classes and new classes (after hallucinating new samples) are used for training a deep network to learn a feature space and also to learn the global prototypical representation in this space for each class. In testing, the unclassified samples are transformed to the embedding-feature space, and are classified according to the similarities with the global prototypical representations based on the nearest neighbor (NN) classifier.

B. Global Representation Learning

The global representation learning is to learn a function (i.e., network) f_θ: $\mathrm{R}^F \rightarrow \mathrm{R}^D$ to transfer the samples from original data space R^F to an embedding-feature space R^D. In the embedding-feature space, each sample is a high-dimensional vector, and a vector called global prototypical representation is learned for each class during training. The number of dimensions of the embedding-feature space is denoted as d.

Before training, ten new samples need to be hallucinated from each new class. For the new class c_j, k_r samples are randomly selected from all the k_t samples in c_j. The selected ith sample can be extracted an embedding feature by the network, which is denoted as f_i. The new sample t_{cj} hallucinated from the new class c_j is defined as (1). ceil $(\cdot)$ is the function to calculate the rounded-up integer. U (0, u) is a uniform distribution ranging from 0 to u

$$t_{cj} = \sum_{i=1}^{k_r} \frac{\tau_i}{\sum_j \tau_j} \cdot f_i \tag{1}$$

$k_r = \text{ceil}\,(k_r{}')$, $k_r{}' \sim \mathrm{U}(0, k_t)$, $\tau_i \sim \mathrm{U}(0, 1)$.

The procedure for training a batch is shown in Table I. $C_{\text{all}} = \{c_1, c_2, c_3, \ldots, c_n\}$ is the labels set of all the classes and n is the number of all the classes, consisting of the labels set of the base classes $C_{\text{base}} = \{c_1, c_2, c_3, \ldots, c_s\}$ and the labels set of the new classes $C_{\text{new}} = \{c_{s+1}, c_{s+2}, c_{s+3}, \ldots, c_{s+t}\}$. s is the number of base classes and t is the number of new classes ($n = s + t$). $\mathbf{g}\,(c_i)$ is the global prototypical representation of the class c_i. $\mathbf{G} = \{\mathbf{g}\,(c_1), \mathbf{g}\,(c_2), \mathbf{g}\,(c_3), \ldots, \mathbf{g}\,(c_n)\}$ is a $d \times n$ matrix consisting of the global prototypical representations of all the

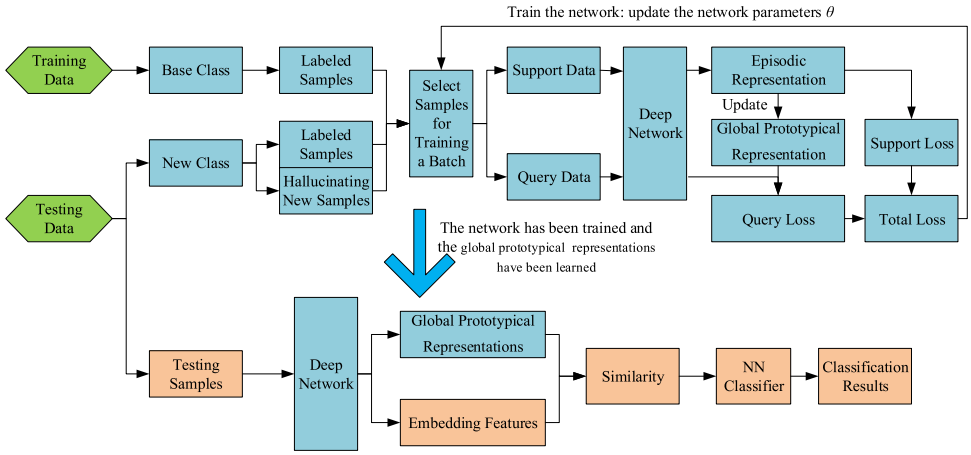


Fig. 1. Procedure of the proposed GPN in this article.

TABLE I
PROCEDURE FOR TRAINING A BATCH

Input: The labeled samples set of all the class $T=\{t_1, t_2, t_3, \ldots, t_m\}$
The labels of all the classes $C_{all} = \{c_1, c_2, c_3, \ldots, c_n\}$
The initialized or updated the global prototypical representations of all the classes, **G**.
The initialized or updated learnable parameter θ

Randomly select n_{batch} classes from C_{all} to form C_{batch}
u samples is randomly selected from each class in C_{batch} to form a support dataset S
v samples is randomly selected from each class in C_{batch} to form a query dataset Q
For each sample in the support dataset S **DO**
  Extract the embedding feature (a $d\times1$ vector) of the sample by the network f_θ: $R^F \rightarrow R^D$
End For
For $c_i \in C_{batch}$ **DO**
  Calculate the mean of embedding features in class c_i as the episodic representation $\mathbf{e}(c_i)$
End For
For $c_i \in C_{batch}$ **DO**
  For $c_j \in C_{all}$ **DO**
    Calculate the similarity score h_i^j between $\mathbf{e}(c_i)$ and $\mathbf{g}(c_j)$ using Equation (4)
  End For
  Calculate the support loss L_s^i using the Equation (5)
  Calculate the probability distribution $\mathbf{P}_i$ using Equation (6)
  Update the global prototypical representation of class c_i using Equation (7)
End For
For $q_k \in Q$ **DO**
  For $c_i \in C_{batch}$ **DO**
    Calculate the similarity score w_k^i between q_k and $\mathbf{g}_{update}(c_j)$ using Equation (8)
  End For
  Calculate the query loss L_q^k using Equation (9)
End For
Calculate the total loss using Equation (10)
Update: $\theta=\theta-\alpha\nabla_\theta L_{total}$

Output: θ, **G**, L_{total}

classes. The initialization of $\mathbf{g}(c_i)$ is the mean of embedding features (extracted by the initialized network) of all the labeled samples in class c_i (2). $T = \{t_1, t_2, t_3, \ldots, t_m\}$ is the labeled samples set of all the classes and m is the number of all the labeled samples in both the base classes and new classes. $\mathbf{x}_i = f_\theta(t_i)$ is a vector representing the sample t_i in the new space R^D and the dimension is $d \times 1$

$$\mathbf{g}_{\text{initial}}(c_i) = \frac{\sum f_\theta(s_g)}{\text{Num}_i} \tag{2}$$

where s_g represents the labeled samples in class c_i and Num_i represents the number of all the labeled samples in class c_i.

In a batch, some classes are randomly selected from C_{all} to form C_{batch}, and the number of selected classes is denoted as n_{batch}. Then, a part of samples are randomly selected from each class in C_{batch} to form a support dataset $S = \{s_1, s_2, s_3, \ldots, s_a\}$, and then another part of samples are also randomly selected from each class in C_{batch} to form a query dataset $Q = \{q_1, q_2, q_3, \ldots, q_b\}$. The number of samples per class in the support dataset S and the query dataset Q is denoted as u and v, respectively. Thus, the total number of samples in S and Q is $a = u \times n_{\text{batch}}$ and $b = v \times n_{\text{batch}}$, respectively. The embedding spatial–spectral feature (a $d \times 1$ vector) of each sample in the support dataset S is extracted by the deep network f_θ: $R^F \rightarrow R^D$. The mean

of embedding features of the samples in class c_i is computed and regarded as the episodic representation $\mathbf{e}(c_i)$ of class c_i in this batch (3). The episodic representations of all the class in this batch form the matrix $\mathbf{E} = \{\mathbf{e}(c_i),\ c_i \in C_{\text{batch}}\}$, which is a $d \times n_{\text{batch}}$ matrix

$$\mathbf{e}(c_i) = \frac{\sum f_\theta(s_{\text{e}})}{u} \tag{3}$$

where s_{e} represents the labeled samples in class c_i in a batch.

