Hybrid Tensor Networks for Fully Supervised and Semisupervised Hyperspectral Image Classification

Yahui Xiu, Fuyin Ye, Zhao Chen, and Yuxuan Liu

Abstract—**Hyperspectral image (HSI) is rich in spectral information and spatial information to explore the physical and chemical properties of the objects, but it also brings many difficulties to the classification task. The problems of the curse of dimensionality and spectral variability in HSIs can affect the efficiency of the classifier and cause the decline of the classification accuracy. Also, the scarcity of manually labeled samples makes it difficult for the fully supervised classifiers to obtain the best results. To this end, this article proposes several novel hybrid tensor networks (HTNs) that represent multiscale spectral–spatial patterns and low-rank features for accurate classification. Moreover, the fully supervised HTN (FHTN) is embedded within a semisupervised framework with unsupervised modules providing pseudolabels, thus creating semisupervised HTN (SHTN) to exploit unlabeled data and reduce dependence on manual annotations. With proper postprocessing techniques, misclassifications are largely reduced and accuracy is further increased. The experimental results show that the proposed HTNs exhibit good generalization and robustness. FHTN and SHTN outperform classic and advanced supervised and semisupervised models in ground object classification for HSIs.**

Index Terms—**Fully supervised classification, hybrid tensor network (HTN), hyperspectral image (HSI), low-rank features, semisupervised classification, spectral–spatial features.**

I. Introduction

Hyperspectral images (HSIs) are digital images obtained by airborne or spaceborne hyperspectral imagers [1]. Containing rich spectral information, HSIs can reflect the spectral properties of the photographed substances. Therefore, HSIs can be used to identify ground objects, which play an important role in varied fields, such as geological exploration, crop detection, and urban management [1], [2].

In traditional pattern recognition [2], [3], [4], HSI classification can be divided into two major steps: feature extraction and class decision. In the era of deep learning (DL), feature extraction and decision making are often realized by one network instead of separate modules [5], [6], [7], [8]. As HSIs are three-dimensional (3-D) data cubes with two spatial dimensions and one spectral dimension, there are spectral, spatial features, and spectral–spatial features to be exploited. Not only recognizing important patterns, feature extraction also reduces data dimension to eliminate noise and improve efficiency. To deal with the complex spectral characteristics of HSIs, the methods, such as manifold learning [9], kernel function transformation [10], and neural networks (NNs) [11], are proposed. However, spectral signals alone are not enough for classification since they can be compromised by limited imaging resolution and distorted by noise and interference. Thus, spatial features are jointly extracted with spectral features to compensate for information loss and exploit local patterns. The spectral–spatial methods mostly use feature stacking [12], kernel mapping [13], [14], filtering [15], [16], [17], tensor analysis [18], [19], and DL [20], [21], [22], [23], [24], which often come in complicated structures with large computation costs.

For class decision, there are unsupervised, fully supervised, and semisupervised classifiers. Typical unsupervised methods, such as K-means [25], make decisions solely on data properties without any human annotations. Meanwhile, their outputs may not be comprehensible and needs further processing. The fully supervised methods, e.g., support vector machine (SVM) [26] and deep belief network [27], can achieve high accuracy but often needs lots of labeled samples for training. The semisupervised methods, such as active learning [28], [29], [30], [31], cotraining [32], [33], [34], and domain adaption networks [35], [36], allow labeled and unlabeled samples to jointly determine the classifier parameters, reducing the dependence on manual annotations. However, if the features of unlabeled data not properly characterized or learned, they may compromise the optimization and reduce classification accuracy.

Moreover, there are still problems in HSI classification that are not well addressed. First, HSIs are high-dimensional data that contain a large amount of redundant information, which can lead to the curse of dimensionality. Second, due to the tradeoff between spatial resolution and spectral resolution in hyperspectral imaging instruments, there is a phenomenon of spectral variability, which can result in class confusion and reduce classification accuracy. Third, calibrating remote sensing data often requires on-site surveys, which are laborious and time-consuming. Thus, it is difficult to manually annotate a large amount of data.

To tackle these problems, this article proposes several novel hybrid tensor networks (HTNs) that represent multiscale spectral–spatial patterns and low-rank features for accurate classification. Assembling the vanilla HTNs as feature

Manuscript received 15 June 2023; revised 4 August 2023; accepted 19 August 2023. Date of publication 25 August 2023; date of current version 7 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61702094, in part by the Dawn Project of Shanghai Municipal Education Commission under Grant 18CG38, and in part by the Young Scientists' Sailing Project of Science and Technology Commission of Shanghai Municipal under Grant 17YF1427400. *(Corresponding author: Zhao Chen.)*

The authors are with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China (e-mail: yahuixiu@163.com; yefuyin1@126.com; chenzhao@dhu.edu.cn; liuyuxuan7388@163.com).

Digital Object Identifier 10.1109/JSTARS.2023.3308723

representation modules, unsupervised clustering modules, and supervised decision layers, semisupervised HTNs (SHTNs) are also proposed. The motivations of this work are as follows. To overcome the curse of dimensionality, it is natural to represent HSIs as tensors and apply tensor decomposition to extract low-rank features and reduce redundant data. Instead of using tensor computations alone, HTNs adopt new architectures that integrate tensor decomposition into convolutional networks based on the mathematical connections between tensor algebra and convolution/pooling/fully connections [37]. Unlike the common networks based on convolutional blocks [38], [39], [40] or the preliminary tensor networks that combine tensor analysis and convolution in simple ways [37], HTNs incorporate tensor decomposition deeply into NNs on different levels and in different dimensions, effectively extracting multidomain low-rank structures and multiscale spectral–spatial patterns. Specifically, local correlation features [41], [42], [43] that reflect spatial continuity in distributions of ground objects are exploited to deal with intraclass heterogeneity and interclass homogeneity. To address the scarcity of manual labels, HTNs can be adapted to semisupervised applications, whereas both labeled and unlabeled data are involved in training and variability of the data is exploited. The contributions are summarized as follows.

1) Combining tensor decomposition with convolutional neural networks (CNNs), the novel HTNs allow for the expression of multiscale spectral–spatial features and multidomain low-rank features, resulting in excellent performances in fully supervised classification for HSIs.
2) In semisupervision, as cluster labels of the unlabeled pixels are used to train the other parts of SHTNs, the feature representation is guided by the discriminative information inherently lying in the data, yielding robust spectral–spatial features that reflect the variability of the unlabeled pixels and enabling accurate classification with minimal manual annotation.
3) High-order HTNs are designed and applied with postprocessing to improve classification performance. In these techniques, tensor decomposition is combined with 3-D convolution to gain feature representation ability and reduce misclassifications.

Experimental results demonstrate that the proposed methods can classify HSIs accurately. The fully supervised HTN achieves higher classification accuracy than fully supervised CNNs. The semisupervised HTNs outperform state-of-the-art semisupervised classification methods. The rest of this article is organized as follows. Section II briefly reviews the related works. Section III introduces the proposed models in full details. The experimental results are presented, analyzed, and discussed in Section IV. The discussion is presented in Section V. Finally, Section VI concludes this article.

II. Related Works

HSIs contain hundreds of continuous bands with rich spectral features, enabling accurate classification [44]. HSI classification mainly includes feature extraction and class decision. We need to consider how to extract discriminative features from HSIs and apply an efficient class decision method to give each pixel of HSIs an appropriate class label.