According to (4), the similarity score $\mathbf{H}_i = [h_i^1, h_i^2, h_i^3, \cdots, h_i^n]^{\text{T}}$ between the episodic representation of class c_i and each global prototypical representation in $\mathbf{G} = \{\mathbf{g}(c_1), \mathbf{g}(c_2), \mathbf{g}(c_3), \ldots, \mathbf{g}(c_{\text{n}})\}$ is calculated

$$h_i^j = -||\delta(\mathbf{e}(c_i)) - \varphi(\mathbf{g}(c_j))||_2, c_j \in C_{\text{all}} \tag{4}$$

where $\delta(\cdot)$ and $\varphi(\cdot)$ are embeddings for the episodic representation and the global prototypical representation, respectively.

In this article, a loss for class c_i in support dataset (called support loss) is defined as

$$L_s^i = \text{CE}(c_i, \mathbf{H}_i) \tag{5}$$

where CE$(\cdot)$ is a cross entropy loss.

The similarity score $\mathbf{H}_i = [h_i^1, h_i^2, h_i^3, \cdots, h_i^n]^{\text{T}}$ is normalized to a probability distribution $\mathbf{P}_i = [p_i^1, p_i^2, p_i^3, \cdots, p_i^n]^{\text{T}}$ by

$$p_i^j = \frac{e^{h_i^j}}{\sum_{j=1}^n e^{h_i^j}}. \tag{6}$$

The global prototypical representation of each class is then updated according to

$$\mathbf{g}_{\text{update}}(c_i) = \mathbf{G}\mathbf{P}_i \tag{7}$$

where $\mathbf{G}$ is a $d \times n$ matrix consisting of the global prototypical representations before being updated, and $\mathbf{P}_i$ is a $n \times 1$ vector.

The similarity score $\mathbf{W}_k = [w_k^1, w_k^2, w_k^3, \cdots, w_k^{n_{\text{batch}}}]^{\text{T}}$ between the updated global prototypical representation and the sample q_k in the query dataset Q is defined as

$$w_k^i = -||f_\theta(q_k) - \mathbf{g}_{\text{update}}(c_i)||_2, c_i \in C_{\text{batch}}. \tag{8}$$

A loss for the sample q_k in the query dataset Q (called query loss) is defined as

$$L_q^k = \text{CE}(C(q_k), \mathbf{W}_k) \tag{9}$$

where CE$(\cdot)$ is a cross entropy loss. $C(q_k)$ belongs to the set C_{batch} and is the class type of the sample q_k.

Based on the support loss and the query loss, the total loss in this article is defined as

$$L_{\text{total}} = \sum_{i=1}^{n_{\text{batch}}} L_s^i + \sum_{k=1}^{v} L_q^k. \tag{10}$$

At the last of training a batch, the parameters of the network θ are updated based on the total loss. The learning rate α of the network is set as 1×10^{-3}, whereas the weight decay and momentum of the proposed network are set as 1×10^{-4} and 9×10^{-1}, respectively.

C. Deep Networks

The architecture of the deep network (f_θ: $\text{R}^F \to \text{R}^D$) is shown in Fig. 2, which consists of two parts: a dense convolutional network and an SSAN.

1) Dense Convolution: For a normal convolutional network, a connection point is present only between two neighbor layers. In other words, there are $n-1$ connections in a normal convolutional network with n layers. However, in a normal convolutional network, the information about gradient can vanish after passing through many layers [38]. In this article, a dense convolutional network is designed with five convolutional layers (see Fig. 2). The difference from a normal convolutional network is that more connections are present, i.e., ten connections are in a five-layer network (see Fig. 2): the first layer is connected to the second, third, fourth, and fifth layer, respectively; the second layer is connected to the third, fourth, and fifth layer, respectively; the third layer is connected to the fourth and fifth layer, respectively; and the fourth layer is connected to the fifth layer.

2) Spectral–Spatial Attention Network: In a normal convolutional network, the size of the convolutional kernel is settled, resulting in a constant size of receptive field for a layer. However, from the view of optic nerve science, the size of receptive field should be different for different nerve cells in a same layer, to response to different stimulations. In this article, an SSAN is proposed to realize different sizes of receptive fields in a same layer. The strategy of SSAN is based on selective kernel network [39]. Kernels with different sizes are applied on a layers, respectively, and the size of receptive field is adjusted adaptively. The SSAN in this article consists of three sections: split, fuse, and select (see Fig. 2).

a) Split: In this section, the SSAN is split to two branches. The one is a convolutional block using a $3 \times 3 \times 3$ kernel, and the other is to use a $5 \times 5 \times 5$ kernel. To improve the operational efficiency, the $5 \times 5 \times 5$ convolution is realized by dilated convolution [40].

b) Fuse: The number of spectral bands of the hyperspectral image is denoted as N. The convolutional results from two branches are fused by the following ("Fuse 1" module in Fig. 2):

$$\mathbf{Y} = \mathbf{Y}^{\text{k1}} + \mathbf{Y}^{\text{k2}} \tag{11}$$

where $\mathbf{Y}^{\text{k1}}$ and $\mathbf{Y}^{\text{k2}}$ are the convolutional results from two branches, respectively. $\mathbf{Y} \in \text{R}^{5 \times 5 \times N}$ is the fused result.

The global information is embedded by the "global average pooling" module (see Fig. 2) to generate the spectralwise statistics $\mathbf{U} = [u_1, u_2, u_3, \ldots, u_N]^{\text{T}} \in \text{R}^{N \times 1}$ by

$$u_c = \frac{1}{5 \times 5} \sum_{i=1}^{5} \sum_{j=1}^{5} \mathbf{Y}_c(i, j) \tag{12}$$

where u_c is the cth component of the spectralwise statistics $\mathbf{U}$, corresponding to the cth spectral band. $\mathbf{Y}_c$ is the cth component of $\mathbf{Y}$, which is a 5×5 matrix ($\mathbf{Y}_c \in \text{R}^{5 \times 5}$).

The compact feature $\mathbf{Z}$ is computed by the "full connected 1" module (see Fig. 2) according to (13). "ceil $(\cdot)$" is an operator to acquire the rounded-up integer. $\mathbf{Z}$ is a ceil (N/r) vector (i.e.,

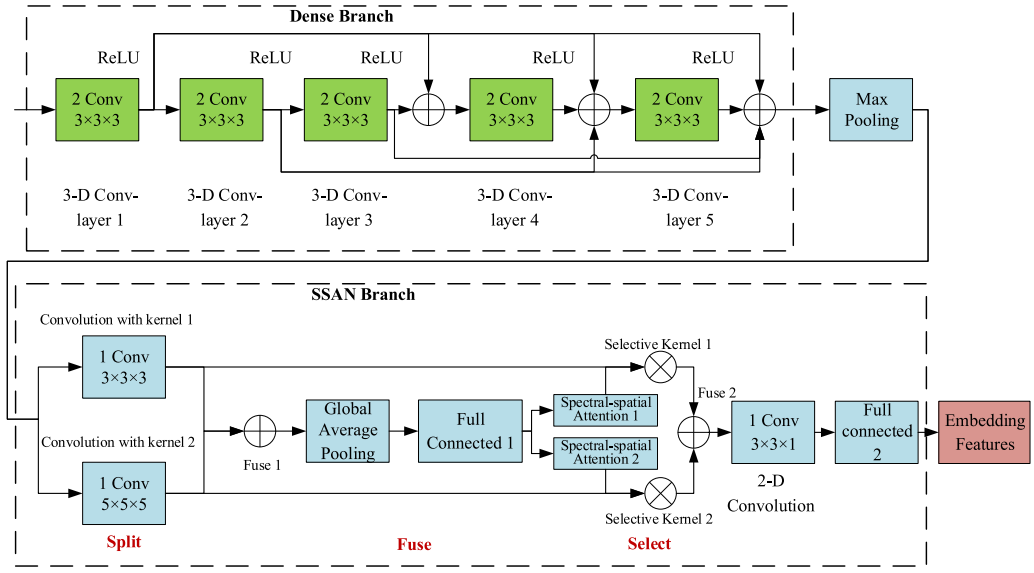


Fig. 2. Architecture of the deep CNN network in this article.

$\mathbf{Z}\in\mathrm{R}^{\mathrm{ceil}\,(N/r)\times 1}$). r is the reduction ratio

$$\mathbf{Z}=\psi(\mathbf{MU}) \tag{13}$$

where ψ is the ReLU function. $\mathbf{M}\in\mathrm{R}^{\mathrm{ceil}\,(N/r)\times N}$ is a parameter matrix needing trained in the network. The purpose of $\mathbf{Z}$ is to guide the adaptive selections.

c) Select: "Spectral–spatial attention 1" and "Spectral–spatial attention 2" are two branches with different receptive scales of spectral–spatial information (see Fig. 2). For the cth spectral band, the weights (a_c, b_c) of the two branches are calculated by

$$a_c=\frac{e^{\mathbf{A}c\mathbf{Z}}}{e^{\mathbf{A}c\mathbf{Z}}+e^{\mathbf{B}c\mathbf{Z}}}$$
$$b_c=\frac{e^{\mathbf{B}c\mathbf{Z}}}{e^{\mathbf{A}c\mathbf{Z}}+e^{\mathbf{B}c\mathbf{Z}}} \tag{14}$$

where $\mathbf{A}c$ and $\mathbf{B}c$ are the cth components of the parameter matrixes $\mathbf{A}$ and $\mathbf{B}$, needing trained in the network. $\mathbf{A}c\in\mathrm{R}^{1\times\mathrm{ceil}\,(N/r)}$, $\mathbf{B}c\in\mathrm{R}^{1\times\mathrm{ceil}\,(N/r)}$, $\mathbf{A}\in\mathrm{R}^{N\times\mathrm{ceil}\,(N/r)}$, and $\mathbf{B}\in\mathrm{R}^{N\times\mathrm{ceil}\,(N/r)}$.

"Fuse 2" is to adaptively select different scales of spectral–spatial information (see Fig. 2). For the cth spectral band, the output ($\mathbf{O}_c$) of "Fuse 2" module is realized by (15). $\mathbf{O}=[\mathbf{O}_1, \mathbf{O}_2, \mathbf{O}_3, \ldots, \mathbf{O}_N]$ is the output of SSAN for all the spectral bands ($\mathbf{O}_c\in\mathrm{R}^{5\times 5}$ and $\mathbf{O}\in\mathrm{R}^{5\times 5\times N}$)

$$\mathbf{O}_c=a_c\cdot\mathbf{Y}_c^{\mathrm{k1}}+b_c\cdot\mathbf{Y}_c^{\mathrm{k2}}. \tag{15}$$

D. NN for Classification

In this article, all the testing samples are transferred to an embedded-feature space, and the embedded features are extracted by the trained deep network. The similarity scores between the last global prototypical representations and the each testing sample are calculated according to (8). The maximum similarity score determines the class type of each testing sample. In fact, (8) is to calculate the opposite number of the Euclidean distance between a global prototypical representation and the embedded features of a sample. Thus, the maximum similarity score corresponds to the minimum Euclidean distance. In other words, the classification of the testing samples is finished based on Euclidean distance using NN classifier.

TABLE II
DETAILS OF THE TRAINING DATASETS

Dataset Name	KSC	Houston	Chikusei	Botswana
Wavelength (nm)	400-2500	380-1050	363-1018	400-2500
Num. of Bands (N)	176	144	128	145
Num. of Classes	13	31	19	14
Width (Pixels)	614	1905	2335	256
Height (Pixels)	512	349	2517	1476

TABLE III
DETAILS OF THE THREE TESTING DATASETS

Dataset Name	Salinas	IP	UP
Wavelength (nm)	400-2500	400-2500	430-860
Num. of Bands (N)	204	200	103
Num. of Classes	16	16	9
Width (Pixels)	217	145	340
Height (Pixels)	512	145	610

III. EXPERIMENTS

A. Experimental Data

1) Training Data: There are four hyperspectral datasets that are used as training data in this article. A brief introduction of the hyperspectral datasets for training the network is shown in Table II [24].

2) Testing Data: There are three hyperspectral datasets that are used as testing data in this article. Both the training datasets and the testing datasets are popular and well known in the academic community of hyperspectral remote sensing. A brief introduction of the testing datasets is shown in Table III [24].

Figs. 3–5 show the ground-truth classifications (i.e., real land cover) of the three testing datasets. Salinas dataset is located

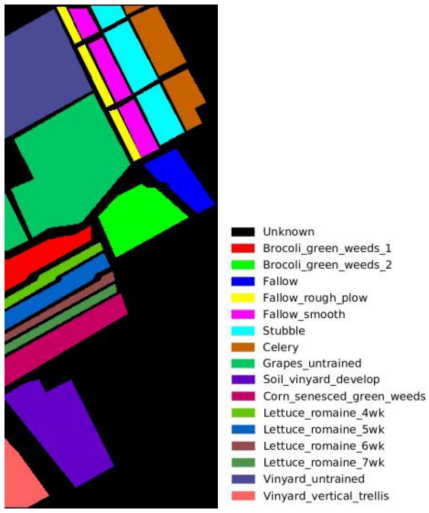


Fig. 3. Real land cover of the Salinas hyperspectral image.

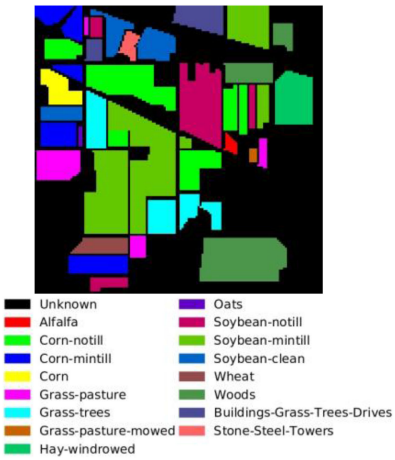


Fig. 4. Real land cover of the IP hyperspectral image.

in California. IP is short for Indian Pines located in Indiana, whereas UP is short for the University of Pavia located in Pavia.