A. Feature Extraction

HSIs have high dimensional with redundant information. Feature extraction not only recognizes important patterns but also reduces data dimension to eliminate noise and improve efficiency. HSIs feature extraction methods can be divided into spectral feature extraction, spatial feature extraction, and spectral–spatial feature extraction.

1) Spectral Feature Extraction: Each pixel is regarded as an independent feature vector without considering the spatial information of the pixel and its adjacent pixels. For example, principal component analysis [3] uses the high correlation between adjacent bands of HSIs to extract the main feature components for dimension reduction. As the NN thrives, such as 1-D generative adversarial network and 1-D recurrent NN [11] have been applied to extract spectral dimension features of HSIs. Later, Li et al. [38] convoluted the spectral dimension of HSIs through 1-D convolution NN to realize HSI classification by extracting pixel pair features. However, extracting spectral features alone is not sufficient for HSI classification, as they can be comprised by limited imaging resolution and distorted by noise. Also, it cannot deal with the spectral variability.

2) Spatial Feature Extraction: Utilizing the local correlation of ground objects can address the spectral variability. However, due to the tradeoff between spatial resolution and spectral resolution, most HSIs exhibit low spatial resolution despite having high spectral resolution. Moreover, since the targets might be blocked out by other objects, the shapes, contours, and other spatial characteristics of the target objects are not visible. Therefore, it is almost impossible to determine ground objects by extracting spatial information alone.

3) Spectral–Spatial Feature Extraction: Extracting both spatial and spectral features simultaneously can leverage local correlations and further improve the classification accuracy. Dong et al. [39] proposed to extract spatial information through gray-level co-occurrence matrix and then fuse spatial information and spectral information through band superposition to obtain the final spectral–spatial features. The features formed by stacking may have high dimensions, leading to the curse of dimensionality. By tensor analysis, some methods, such as low-rank tensor estimation algorithm [18], linear tensor discriminant analysis [19], and tensor decomposition network [45], can simultaneously extract the spectral–spatial features. Based on DL, CNN is often used to express spectral—spatial features of HSIs. Gao et al. [40] proposed an HSI feature extraction method based on a new multiscale residual network and introduced the mixed depthwise convolution. These spectral–spatial methods often involve multiple different algorithms, leading to the problems, such as strong parameter dependencies and high computational complexity.

B. Class Decision

According to whether the labeled samples are used in the training process, the classifiers can be divided into unsupervised, fully supervised, and semisupervised. With the development of DL technology, there are many new supervised modes, such as self-supervised [44], [46], [47]. As unsupervised and self-supervised networks are not the focus of this article, these works are not discussed further.

1) Fully Supervised: Fully supervised learning requires labeled samples for classification. Due to the data redundancy of HSIs, some classic methods, such as the K-nearest neighbor algorithm [48] and Gaussian process classifier [49], are difficult to obtain good classification results. In recent years, NN models, such as stacked autoencoder [32], [50], [51] and CNN [52], [53], [54], [55], [56], have been applied to the HSI classification. To further improve the classification accuracy, postprocessing methods are also widely used. For example, Zhong et al. [6] combine conditional random field (CRF) [7] with SVM forming support vector CRFs classifier [7] methods to realize high-accuracy full supervised classification. Then, Zhong et al. [6] propose combine CRF with CNN forming CNN-CRF. Spectral–spatial features are extracted by CNN, and spatial context information is extracted by CRF. However, the supervised algorithms need lots of labeled samples for training, and the lack of manual annotations limits their ability to achieve high accuracy.

2) Semisupervised: Semisupervised learning allows labeled and unlabeled samples to jointly determine the classifier parameters, and it can achieve better classification results with limited manual annotations than fully supervised leaning. Wei et al. [8] use multiple branch networks to perform clustering and classification, embed key unsupervised information within the supervised learning process through sharing modules, and improve the generalization ability of the model. Sellars et al. [56] use the superpixel method to define local regions in HSIs, and then extract spatial–spectral features from the regions. But if the similarity between the labeled and the unlabeled samples is low, they may compromise the optimization and reduce classification accuracy.

From above, it can be seen that HSI classification encounters challenges, such as spectral variability, curse of dimensionality, and scarcity of manual annotations. We propose novel HTNs to address these problems.

III. Method

We propose several novel HTNs that represent multiscale spectral–spatial features and low-rank features for accurate classification. Moreover, we incorporate the fully supervised framework into semisupervised framework that includes unsupervised modules providing pseudolabels to reduce dependence on manual annotations and enhance classification accuracy. In order to better introduce the basic concepts and operations of tensor algebra involved in HTNs, we characterize the connected structures of HTNs by graphics, as illustrated in Fig. 1. The HSIs are denoted by tensors $\underline{\mathbf{X}} \in \mathbb{R}^{I_1\times I_2\times I_3}$, where I_1 and I_2 represent the rows and columns. I_3 represents the number of bands. The neighborhood of each pixel is represented by the subtensor $\underline{\mathbf{X}}^{(0)} \in \mathbb{R}^{I_1^{(0)}\times I_2^{(0)}\times I_3^{(0)}}$ as the input of the networks. Also, the predicted label and the class label of the networks are denoted as $\hat{\mathbf{y}} \in \mathbb{R}^C$ and $\mathbf{y} \in \mathbb{R}^C$, respectively, where C is the number of categories.

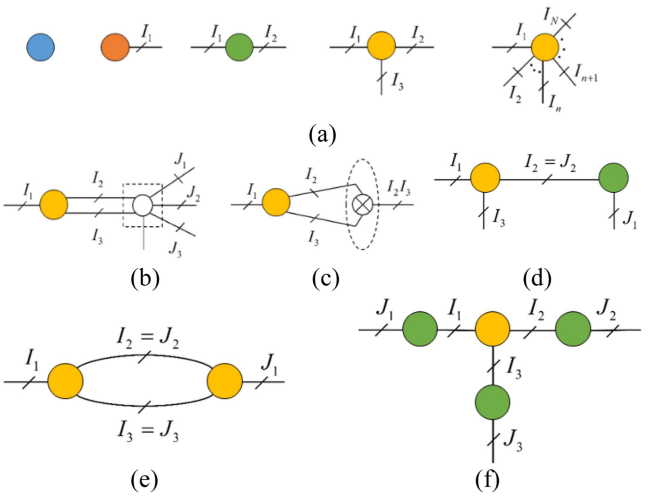


Fig. 1. Graphical representation of tensor operations for tensor network. Each colored node denotes a tensor and each outgoing line indicates a dimension of the tensor. (a) Basic structure of tensors, which are scalars, the first-order tensors, the second-order tensors, and tensors with orders higher than 2 from left to right. (b) Subtensor generation, where the white node with the dashed box denotes that $J_1 J_2$ subtensors $\underline{\mathbf{Q}}^{(j_1,j_2)} \in \mathbb{R}^{K_1\times K_2\times I_3}$ are generated from the red tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and stacked into tensor $\underline{\mathbf{B}} \in \mathbb{R}^{J_1\times J_2\times J_3\times I_3}$ as $J_1 = I_1 - K_1 + 1$, $J_2 = I_2 - K_2 + 1$, and $J_3 = K_1 K_2$. (c) Matricization of tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1\times I_2\times I_3}$. (d) Multiplication of $\underline{\mathbf{X}} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and $\underline{\mathbf{Y}} \in \mathbb{R}^{J_1\times J_2}$. (e) Tensor contraction of $\underline{\mathbf{X}} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and $\underline{\mathbf{Y}} \in \mathbb{R}^{J_1\times J_2\times J_3}$. (f) TD of tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1\times I_2\times I_3}$, where the red node stands for core tensor, and the green ones represent the projection matrices.