B. Architecture Details of the Proposed Deep CNN

The architecture and parameters of the proposed network are shown in Table IV. The inputting size of GPN is $9 \times 9 \times N\_BAND$. *N_BAND* is the number of spectral bands. Hence, both the spectral curve of a pixel and a neighborhood (9×9) of the pixel (spatial information) are input to the network. The layers described in Table IV can be found in Fig. 2. An experiment of parameter sensitivity was conducted to set the parameter d, with $L = 15$ (the number of supervised samples in each new class) as an example. The result is shown in Fig. 6. d was set to be 150, where the OA reached the highest.

IV. Results and Discussion

A. Accuracy

In this study, for the Salinas and the UP datasets, the experiments were conducted with the number (L) of supervised

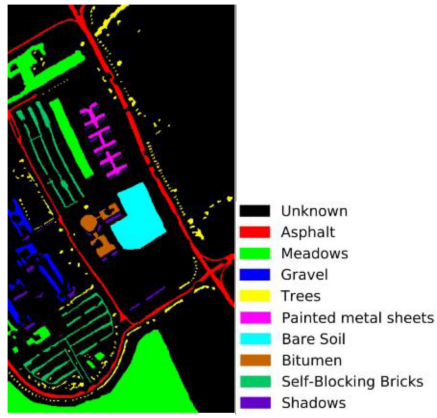


Fig. 5. Real land cover of the UP hyperspectral image.

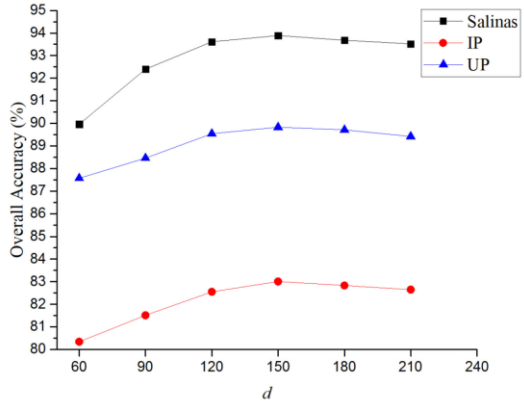


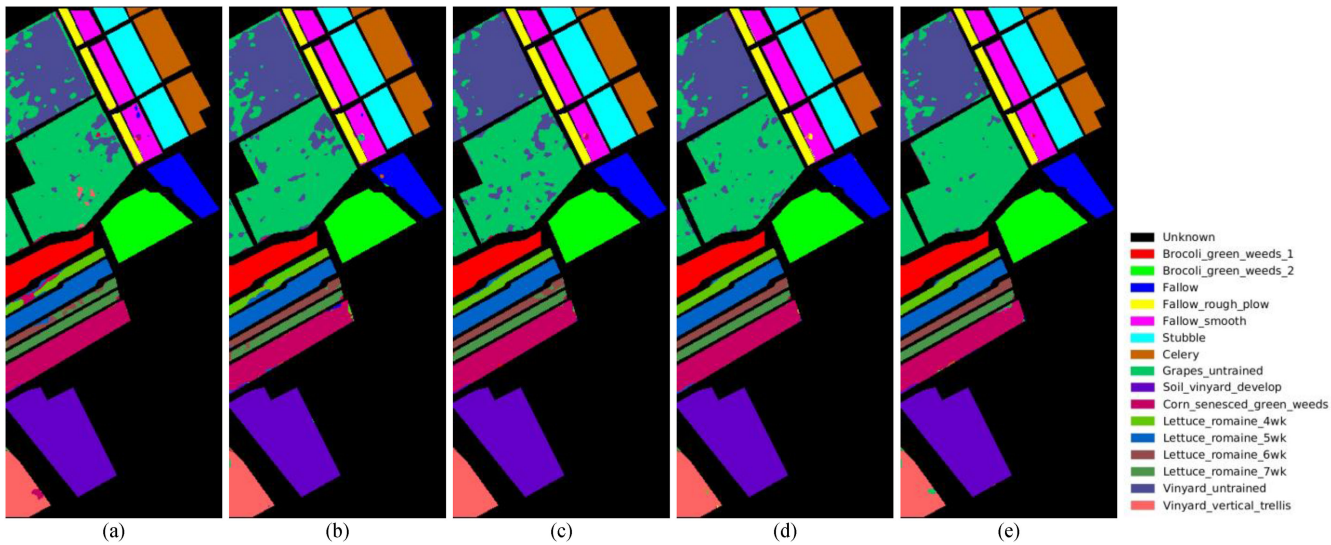
Fig. 6. OA with a variety of d.

samples in each class as 5, 10, 15, 20, and 25, respectively. For the IP dataset, L was set to be 5, 10, and 15, respectively, due to the limited number of labeled samples for a class (i.e., there are only 20 labeled samples in the class "Oats"). The results of classification are shown in Figs. 7–9. For each case of L, ten runs of experiments were performed repeatedly. The averages of the OA are shown in Tables V–VII, which also show the standard deviation (STD). The OA of the our method (GPN) is compared with that of existing popular methods, which is also shown in Tables V–VII, including SVM, DFSL+SVM [24], spatial-contextual semisupervised SVM (SC^3SVM) [41], DFSL+NN [24], MSDN (end-to-end 3-D dense convolutional network) [42], spatial–spectral label propagation based on the SVM (SS-LPSVM) [43], double-branch multiattention (DBMA) [44], Laplacian SVM [45], k-NN + spatial neighborhood information (KNN+SNI) [46], MSDN with spectralwise attention mechanism (MSDN-SA) [42], multinomial logistic regression + random selection (MLR+RS) [47], Transductive SVM [48], 3D-CNN [21], and SVM+Siamese-CNN [24].

In Tables V–VII, the average OA and STD in ten runs of DBMA, MSDN, MSDN-SA, and 3D-CNN were tested by this article, whereas others were from Liu *et al.* [24]. Tables V–VII show that the proposed GPN obtains the better results than other popular methods. Hence, the results in this article suggest that the proposed GPN is state-of-the-art in terms of OA.