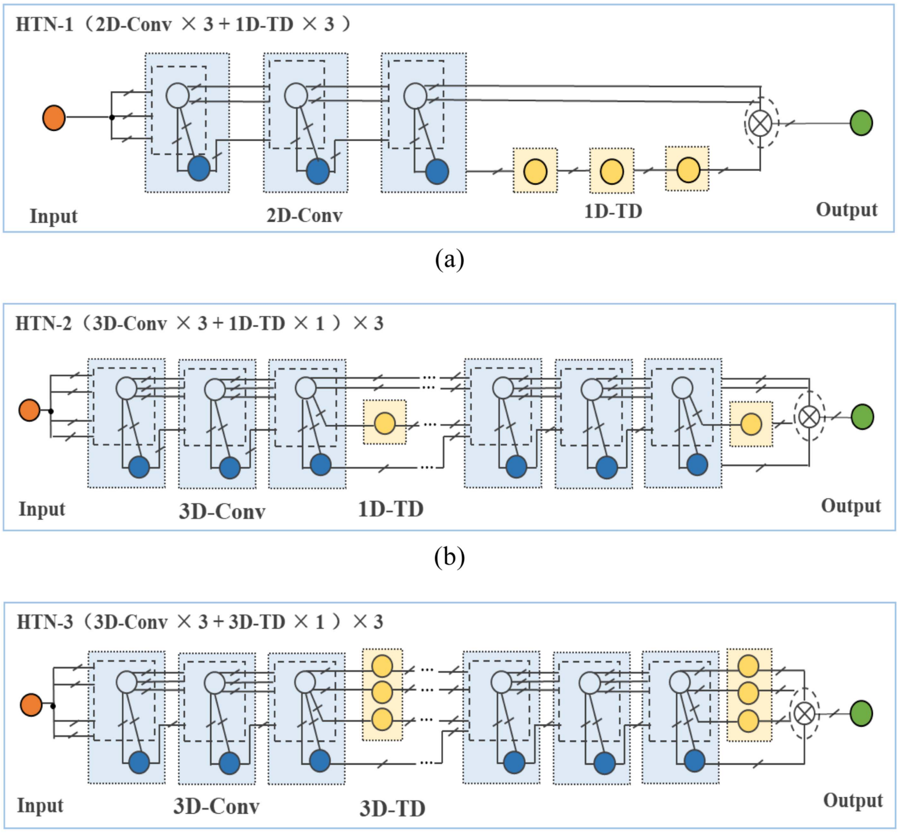
A. Hybrid Tensor Network

Existing studies [57], [58] show that convolution/pooling/fully connections and tensor matrix multiplication can be expressed in the form of tensor contraction. Therefore, Tucker decomposition (TD) can be mixed with convolution/pooling/fully connections to build HTN. High-order convolution and TD can deal with the HSIs in arbitrary dimensions. HTN not only learns the multiscale spectral–spatial features based on CNN but also can extract the low-rank features by combining the TD module. At the same time, it can compress the amount of data and reduce the computational complexity of the model.

Suppose that there are L layers in the network and the layer number is l $(l = 1, 2, \ldots, L)$. If the lth is a convolutional layer and $\underline{\mathbf{X}}^{(l-1)} \in \mathbb{R}^{I_1^{(l-1)}\times I_2^{(l-1)}\times I_3^{(l-1)}}$ is the input, the output is computed by the tensor contraction as follows:

$$\underline{\mathbf{Y}}^{(l)} = \sigma\left(\underline{\mathbf{S}}^{(l)} \times_{3,4}^{1,2} \underline{\mathbf{W}}^{(l)}\right) \qquad (1)$$

where $\underline{\mathbf{Y}}^{(l)} \in \mathbb{R}^{I_1^{(l)}\times I_2^{(l)}\times I_3^{(l)}}$ is the output feature, $\underline{\mathbf{S}}^{(l)} \in \mathbb{R}^{I_1^{(l)}\times I_2^{(l)}\times K_1^{(l)} K_2^{(l)}\times I_3^{(l-1)}}$ stores all the spatially overlapping subtensors generated from $\underline{\mathbf{X}}^{(l-1)}$, $\underline{\mathbf{W}}^{(l)} \in \mathbb{R}^{K_1^{(l)} K_2^{(l)}\times I_3^{(l-1)}\times I_3^{(l)}}$ is the kernel weights, $K_1^{(l)}$ and $K_2^{(l)}$ are the width and height of the convolution filter, and $K_3^{(l)}$ is the number


HTN-1（2D-Conv × 3 + 1D-TD × 3 ）
Input
2D-Conv
1D-TD
Output
HTN-2（3D-Conv × 3 + 1D-TD × 1 ）× 3
Input
3D-Conv
1D-TD
Output
HTN-3（3D-Conv × 3 + 3D-TD × 1 ）× 3
Input
3D-Conv
3D-TD
Output

(a)

(b)

(c)

Fig. 2. Architectures of the proposed models. (a) HTN-1. (b) HTN-2. (c) HTN-3.

of feature maps. $\sigma(\cdot)$ is the nonlinear activation function. If stride is $S_n^{(l)} \in \mathbb{N}^*$, padding is $P_n^{(l)} \in \mathbb{Z}$, then $I_n^{(l)} = (I_n^{(l-1)} - K_n^{(l)} + 2P_n^{(l)})/S_n^{(l)} + 1$.

If the lth is the TD layer based on the Tucker model, as illustrated in Fig. 1(f), the output of this layer can be formulated in tensor matricization as follows:

$$\underline{\mathbf{Y}}^{(l)}_{\text{low-rank}} = \sigma\left(\underline{\mathbf{X}}^{(l)} \times_1^1 \mathbf{W}^{(l,1)} \times_1^1 \mathbf{W}^{(l,2)} \times_1^1 \mathbf{W}^{(l,3)}\right) \tag{2}$$

where $\underline{\mathbf{Y}}^{(l)}_{\text{low-rank}} \in \mathbb{R}^{I_1^{(l)} \times I_2^{(l)} \times I_3^{(l)}}$ is the low-rank feature, $\underline{\mathbf{X}}^{(l)} \in \mathbb{R}^{I_1^{(l-1)} \times I_2^{(l-1)} \times I_3^{(l-1)}}$ is the input, and $\mathbf{W}^{(l,n)} \in \mathbb{R}^{I_n^{(l-1)} \times I_n^{(l)}}$ are the three factor matrices. After training, the TD layer can map the input samples to the optimal low-rank subspace and extract multidimensional joint features.

Suppose that the lth is the full connection layer. Given $\mathbf{x}^{(l)} \in \mathbb{R}^{K^{(l-1)}}$ as input, the output of the fully connected layer can also be formulated in tensor contraction as follows:

$$\mathbf{y}^{(l)} = \sigma\left(\mathbf{x}^{(l)} \times_1^1 \mathbf{W}^{(l)}\right) \tag{3}$$

where $\mathbf{y}^{(l)} \in \mathbb{R}^{K^{(l)}}$ is the output, and $\mathbf{W}^{(l)} \in \mathbb{R}^{K^{(l-1)} \times K^{(l)}}$ is the weight matrix. $\sigma(\cdot)$ is the activation function used in (1).

As the classification network, HTN makes a decision on the final output $\mathbf{y}^{(L)}$ through softmax to obtain the estimated label of training sample

$$\hat{\mathbf{y}} = \frac{e^{\mathbf{y}^{(L)}}}{\mathbf{1}^T e^{\mathbf{y}^{(L)}}} \tag{4}$$

where $\mathbf{y}^{(L)}$ is the output of the Lth layer, and $\mathbf{1} \in \mathbb{R}^M$ is an all-one vector.

1) HTN-1: One of the proposed HTN models is named HTN-1, as illustrated in Fig. 2(a). There are six layers, including a two-dimensional convolutional (2D-Conv) layer and a one-dimensional TD (1D-TD) layer. Then, use flatten to vectorize the output features. Through the 1D-TD layer, we can extract the spectral low-rank features and reduce the spectral data redundancy.

Suppose that the lth is the TD layer based on the 1D-Tucker model. From (2), it can be inferred that

$$\underline{\mathbf{Y}}^{(l)} = \sigma\left(\underline{\mathbf{X}}^{(l)} \times_1^1 \mathbf{W}^{(l,3)}\right). \tag{5}$$

2) HTN-2: As shown in Fig. 2(b), the HTN-2 has 12 layers, which is composed of 3D-Conv layer and 1D-TD layer. In order to give consideration to both efficacy and efficiency, the TD layer only projects the subspace of mode-3 (i.e., spectral dimension), which can ensure the computational efficiency without pooling layer. Also, some convolutional layers adopt rectangular convolution template (i.e., the dimensions of the first two dimensions are not equal) [59], [60], [61], which cannot only reduce the

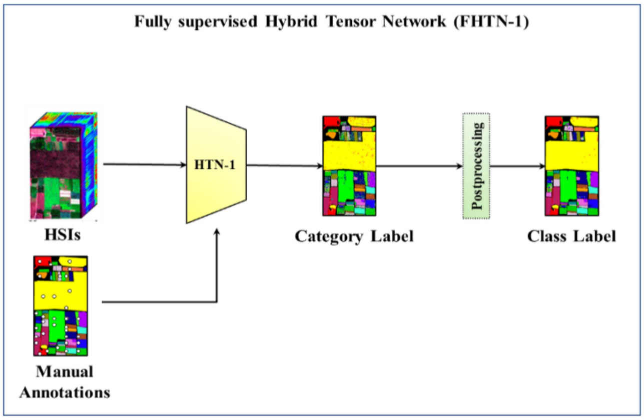


Fig. 3. Proposed fully supervised network HTN-1.

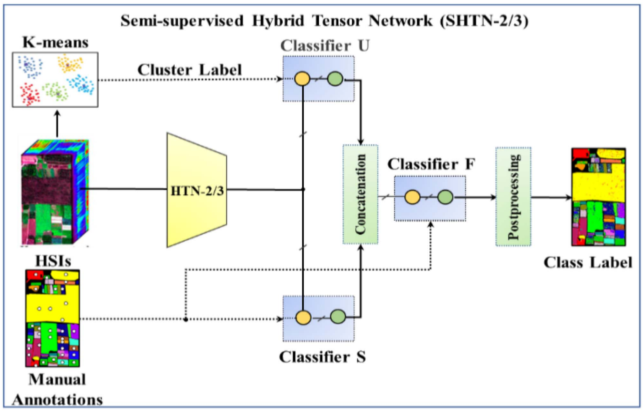


Fig. 4. Proposed semisupervised network HTN-2/3.

amount of calculation but also extract features in different directions and improve the effect of feature expression.

3) HTN-3: Fig. 2(c) shows the architectures of HTN-3, which contains 3D-Conv layer and 3D-TD layer. If the TD layer is based on 3D-Tucker model, it is defined in (2). After training, the high-dimensional TD layer will extract the optimal low-rank subspace of each dimension of the input sample so as to obtain the multidimensional joint features of the input sample. Compared with the 1D-Tucker model used in the first two models, the high-order TD layer not only performs subspace projection for mode-3 (i.e., spectral dimension) but also performs synchronous feature extraction for mode-1 and mode-2 (i.e., spatial dimension).

B. HTN for HSI Classification

Based on HTN-1, we design a fully supervised network for HSI classification. However, the fully supervised network needs lots of manual annotations to obtain better classification results. Therefore, we embed the fully supervised model within the semisupervised network to reduce the dependence on labeled data and obtain better classification results. Also, a postprocessing technique is employed to improve the classification accuracy. Due to the local spatial correlation of HSIs, the pixels in the same spatial neighborhood have the same class label. According to this characteristic, the classification result obtained by the classifier can be further modified. First, select the pixels that need postprocessing, and count the class labels of other pixels (including the class label of the current pixel itself) in the spatial neighborhood of this pixel (such as 8 neighborhood, 24 neighborhood, and 35 neighborhood). Then, select the class label with the highest proportion as the final classification label of the pixel. After processing the classification results through the majority voting (MV) strategy, we can make full use of the spatial context information, smooth the "salt and pepper noise" on the classification maps, and improve the classification accuracy.

1) FHTN: The fully supervised network FHTN-1 is illustrated in Fig. 3. Through the HTN-1 to extract low-rank spectral features, use the MV strategy to smooth the noise for postprocessing. Suppose that there are I training samples $\underline{\mathbf{X}}_i^{(0)}(i = 1, 2, \ldots, I)$ and the labels of $\underline{\mathbf{X}}_i^{(0)}$ are $\mathbf{y}_i \in \mathbb{R}^C$, then the loss function of FHTN-1 is

$$\ell = -\frac{1}{I}\sum_{i=1}^{I} \mathbf{y}_i^T \log(\hat{\mathbf{y}}_i). \tag{6}$$

The FHTN-1 is regarded as a nonlinear function $\mathbf{f}(\cdot, \underline{\boldsymbol{\theta}})$, where the $\underline{\boldsymbol{\theta}}$ is the weight parameter. Then, the FHTN-1 can be optimized through backpropagation based on the gradient descent method [37], [62] and it is updated by

$$\underline{\boldsymbol{\theta}} \leftarrow \underline{\boldsymbol{\theta}} - \lambda \nabla_{\underline{\boldsymbol{\theta}}} \ell \tag{7}$$

where λ is the learning rate. The optimization algorithm is summarized in Algorithm 1. From (1) and (2), we can compute the computational complexity of FHTN-1 as follows:

$$O\left(3\left(\sum_{l\in S_1}\left(I_3^{(l-1)}\prod_{n=1}^{3} I_n^{(l)} \prod_{n=1}^{2} K_n^{(l-1)}\right)\right.\right.$$
$$\left.\left. + \sum_{l\in S_2}\left(I_3^{(l)}\prod_{n=1}^{3} I_n^{(l-1)}\right)\right)\right) \tag{8}$$

where S_1 and S_2 are the sets containing all convolutional layer serial numbers and all Tucker layer serial numbers, respectively.

2) SHTN: As shown in Fig. 4, the two semisupervised networks (i.e., SHTN-2 and SHTN-3) are composed of HTN-2 and HTN-3, respectively. Also, the network contains multiple supervised classifiers and an unsupervised clustering module [52] to achieve high classification accuracy. K-means [63] generates unsupervised clustering labels, the classifier U learns all sample features, the classifier S learns labeled sample features, and the classifier F fuses labeled sample features. The input samples of K-means are the original HSIs' pixels, the input samples of fully connected classifier U and S are the final output features of HTN-2/3 (i.e., $\mathbf{y}^{(L)} \in \mathbb{R}^{K^{(L)}}$, $K^{(L)} = 512$), and the input samples of F are spliced by the output features of the first hidden layer of U and S.