TABLE IV
ARCHITECTURE AND PARAMETERS OF THE DEEP CNN

Layer Name	Input Layer	Filter Size	Padding	Output Size
Input layer	/	/	/	9×9×*N_BAND*
3-D Conv-layer 1	Input layer	3×3×3×2	Yes	9×9×*N_BAND*×2
3-D Conv-layer 2	3-D Conv-layer 1	3×3×3×2	Yes	9×9×*N_BAND*×2
3-D Conv-layer 3	3-D Conv-layer 2	3×3×3×2	Yes	9×9×*N_BAND*×2
Short-connection 1	3-D Conv-layer 1&3	3-D Conv-layer 1 + 3-D Conv-layer 3		9×9×*N_BAND*×2
3-D Conv-layer 4	Short-connection 1	3×3×3×2	Yes	9×9×*N_BAND*×2
Short-connection 2	3-D Conv-layer 1&2&4	3-D Conv-layer 1 +3-D Conv-layer 2 + 3-D Conv-layer 4		9×9×*N_BAND*×2
3-D Conv-layer 5	Short-connection 2	3×3×3×2	Yes	9×9×*N_BAND*×2
Short-connection 3	3-D Conv-layer 1&2&3&5	3-D Conv-layer 1 +3-D Conv-layer 2 +3-D Conv-layer 3 + 3-D Conv-layer 5		9×9×*N_BAND*×2
Max Pooling	Short-connection 3	2×2×1	No	5×5×*N_BAND*×1
Convolution with kernel 1	Max Pooling	3×3×3×1	Yes	5×5×*N_BAND*×1
Convolution with kernel 2	Max Pooling	5×5×5×1	Yes	5×5×*N_BAND*×1
Fuse 1	Convolution with kernel 1 & 2	Convolution with kernel 1 + Convolution with kernel 2		5×5×*N_BAND*×1
Global Average Pooling	Fuse 1	5×5×1×1	No	1×1×*N_BAND*×1
Full Connected 1	Global Average Pooling	/	/	1×1×ceil (*N_BAND*/*r*)×1
Spectral-spatial Attention 1	Full Connected 1	/	/	1×1×*N_BAND*×1
Spectral-spatial Attention 2	Full Connected 1	/	/	1×1×*N_BAND*×1
Selective Kernel 1	Convolution with kernel 1 & Spectral-spatial Attention 1	Convolution with kernel 1 ⊗ Spectral-spatial Attention 1		5×5×*N_BAND*×1
Selective Kernel 2	Convolution with kernel 2 & Spectral-spatial Attention 2	Convolution with kernel 2 ⊗ Spectral-spatial Attention 2		5×5×*N_BAND*×1
Fuse 2	Selective Kernel 1 & 2	Selective Kernel 1 + Selective Kernel 2		5×5×*N_BAND*×1
2-D Convolution	Fuse 2	3×3×1×1	No	3×3×*N_BAND*×1
Full Connected 2	2-D Convolution	/	/	*d*×1



(a) (b) (c) (d) (e)

Fig. 7. Classification results of the Salinas dataset. (a)–(e) Number (*L*) of supervised samples is 5, 10, 15, 20, and 25 in each class, respectively.

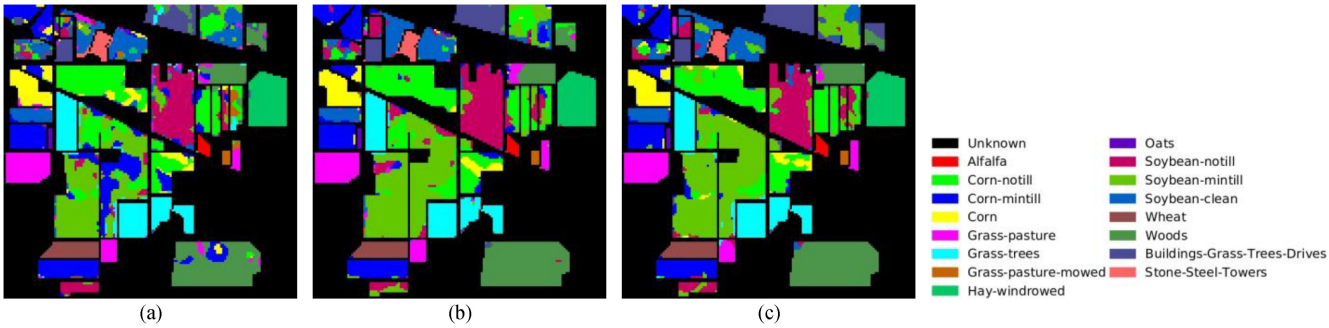


Fig. 8. Classification results of the IP dataset. (a)–(c) Number (L) of supervised samples is 5, 10, and 15 in each class, respectively.

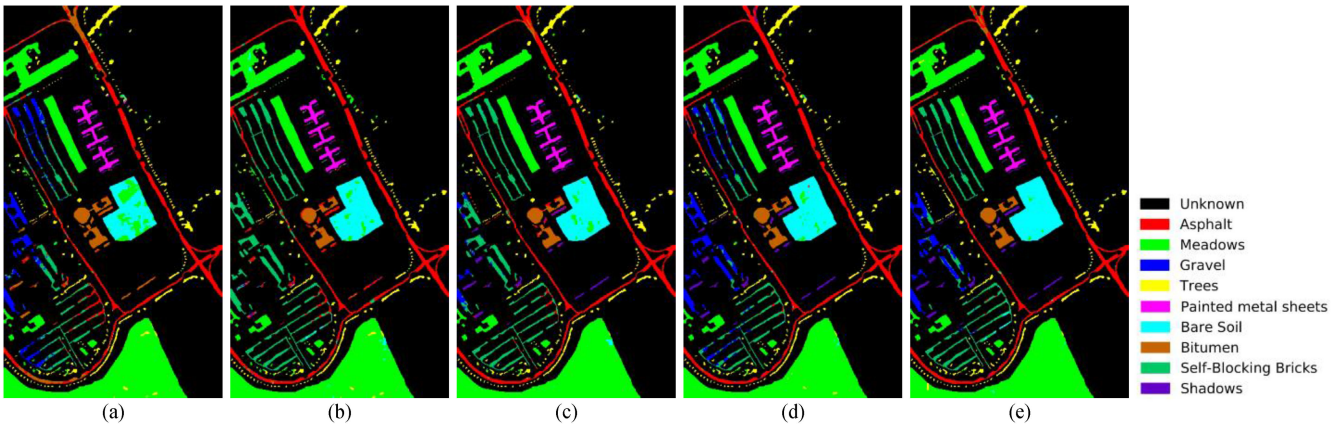


Fig. 9. Classification results of the UP dataset. (a)–(e) Number (L) of supervised samples is 5, 10, 15, 20, and 25 in each class, respectively.

TABLE V
AVERAGE ± STD (OA, %) OF TEN RUNS FOR SALINAS DATASET (THE BOLD RESULT IS THE HIGHEST ACCURACY WITH EACH L)

Number (L) of supervised samples	5	10	15	20	25
Support Vector Machine (SVM)	73.90 ± 1.91	75.62 ± 1.73	79.08 ± 1.45	77.89 ± 1.20	78.05 ± 1.49
DFSL+SVM	85.58 ± 1.87	89.73 ± 1.24	91.21 ± 1.64	93.42 ± 1.25	94.28 ± 0.80
SC^3SVM	74.12 ± 2.44	78.49 ± 2.02	81.83 ± 0.93	81.22 ± 1.27	77.08 ± 0.80
DFSL+NN	88.40 ± 1.54	89.86 ± 1.69	92.15 ± 1.24	92.69 ± 0.98	93.61 ± 0.83
SS-LPSVM	86.79 ± 1.75	90.36 ± 1.35	90.86 ± 1.36	91.77 ± 0.96	92.11 ± 1.07
DBMA	89.76 ± 2.34	90.64 ± 1.99	93.01 ± 1.73	94.12 ± 1.70	94.75 ± 1.47
Laplacian SVM	75.31 ± 2.31	76.34 ± 1.77	77.93 ± 2.42	79.40 ± 0.73	80.56 ± 1.33
KNN+SNI	80.39 ± 1.58	84.64 ± 1.54	86.94 ± 1.52	88.28 ± 1.49	87.64 ± 1.72
MSDN	89.45 ± 2.26	90.51 ± 2.06	92.74 ± 1.82	93.82 ± 1.68	94.37 ± 1.59
MLR+RS	78.29 ± 2.16	85.03 ± 1.43	87.20 ± 1.74	88.76 ± 1.87	89.42 ± 0.85
Transductive SVM	60.43 ± 1.40	67.47 ± 1.05	69.12 ± 1.32	71.03 ± 1.78	71.83 ± 1.16
3D-CNN	85.58 ± 2.18	86.26 ± 1.84	88.01 ± 1.47	88.94 ± 1.38	91.21 ± 1.23
MSDN-SA	89.91 ± 2.15	90.79 ± 1.86	93.18 ± 1.66	94.15 ± 1.59	94.86 ± 1.38
SVM+Siamese-CNN	12.66 ± 2.75	50.04 ± 14.34	60.72 ± 4.85	70.30 ± 2.61	71.62 ± 12.05
GPN	**90.36 ± 1.96**	**91.44 ± 1.74**	**93.92 ± 1.61**	**94.82 ± 1.36**	**95.95 ± 1.25**