In the semisupervised network, the total loss function is composed of the three submodules' loss function. Suppose that K-means get $\tilde{C}$ clusters, and the cluster label of pixel $\underline{\mathbf{X}}_i$ is $\mathbf{y}_{ui} \in \mathbb{R}^{\tilde{C}}(i = 1, 2, \ldots, I_1 I_2)$, then the loss function of fully

Algorithm 1: Optimization of FHTN.

```
Input: training samples $\underline{\mathbf{X}}_i^{(0)} \in \mathbb{R}^{I_1^{(0)} \times I_2^{(0)} \times I_3^{(0)}} (i = 1, 2, \ldots, I)$, testing samples $\underline{\mathbf{X}}_{\text{test}} \in \mathbb{R}^{I_1^{(0)} \times I_2^{(0)} \times I_3^{(0)}}$, and class labels $\mathbf{y}_i \in \mathbb{R}^C (i = 1, 2, \ldots, I)$;
Output: model $\mathbf{f}(\bullet, \underline{\boldsymbol{\theta}})$ and estimated labels $\hat{\mathbf{y}}_i$;

Initialize $\mathbf{f}(\bullet, \underline{\boldsymbol{\theta}})$, where $\underline{\boldsymbol{\theta}}$ is the parameter of FHTN-1 and set maximum epochs $T$;
while $\underline{\boldsymbol{\theta}}$ not converged and maximum epochs not reached do
    compute $\ell$ by (6);
    update $\underline{\boldsymbol{\theta}}$ by (7) using gradient descent;
end while
Feed the test samples $\underline{\mathbf{X}}_{\text{test}}$ to $\mathbf{f}(\bullet, \underline{\boldsymbol{\theta}})$ and estimate their labels $\hat{\mathbf{y}}_i$.
Return $\underline{\boldsymbol{\theta}}$ and $\hat{\mathbf{y}}_i$.
```

Algorithm 2: Optimization of SHTN.

```
Input: subtensors $\underline{\mathbf{Y}}_i^{(0)} \in \mathbb{R}^{I_1^{(0)} \times I_2^{(0)} \times I_3^{(0)}} (i = 1, 2, \ldots, I_1 I_2)$ of HSIs, testing samples $\underline{\mathbf{Y}}_{\text{test}} \in \mathbb{R}^{I_1^{(0)} \times I_2^{(0)} \times I_3^{(0)}}$, and a few class labels $\mathbf{y}_j \in \mathbb{R}^C (j = 1, 2, \ldots, I_l)$;
Output: model $\mathbf{f}(\bullet, \underline{\boldsymbol{\theta}})$ and estimated labels $\hat{\mathbf{y}}_{fi} \in \mathbb{R}^C (i = 1, 2, \ldots, I_1 I_2)$;

Initialize $\mathbf{f}(\bullet, \underline{\boldsymbol{\theta}})$, where $\underline{\boldsymbol{\theta}} = \{\underline{\boldsymbol{\theta}}_U, \underline{\boldsymbol{\theta}}_L, \underline{\boldsymbol{\theta}}_F, \underline{\boldsymbol{\theta}}_{\text{HTN}}\}$ and set maximum epochs $T$;
while $\underline{\boldsymbol{\theta}}$ not converged and maximum epochs not reached do
    compute $\ell$ by (9)–(12);
    update $\underline{\boldsymbol{\theta}}$ using gradient descent;
end while
Feed the test samples $\underline{\mathbf{Y}}_{\text{test}}$ to $\mathbf{f}(\bullet, \underline{\boldsymbol{\theta}})$ and estimate their labels $\hat{\mathbf{y}}_{fi}$.
Return $\underline{\boldsymbol{\theta}}$ and $\hat{\mathbf{y}}_{fi}$.
```

connected network U is

$$\ell_u = -\frac{1}{I_1 I_2} \sum_{i=1}^{I_1 I_2} \mathbf{y}_{ui}^T \log(\hat{\mathbf{y}}_{ui}) \tag{9}$$

where $\hat{\mathbf{y}}_{ui}$ is the estimation label of the subtensor sample $\underline{\mathbf{Y}}_i^{(0)}$. Suppose that there are I_l manually labeled samples $\underline{\mathbf{Y}}_j^{(0)} (j = 1, 2, \ldots, I_l, I_l \ll I_1 I_2)$, then the loss function of classifier S is

$$\ell_l = -\frac{1}{I_l} \sum_{i=1}^{I_l} \mathbf{y}_j^T \log(\hat{\mathbf{y}}_j). \tag{10}$$

Similarly, the loss function of F is

$$\ell_f = -\frac{1}{I_l} \sum_{i=1}^{I_l} \mathbf{y}_j^T \log(\hat{\mathbf{y}}_{fi}) \tag{11}$$

where $\hat{\mathbf{y}}_{fi}$ is also the estimation label of the subtensor sample $\underline{\mathbf{Y}}_j^{(0)}$ generated by F. The total loss is computed as follows:

$$\ell = \ell_f + \ell_l + \ell_u. \tag{12}$$

We also could denote SHTN as a nonlinear function $\mathbf{f}(\bullet, \underline{\boldsymbol{\theta}})$, where $\underline{\boldsymbol{\theta}}$ contains all learnable parameters (i.e., $\{\underline{\boldsymbol{\theta}}_U, \underline{\boldsymbol{\theta}}_L, \underline{\boldsymbol{\theta}}_F, \underline{\boldsymbol{\theta}}_{\text{HTN}}\}$). Then, train the SHTN in an end-to-end fashion by gradient descent. Computational cost in SHTN can be measured by the number of multiplications contained in the tensor contraction. Also, the computation of feedforward, backpropagation, and gradient descent is similar [37].

Therefore, the computational complexity of SHTN-2 and SHTN-3 is defined as

$$O\left(3\left(\sum_{l\in S_1}\left(I_4^{(l-1)}\prod_{n=1}^{4} I_n^{(l)} \prod_{n=1}^{3} K_n^{(l-1)}\right) + \sum_{l\in S_2}\left(I_3^{(l)}\prod_{n=1}^{4} I_n^{(l-1)}\right)\right)\right) \tag{13}$$

and

$$O\left(3\left(\sum_{l\in S_1}\left(I_4^{(l-1)}\prod_{n=1}^{4} I_n^{(l)} \prod_{n=1}^{3} K_n^{(l-1)}\right) + \sum_{l\in S_2} I_4^{(l-1)}\prod_{n=1}^{3}\left(I_n^{(l-1)} I_n^{(l)}\right)\right)\right) \tag{14}$$

where S_1 and S_2 are used in (8). As SHTN-3 uses the high-order convolutional and TD operations, its computational complexity is slightly larger than SHTN-2.

IV. Experiments

A. Datasets

To evaluate the performance of the proposed models, six datasets[1] are used in our work, including three classic datasets, i.e., Indian Pines, PaviaU, and Salinas, and three new datasets named LongKou, HanChuan, and HongHu from WHU-Hi dataset [6]. The detailed information of datasets is shown in Table I. The three new datasets are provided by the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. These three datasets have the characteristics of high spatial resolution, high labeled pixels proportion, and abundant ground feature types. Fig. 5 shows false-color images of six datasets.

B. Setup

According to whether the proposed networks have postprocessing operation, we divide them into six networks: FHTN-1, FHTN-1-MV, SHTN-2, SHTN-2-MV, SHTN-3, and SHTN-3-MV. Then, several state-of-the-art HSI classification methods

[1] [Online]. Available: http://rsidea.whu.edu.cn/resource_WHUHi_sharing.htm. Accessed: Feb. 15, 2023. [Online]. Available: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes, Accessed: Feb. 15, 2023.

TABLE I
DESCRIPTION OF SIX DATASETS

Data Sets	Indian Pines	PaviaU	Salinas	Longkou	HanChuan	HongHu
Spatial dimension	145×145	610×340	512×217	550×400	1217×303	940×475
Bands	200	103	204	270	274	270
Spatial resolution	20m	1.3m	3.7m	0.463m	0.109m	0.043m
Equipment	AVIRIS	ROSIS	AVIRIS	UAV	UAV	UAV
Class	16	9	16	9	16	22

The bold entities are the best results to facilitate reading.

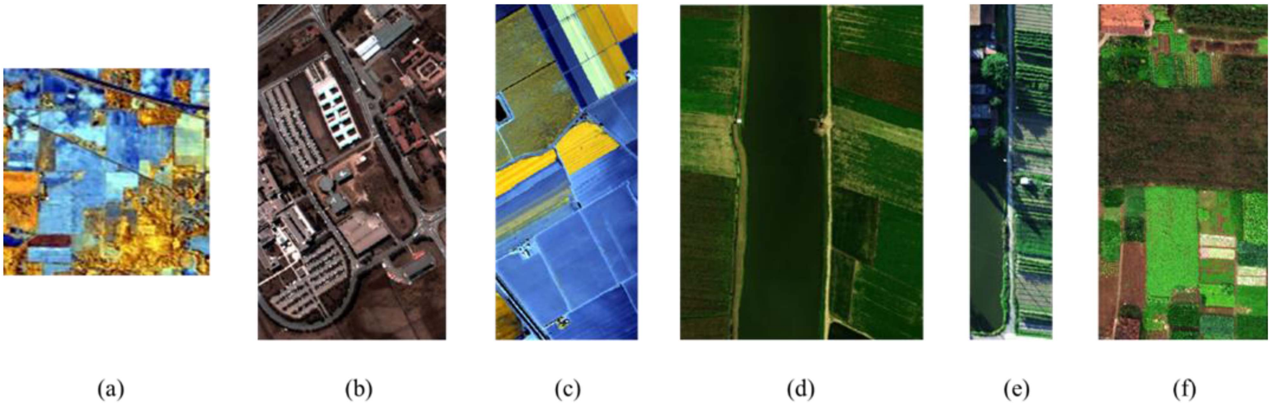


Fig. 5. False-color images of six datasets. (a) Indian Pines. (b) PaviaU. (c) Salinas. (d) LongKou. (e) HanChuan. (f) HongHu.

are used in the performance comparison. They are two classic machine learning methods, SVM [64] and KNN [48], a classic CNN [6] model, an advanced self-supervised learning with adaptive distillation (SSAD) [46] method, semisupervised classification with CNN (SCNN) [52], superpixel contracted graph-based learning (SGL) [56], and with unsupervised feature learning (WUFL) [8].

All the fully supervised networks and seminetworks are optimized according to Algorithm 1 and Algorithm 2, respectively. The learning rate is set as 0.001 and the exponential decay index is set as 0.9. They are then optimized by the Adam optimizer and use ReLU and Swish [65] activation function for fully supervised network and semisupervised network, respectively. For the benchmark datasets, we randomly select 3, 5, 10, 15, and 20 samples for each class as manual labeled samples and the first two dimensions of the input subtensor $\underline{\mathbf{X}}^{(0)}$ are 9. While we choose 100 samples for the new datasets and set $I_1^{(0)} = I_2^{(0)} = 23$. Then, we set the cluster numbers of K-means as 100 and the experiments carried out in this article are repeated 10 times.

To evaluate the performance of the networks, we use overall accuracy (OA), average class accuracy (AA), and Kappa coefficient as the evaluation index. All the models are implemented by Tensorflow and run on a computer equipped with Intel(R) Xeon E3-1225 V6 CPU @ 3.31 GHz, 64GB RAM, and NVIDIA GeForce RTX 2080Ti GPU (11GB RAM).

C. Efficacy

1) Feature Representation Ability: In order to verify that our proposed methods have good multiscale and low-rank feature representation ability, we select two datasets, such as Indian Pines and LongKou, and use Umap to visualize the original features and the low-rank features obtained by HTNs. For Indian Pines, choose 3 samples for each class and, for LongKou dataset, 100 samples per class are selected. From Figs. 6 and 7, we can see that before HTN processing, hyperspectral pixels generally exhibit spectral variability, and the confusion between classes is serious, which is not conducive to classification. Then, the pixels are input into our proposed HTNs. After processing, the features

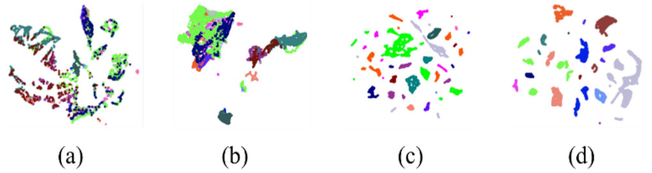


Fig. 6. Umap of Indian Pines dataset. (a) Scatter plot of original pixel. (b) HTN-1 output feature scatter plot. (c) HTN-2 output feature scatter plot. (d) HTN-3 output feature scatter plot.

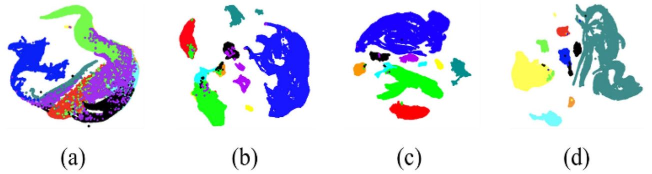


Fig. 7. Umap of LongKou dataset. (a) Scatter plot of original pixel. (b) HTN-1 output feature scatter plot. (c) HTN-2 output feature scatter plot. (d) HTN-3 output feature scatter plot.

TABLE II
CLASSIFICATION ACCURACIES OF INDIAN PINES THROUGH DIFFERENT METHODS

Samples	3			5			10			15			20		
Methods	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
KNN	26.78	43.22	20.17	35.93	49.11	29.37	44.31	57.66	37.75	50.57	62.46	44.74	51.32	64.4	45.68
SVM	33.63	43.34	26.13	44.31	59.20	38.11	52.72	63.45	47.20	56.29	69.90	51.13	62.00	74.38	57.42
CNN	40.74	47.41	32.61	46.18	57.64	39.25	54.64	65.92	48.71	59.71	73.97	54.72	62.7	76.13	58.01
FHTN-1	42.01	47.93	34.82	48.55	63.77	42.34	56.17	70.08	50.61	61.20	75.78	56.52	66.99	79.71	62.66
FHTN-1-MV	49.40	55.86	42.89	54.22	68.68	48.48	61.69	74.61	56.74	68.21	81.28	64.38	73.55	85.52	70.20
SSAD	85.98	76.87	73.85	88.83	83.14	80.81	93.38	90.1	88.73	94.93	90.45	89.14	96.04	91.1	95.47
SCNN	86.38	94.17	84.7	87.15	92.57	85.48	92.62	96.98	91.59	95.9	97.98	95.34	96.77	98.12	96.31
SGL	79.4	87.64	76.68	82.49	88.5	80.06	90.97	94.47	89.7	92.84	95.79	91.83	93.84	90.38	92.97
WUFL	86.73	93.08	85.06	89.73	93.66	88.37	95.07	98.57	94.42	95.42	97.73	94.8	97.62	98.94	97.28
SHTN-2	97.49	98.12	97.15	98.08	98.34	98.19	98.63	99.06	98.43	99.05	99.31	98.92	99.43	99.81	99.47
SHTN-2-MV	97.84	99.02	97.55	98.82	98.46	98.67	98.94	99.34	98.79	99.31	99.55	99.21	99.75	99.83	99.72
SHTN-3	97.75	98.81	97.46	98.54	99.07	98.35	98.79	99.46	98.62	99.09	99.70	98.96	99.60	99.85	99.54
SHTN-3-MV	**98.11**	**99.43**	**97.85**	**99.00**	**99.14**	**98.86**	**99.01**	**99.54**	**98.88**	**99.76**	**99.89**	**99.73**	**99.86**	**99.92**	**99.84**

The bold entities are the best results to facilitate reading.