B. Ablation Study

The proposed method in this article consists of three main modules, i.e., global representation learning, dense CNN, and SSAN. An ablation study was performed in this article to demonstrate that every module was effective for improving accuracy. In other words, the accuracy of the method was tested when the three modules were replaced, respectively. In the ablation study, the strategy of global representation learning was replaced by a traditional normal triplet learning [49]. The dense branch and SSAN branch were replaced by a normal CNN module, respectively. When a module was replaced, other modules were kept same with the proposed method (GPN). The results of ablation study are shown in Tables VIII–Table X. When the global representation learning, dense CNN, and SSAN were replaced, respectively, the classification accuracy presented decline. In other words, the accuracy of the proposed method is better than that of other cases where the three modules were replaced, respectively. Thus, it is demonstrated that all the three modules

TABLE VI
AVERAGE ± STD (OA, %) OF TEN RUNS FOR IP DATASET (THE BOLD RESULT IS THE HIGHEST ACCURACY WITH EACH L)

Number (L) of supervised samples	5	10	15
Support Vector Machine (SVM)	50.23 ± 1.74	55.56 ± 2.04	58.58 ± 0.80
DFSL+SVM	64.58 ± 2.78	75.53 ± 1.89	79.98 ± 2.23
SC^3SVM	55.42 ± 0.35	60.86 ± 5.08	67.24 ± 0.47
DFSL+NN	67.84 ± 1.29	76.49 ± 1.44	78.62 ± 1.59
SS-LPSVM	56.95 ± 0.95	64.74 ± 0.39	78.76 ± 0.04
DBMA	69.76 ± 2.32	79.42 ± 2.05	82.28 ± 1.84
Laplacian SVM	52.31 ± 0.67	56.36 ± 0.71	59.99 ± 0.65
KNN+SNI	56.39 ± 1.03	74.88 ± 0.54	78.92 ± 0.61
MSDN	69.15 ± 2.24	78.96 ± 2.16	81.79 ± 2.05
MLR+RS	55.38 ± 3.98	69.28 ± 2.63	75.15 ± 1.43
Transductive SVM	62.57 ± 0.23	63.45 ± 0.17	65.42 ± 0.02
3D-CNN	63.54 ± 2.72	71.25 ± 1.64	76.25 ± 2.17
MSDN-SA	69.87 ± 2.13	79.54 ± 2.01	82.36 ± 1.96
SVM+Siamese-CNN	10.02 ± 1.48	17.71 ± 4.90	44.00 ± 5.73
GPN	**70.45 ± 1.84**	**80.01 ± 1.72**	**82.97 ± 1.63**

TABLE VII
AVERAGE ± STD (OA, %) OF TEN RUNS FOR UP DATASET (THE BOLD RESULT IS THE HIGHEST ACCURACY WITH EACH L)

Number (L) of supervised samples	5	10	15	20	25
Support Vector Machine (SVM)	53.73 ± 1.30	61.53 ± 1.14	60.43 ± 0.94	64.89 ± 1.14	68.01 ± 2.62
DFSL+SVM	72.57 ± 3.93	84.56 ± 1.83	87.23 ± 1.38	90.69 ± 1.29	93.08 ± 0.92
SC^3SVM	56.76 ± 2.28	64.25 ± 0.40	66.87 ± 0.37	68.24 ± 1.18	69.45 ± 2.19
DFSL+NN	80.81 ± 3.12	84.79 ± 2.27	86.68 ± 2.61	89.59 ± 1.05	91.11 ± 0.83
SS-LPSVM	69.60 ± 2.30	75.88 ± 0.22	80.67 ± 1.21	78.41 ± 0.26	85.56 ± 0.09
DBMA	80.89 ± 1.77	85.76 ± 1.46	89.07 ± 1.39	92.71 ± 1.61	94.28 ± 1.31
Laplacian SVM	65.72 ± 0.34	68.26 ± 2.20	68.34 ± 0.29	65.91 ± 0.45	68.88 ± 1.34
KNN+SNI	70.21 ± 1.29	78.97 ± 2.33	82.56 ± 0.51	85.18 ± 0.65	86.26 ± 0.37
MSDN	79.59 ± 1.95	84.63 ± 1.64	88.16 ± 1.48	92.89 ± 1.58	93.56 ± 1.37
MLR+RS	69.73 ± 3.15	80.30 ± 2.54	84.10 ± 1.94	83.52 ± 2.13	87.97 ± 1.69
Transductive SVM	63.43 ± 1.22	63.73 ± 0.45	68.45 ± 1.07	73.72 ± 0.27	69.96 ± 1.39
3D-CNN	71.58 ± 3.58	79.63 ± 1.75	83.89 ± 2.93	85.98 ± 1.76	89.56 ± 1.20
MSDN-SA	80.94 ± 1.86	85.84 ± 1.59	89.18 ± 1.36	92.76 ± 1.45	94.35 ± 1.26
SVM+Siamese-CNN	23.68 ± 6.34	66.64 ± 2.37	68.35 ± 4.70	78.43 ± 1.93	72.87 ± 7.36
GPN	**81.31 ± 1.60**	**86.65 ± 1.33**	**89.81 ± 1.16**	**93.48 ± 1.21**	**94.94 ± 1.13**

TABLE VIII
AVERAGE ± STD (OA, %) OF TEN RUNS FOR SALINAS DATASET IN THE ABLATION STUDY

Number (L) of supervised samples	5	10	15	20	25
Experimental Way	The proposed GPN				
Overall Accuracy (%)	90.36 ± 1.96	91.44 ± 1.74	93.92 ± 1.61	94.82 ± 1.36	95.95 ± 1.25
Experimental Way	A traditional normal triplet learning took the place of the global representations learning				
Overall Accuracy (%)	89.23 ± 2.14	90.58 ± 1.96	93.05 ± 1.99	93.97 ± 1.67	94.89 ± 1.43
Experimental Way	A normal convolutional neural network took the place of the dense convolutional neural network				
Overall Accuracy (%)	90.08 ± 2.36	91.01 ± 2.18	93.59 ± 1.94	94.63 ± 1.68	95.46 ± 1.49
Experimental Way	A normal convolutional neural network took the place of the SSAN				
Overall Accuracy (%)	89.86 ± 2.34	90.94 ± 2.28	93.28 ± 2.16	94.17 ± 1.95	95.18 ± 1.67
Experimental Way	The dense convolutional neural network was used alone				
Overall Accuracy (%)	82.92 ± 2.56	84.47 ± 2.35	87.75 ± 2.36	89.87 ± 2.12	91.47 ± 1.88