TABLE III
CLASSIFICATION ACCURACIES OF PAVIAU THROUGH DIFFERENT METHODS

Samples	3			5			10			15			20		
Methods	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
KNN	39.10	52.02	28.37	48.48	62.33	38.23	56.28	70.82	47.29	61.76	74.78	53.32	64.12	75.79	55.87
SVM	42.53	50.96	30.90	44.31	56.67	34.56	53.06	68.55	44.75	62.53	72.38	53.42	63.96	76.09	55.59
CNN	52.21	52.69	38.13	56.98	61.61	46.48	70.87	71.58	61.76	71.51	72.15	63.05	77.64	78.49	70.72
FHTN-1	53.51	53.44	40.24	63.90	62.68	53.23	72.31	71.97	63.99	77.90	80.63	71.37	79.98	80.71	73.70
FHTN-1-MV	55.24	57.88	42.65	67.69	64.72	57.31	76.24	77.00	69.03	81.80	84.00	76.03	84.51	85.26	79.56
SSAD	88.8	85.58	81.45	89.26	89.31	86.04	92.68	92.61	90.28	93.3	94.53	91.84	95.6	95.78	94.51
SCNN	94.76	96.84	93.18	96.84	96.76	95.83	97.91	97.35	97.23	99.03	98.69	98.71	99.15	98.85	98.87
SGL	83.22	88.12	78.93	88.01	90.81	84.61	92.61	92.68	90.28	95.1	94.74	93.53	95.51	95.05	94.06
WUFL	91.1	92.5	88.39	96.36	96.33	95.21	97.77	97.12	97.05	98.32	98.9	97.66	99.09	98.97	98.78
SHTN-2	97.74	97.82	97.02	98.02	98.28	97.39	98.84	98.79	98.45	99.23	99.20	98.78	99.55	99.46	99.41
SHTN-2-MV	98.26	98.17	97.72	98.67	98.72	98.20	99.12	99.12	98.83	99.32	99.36	99.10	99.73	99.48	99.64
SHTN-3	97.78	98.54	97.09	98.55	98.65	98.08	99.23	99.24	98.99	99.34	99.39	99.12	99.59	99.59	99.45
SHTN-3-MV	**98.78**	**98.67**	**98.39**	**99.11**	**99.04**	**98.83**	**99.42**	**99.28**	**99.24**	**99.43**	**99.57**	**99.25**	**99.97**	**99.94**	**99.96**

The bold entities are the best results to facilitate reading.

are more distinguishable and easier to classify. Also, through the feature expression, clustering optimization, and information fusion, our network can get the better visual effect than them. It can be seen that the network that we proposed can effectively express the multiscale and low-rank features of HSIs.

2) Classification Accuracy: Tables II –IV present the metrics of different methods for the classic datasets. Under different labeled samples' conditions for each class, the DL methods can yield better classification results than the machine learning methods. It indicates that the DL methods can learn the useful high-level features for HSI classification. Compared with other state-of-the-art algorithms, our proposed network achieves better classification accuracy. For the classic datasets, our proposed semisupervised networks, such as SHTN-2 and SHTN-3, are less dependent on manual annotation and achieve higher classification accuracy. Due to the limitation of spatial resolution, Indian Pines dataset has obvious spectral variability problem, which poses great obstacles for HSIs feature learning and classification, especially has a great negative impact on classifiers based on spectral features (such as SVM and KNN). However, the proposed SHTN-2 and SHTN-3 can extract multiscale spectral–spatial features for accurate classification, which shows that it has good learning ability and can mine effective information. Also, when the amount of manual labeling changes, the classification performance of SHTN-2 and SHTN-3 is better than the state-of-the-art comparative algorithms. Meanwhile, compared with SHTN model, SGL involves a large number of parameter adjustment steps with strong parameter dependence. It is a time-consuming and labor-consuming complex task to select these parameters. For Salinas dataset, when using 20 samples per class, the OA, AA, and Kappa of SHTN-3 can reach 99.77%, 99.93%, and 99.74%, respectively. Through the postprocessing operation, the classification accuracy will be further improved. The overall classification accuracy of SHTN is higher than the SCNN method, using only stacked convolutional layer without the TD layer. It demonstrates that TD combined with convolutional layers can extract low-rank features and improve classification accuracy. As shown in Figs. 8–10, the

TABLE IV
CLASSIFICATION ACCURACIES OF SALINAS THROUGH DIFFERENT METHODS

Samples	3			5			10			15			20		
Methods	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
KNN	67.27	71.13	63.93	76.32	80.93	73.72	78.14	84.84	75.78	79.59	87.27	77.44	81.3	88.6	79.26
SVM	76.63	82.31	74.09	78.05	83.82	75.50	79.87	88.78	77.77	84.61	91.7	82.96	84.84	91.98	83.18
CNN	78.14	81.32	75.61	79.61	84.79	77.33	80.51	87.16	78.35	84.89	90.87	83.19	85.95	92.33	84.38
FHTN-1	80.16	84.21	78.07	81.94	87.01	79.99	82.66	89.46	80.65	86.98	93.08	85.56	87.23	93.49	85.82
FHTN-1-MV	83.05	87.28	81.26	84.90	89.87	83.26	85.17	91.79	83.41	91.46	96.37	90.51	92.41	96.98	91.55
SSAD	94.48	90.09	93.87	95.29	92.31	94.61	97.33	96.5	97.04	97.78	96.83	97.53	98.11	97.01	97.89
SCNN	86.62	91.8	85.13	87.39	92.96	85.99	90.56	95.7	89.5	92.44	96.6	91.6	92.66	96.88	91.84
SGL	96.05	92.95	96.05	97.18	97.10	96.77	99.04	98.86	98.93	99.07	98.91	98.97	99.09	98.96	98.99
WUFL	92.79	95.88	91.99	93.67	96.91	92.83	94.86	97.59	94.28	95.26	98.16	94.73	95.59	98.27	95.09
SHTN-2	97.42	98.85	97.13	97.78	99.17	97.54	99.22	99.66	99.13	99.61	99.82	99.57	99.71	99.87	99.68
SHTN-2-MV	98.36	99.47	98.18	98.50	99.53	98.34	99.40	99.72	99.33	99.67	99.86	99.63	99.81	99.93	99.79
SHTN-3	97.68	99.05	97.42	98.34	99.46	98.49	99.30	99.75	99.22	99.66	99.87	99.62	99.77	99.93	99.74
SHTN-3-MV	**98.44**	**99.53**	**98.27**	**98.71**	**99.61**	**98.56**	**99.52**	**99.77**	**99.46**	**99.83**	**99.94**	**99.81**	**99.89**	**99.97**	**99.89**

The bold entities are the best results to facilitate reading.