TABLE IX
AVERAGE ± STD (OA, %) OF TEN RUNS FOR IP DATASET IN THE ABLATION STUDY

Number (L) of supervised samples	5	10	15
Experimental Way	The proposed GPN		
Overall Accuracy (%)	70.45 ± 1.84	80.01 ± 1.72	82.97 ± 1.63
Experimental Way	A traditional normal triplet learning took the place of global representations learning		
Overall Accuracy (%)	69.49 ± 2.45	79.08 ± 2.23	82.06 ± 1.95
Experimental Way	A normal convolutional neural network took the place of dense convolutional neural network		
Overall Accuracy (%)	70.04 ± 2.08	79.82 ± 1.89	82.63 ± 1.74
Experimental Way	A normal convolutional neural network took the place of the SSAN		
Overall Accuracy (%)	69.87 ± 2.24	79.63 ± 2.03	82.44 ± 1.86
Experimental Way	The dense convolutional neural network was used alone		
Overall Accuracy (%)	63.62 ± 2.84	76.11 ± 2.56	79.12 ± 2.19

TABLE X
AVERAGE ± STD (OA, %) OF TEN RUNS FOR UP DATASET IN THE ABLATION STUDY

Number (L) of supervised samples	5	10	15	20	25
Experimental Way	The proposed GPN				
Overall Accuracy (%)	81.31 ± 1.60	86.65 ± 1.33	89.81 ± 1.16	93.48 ± 1.21	94.94 ± 1.13
Experimental Way	A traditional normal triplet learning took the place of global representations learning				
Overall Accuracy (%)	80.46 ± 2.36	85.59 ± 2.15	88.92 ± 1.89	92.57 ± 1.74	94.09 ± 1.62
Experimental Way	A normal convolutional neural network took the place of dense convolutional neural network				
Overall Accuracy (%)	81.02 ± 2.01	86.18 ± 1.87	89.39 ± 1.79	93.03 ± 1.57	94.48 ± 1.43
Experimental Way	A normal convolutional neural network took the place of the SSAN				
Overall Accuracy (%)	80.79 ± 2.13	85.94 ± 1.88	89.07 ± 1.82	92.86 ± 1.73	94.29 ± 1.58
Experimental Way	The dense convolutional neural network was used alone				
Overall Accuracy (%)	74.73 ± 3.46	80.41 ± 3.07	83.91 ± 2.89	88.37 ± 2.17	90.88 ± 2.04

TABLE XI
TIME COMPARISON BETWEEN THE PROPOSED GPN AND SOME EXISTING METHODS

L=5	DBMA+NN	MSDN-SA+NN	3D-CNN+NN	DFSL+NN	DFSL+SVM	GPN
	13.47s+0.37s	12.56s+0.41s	13.59s+0.38s	11.09s+0.34s	11.09s+2.09s	10.03s+0.33s
L=25	DBMA+NN	MSDN-SA+NN	3D-CNN+NN	DFSL+NN	DFSL+SVM	GPN
	13.47s+0.95s	12.56s+0.88s	13.59s+0.84s	11.09s+0.78s	11.09s+123.48s	10.03s+0.77s

make contribution to improving the accuracy of the proposed method. Specifically, when the dense branch was replaced by a normal CNN, the accuracy presented a little decline, whereas the accuracy with replacing global representation learning declined most in the three modules. In other words, the accuracy reached the worst when the global representation learning was replaced, whereas the accuracy with replacing dense CNN was better than replacing SSAN and global representation learning. Hence, it can be inferred that the strategy of global representation learning makes the most contribution to improving the accuracy, whereas the SSAN and dense CNN take the second and third places, respectively. But it should be emphasized that all the three modules are effective for improving classification accuracy. In particular, the experiments that directly applied dense CNN and SSRN were conducted without global representation learning, and the results of the declining accuracy in Tables VIII–X where global representation learning was replaced have demonstrated the effectiveness of the global representation learning. It is worth noting that, in the ablation study, the proposed method is not compared to existing methods, and is compared to the method with one module being replaced by a normal module. For example, when the dense CNN was replaced by a normal CNN, the method for comparison still used the global representation learning and SSAN, so the difference in accuracy was just because of the absence of dense CNN, which was less obvious than the comparison to existing methods (e.g., SVM and SC^{3}SVM).

In addition, the dense CNN was used alone for supervised classification. The results are shown in Tables VIII–X. The accuracy with using the dense CNN alone is lower than that with using two or three modules simultaneously, which confirms the effectiveness of the proposed GPN.

C. Time Consumption

The time consumption of the proposed method was tested and compared with some existing methods using the IP dataset. The details of computer environment for time testing were same for different methods in this article. The results of time consumption for different methods are shown in Table XI. It suggests that the time consumption of GPN is similar with that of several popular methods.

V. CONCLUSION

This study proposed a GPN for hyperspectral image classification using limited supervised samples (i.e., few-shot hyperspectral image classification). The experiments were conducted for verification, and some main conclusions were reached as follows.

1) The accuracy of GPN shows better than existing popular methods under the condition of small samples. The comparative analysis suggests that the GPN is state-of-the-art for solving the classification problem of hyperspectral image using limited supervised samples.
2) The ablation study demonstrates that all the three modules (global representation learning, dense branch, and SSAN branch) are effective for improving the accuracy, whereas the strategy of global representation learning makes the most contribution.
3) The time expenditure of the proposed method (GPN) and several existing popular methods is similar in the same operational environment.

In the follow-up study, more hyperspectral datasets should be used to test the effectiveness of the proposed GPN for few-shot hyperspectral image classification.

ACKNOWLEDGMENT

The authors would like to thank the National Center for Airborne Laser Mapping for providing the “Houston” dataset, the Space Application Laboratory, Department of Advanced Interdisciplinary Studies, University of Tokyo for providing the “Chikusei” dataset, and Grupo de Inteligencia Computacional for providing other datasets.

REFERENCES

[1] H. Su, B. Zhao, Q. Du, P. Du, and Z. Xue, "Multi-feature dictionary learning for collaborative representation classification of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2467–2484, Apr. 2018.

[2] H. Su, B. Yong, and Q. Du, "Hyperspectral band selection using improved firefly algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 68–72, Jan. 2016.

[3] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[4] L. Gao *et al.*, "Subspace-based support vector machines for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 349–353, Feb. 2015.

[5] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Neural network approaches versus statistical methods in classification of multisource remote sensing data," in *Proc. 12th Can. Symp. Remote Sens. Geosci. Remote Sens. Symp.*, Jul. 1989, vol. 2, pp. 489–492.

[6] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognit. Lett.*, vol. 27, no. 4, pp. 294–300, 2006.

[7] J. Ham, Y. C. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006, doi: 10.1126/science.1127647.

[9] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[10] P. Du *et al.*, "Advances of four machine learning methods for spatial data handling: A review," *J. Geovis. Spatial Anal.*, vol. 4, 2020, Art. no. 13. [Online]. Available: https://doi.org/10.1007/s41651-020-00048-5

[11] H. Su, Y. Yu, Q. Du, and P. Du, "Ensemble learning for hyperspectral image classification using tangent collaborative representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3778–3790, Jun. 2020.

[12] H. Su, B. Zhao, Q. Du, and P. Du, "Kernel collaborative representation with local correlation features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1230–1241, Feb. 2019.

[13] W. Zhao, Z. Guo, J. Yue, X. Zhang, and L. Luo, "On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery," *Int. J. Remote Sens.*, vol. 36, no. 13, pp. 3368–3379, 2015, doi: 10.1080/2150704X.2015.1062157.

[14] J. Yue, S. Mao, and M. Li, "A deep learning framework for hyperspectral image classification using spatial pyramid pooling," *Remote Sens. Lett.*, vol. 7, no. 9, pp. 875–884, 2016, doi: 10.1080/2150704X.2016.1193793.

[15] V. Singhal and A. Majumdar, "Row-sparse discriminative deep dictionary learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 5019–5028, Dec. 2018.

[16] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, no. A, pp. 120–147, 2018, doi: 10.1016/j.isprsjprs.2017.11.021.

[17] W. Zhao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[18] P. Ghamisi *et al.*, "New frontiers in spectral-spatial hyperspectral image classification the latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.

[19] J. Li, B. Xi, Q. Du, R. Song, Y. Li, and G. Ren, "Deep kernel extreme-learning machine for the spectral-spatial classification of hyperspectral imagery," *Remote Sens.*, vol. 10, no. 12, 2018, Art. no. 2036, doi: 10.3390/rs10122036.

[20] A. Song, J. Choi, Y. Han, and Y. Kim, "Change detection in hyperspectral images using recurrent 3D fully convolutional networks," *Remote Sens.*, vol. 10, no. 11, 2018, Art. no. 1827, doi: 10.3390/rs10111827.

[21] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.

[22] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral-spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, Oct. 2018.

[23] H. Shen, M. Jiang, J. Li, Q. Yuan, Y. Wei, and L. Zhang, "Spatial-spectral fusion by combining deep learning and variational model," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 6169–6181, Aug. 2019.

[24] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019.

[25] S. Xu, J. Li, M. Khodadadzadeh, A. Marinoni, P. Gamba, and B. Li, "Abundance-indicated subspace for hyperspectral classification with limited training samples," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1265–1278, Apr. 2019.

[26] J. Choe, S. Park, K. Kim, J. H. Park, D. Kim, and H. Shim, "Face generation for low-shot learning using generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 1940–1948.

[27] X. Dong, L. Zhu, D. Zhang, Y. Yang, and F. Wu, "Fast parameter adaptation for few-shot image captioning and visual question answering," in *Proc. ACM Multimedia Conf.*, Seoul, South Korea, 2018, pp. 54–62.

[28] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, London, U.K., Sep. 2017. Accessed: Feb. 22, 2020. [Online]. Available: https://kopernio.com/viewer?doi = arXiv:1709.03410v1&route = 6

[29] E. Schwartz *et al.*, "Delta-encoder: An effective sample synthesis method for few-shot object recognition," in *Proc. Neural Inf. Process. Syst.*, Montréal, QC, Canada, Dec. 2018, pp. 2850–2860.

[30] Y. Wang *et al.*, "Generalizing from a few examples: A survey on few-shot learning," 2019. Accessed: Feb. 1, 2020. [Online]. Available: https://arxiv.org/pdf/1904.05046.pdf

[31] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[32] R. Kemker and C. Kanan, "Self-taught feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2693–2705, May 2017.

[33] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.

[34] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839–848, Sep. 2017.

[35] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1909–1921, Apr. 2018.

[36] H. Tang, Y. Li, X. Han, Q. Huang, and W. Xie, "A Spatial–spectral prototypical network for hyperspectral remote sensing image," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 1, pp. 167–171, Jan. 2020.

[37] A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, "Few-shot learning with global class representations," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 9715–9724.

[38] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[39] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," 2019. Accessed: Dec. 22, 2019. [Online]. Available: https://arxiv.org/pdf/1903.06586.pdf

[40] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 636–644.

[41] B. Kuo, C. Huang, C. Hung, Y. Liu, and I. Chen, "Spatial information based support vector machine for hyperspectral image classification," in *Proc. IEEE Int. Symp. Geosci. Remote Sens. Symp.*, Honolulu, HI, USA, Jul. 2010, pp. 832–835.

[42] B. Fang, Y. Li, H. Zhang, and J. C. Chan, "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sens.*, vol. 11, no. 2, 2019, Art. no. 159, doi: 10.3390/rs11020159.

[43] L. Wang, S. Hao, Q. Wang, and Y. Wang, "Semi-supervised classification for hyperspectral imagery based on spatial-spectral label propagation," *ISPRS J. Photogramm. Remote Sens.*, vol. 97, pp. 123–137, 2014, doi: 10.1016/j.isprsjprs.2014.08.016.

[44] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1307, doi: 10.3390/rs11111307.

[45] M. Belkin, P. Niyogi and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.
[46] K. Tan, J. Hu, J. Li, and P. Du, "A novel semi-supervised hyperspectral image classification approach based on spatial neighborhood information and classifier combination," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 19–29, 2015, doi: 10.1016/j.isprsjprs.2015.03.006.
[47] I. Dopido, J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas Dias, and J. A. Benediktsson, "Semisupervised self-learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 71, pp. 4032–4044, Jul. 2013.
[48] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th Int. Conf. Mach. Learn.*, Bled, Slovenia, Jun. 1999, pp. 200–209.
[49] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Lecture Notes in Computer Science*, A. Feragen *et al.*, Eds. Berlin, Germany: Springer, 2015, pp. 84–92.

Chengye Zhang received the B.Eng. degree in remote sensing from Beihang University, Beijing, China, in 2013, and the Ph.D. degree in GIS from Peking University, Beijing, China, in 2018.

Since 2018, he has been an Assistant Professor with the China University of Mining and Technology, Beijing, China. He has been conducting research in the area of hyperspectral remote sensing.

Dr. Zhang is a Reviewer of the IEEE J-STARS, *ISPRS Journal of Photogrammetry and Remote Sensing*, *International Journal of Remote Sensing*, *Remote Sensing Letters*, and *Journal of Applied Remote Sensing*.

Jun Yue received the B.Eng. degree in geodesy from Wuhan University, Wuhan, China, in 2013, and the Ph.D. degree in GIS from Peking University, Beijing, China, in 2018.

He is currently an Assistant Professor with the Changsha University of Science and Technology, Changsha, China. His current research interests include remote sensing, image processing, and object detection. In particular, his interests include few-shot learning, reinforcement learning, neural architecture search, multimodal learning, and hyperspectral image understanding.

Dr. Yue is a Reviewer of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *ISPRS Journal of Photogrammetry and Remote Sensing*, *IEEE Geoscience and Remote Sensing Letters*, *Remote Sensing*, *Information Sciences*, *International Journal of Remote Sensing*, *Remote Sensing Letters*, IEEE ACCESS, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, *Plos One*, *Journal of Supercomputing*, *Infrared Physics and Technology*, *Journal of Supercomputing*, *Current Medical Imaging Reviews*, *Energy Conversion and Management*, *Transactions of the ASABE,* and *Acta Oceanologica Sinica*.

Qiming Qin received the B.S. degree in geography from Nanjing Normal University, Nanjing, China, in 1982, the M.S. degree from Shaanxi Normal University, Xi'an, China, in 1987, and the Ph.D. degree from Peking University, Beijing, China, in 1990.

He is currently a Professor with the School of Earth and Space Sciences, Peking University, and conducting research in remote sensing and GIS. He has authored or coauthored more than 100 peer-reviewed articles.