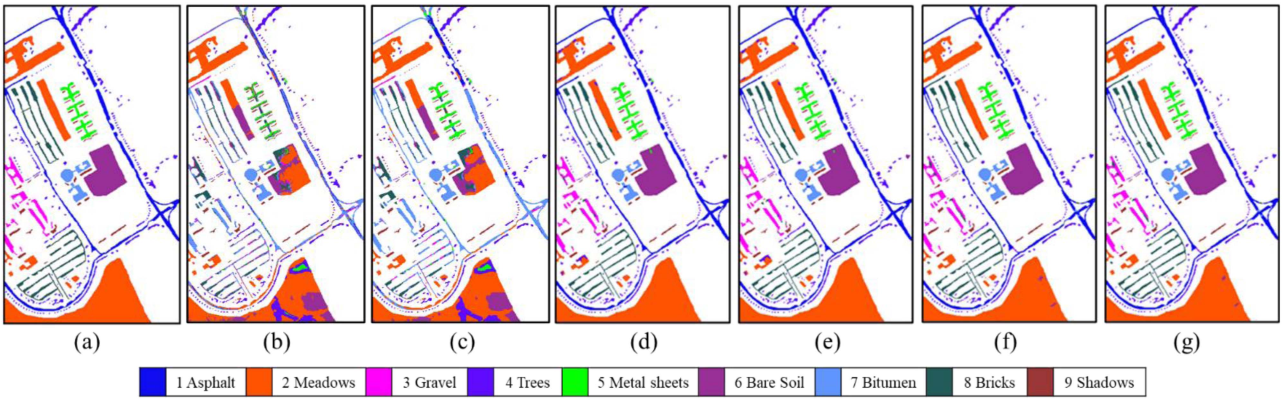


Fig. 8. Ground truth and classification maps achieved by our proposed models on the PaviaU. (a) GT. (b) FHTN-1. (c) FHTN-1-MV. (d) SHTN-2. (e) SHTN-2-MV. (f) SHTN-3. (g) SHTN-3-MV.

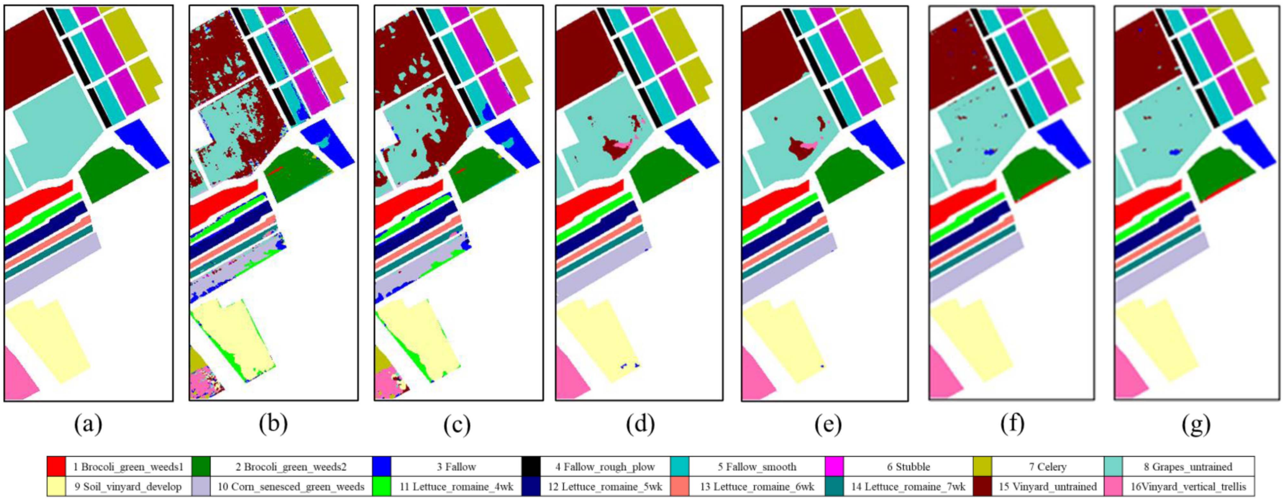


Fig. 9. Ground truth and classification maps achieved by our proposed models on the Salinas. (a) GT. (b) FHTN-1. (c) FHTN-1-MV. (d) SHTN-2. (e) SHTN-2-MV. (f) SHTN-3. (g) SHTN-3-MV.

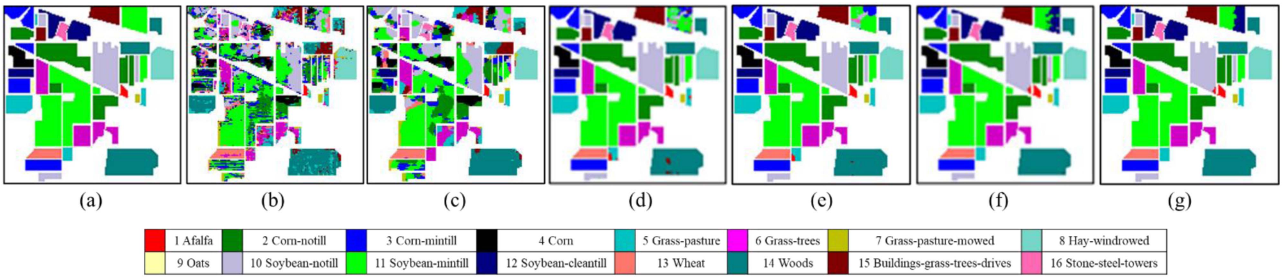


Fig. 10. Ground truth and classification maps achieved by our proposed models on the Indian Pines. (a) GT. (b) FHTN-1. (c) FHTN-1-MV. (d) SHTN-2. (e) SHTN-2-MV. (f) SHTN-3. (g) SHTN-3-MV.

TABLE V
CLASSIFICATION ACCURACIES OF NEW DATASETS THROUGH DIFFERENT METHODS

Data Sets	LongKou			HanChuan			HongHu		
Methods	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
KNN	83.21	79.04	78.78	69.15	59.43	64.29	62.63	58.19	55.64
SVM	94.12	93.54	92.37	78.72	73.96	75.39	72.76	69.52	67.05
CNN	97.30	97.40	96.47	87.13	84.93	84.97	85.43	86.16	82.10
FHTN-1	97.83	98.11	97.17	88.71	86.67	86.84	87.46	86.26	84.32
FHTN-1-MV	98.92	98.89	98.58	90.04	88.40	88.40	89.35	88.94	86.69
SSAD	94.36	93.48	92.67	85.5	82.12	83.14	89.27	88.39	86.59
SCNN	94.69	95.2	93.13	91.2	88.94	89.72	85.05	84.23	81.47
SGL	91.85	90.65	87.64	84.6	86.21	84.02	87.81	87.5	84.51
WUFL	95.30	96.45	93.9	87.29	84.38	85.21	89.73	89.19	87.2
SHTN-2	98.09	98.40	97.50	92.52	91.79	91.29	90.93	91.72	88.69
SHTN-2-MV	99.23	99.12	98.99	93.07	92.35	91.92	92.54	93.43	90.74
SHTN-3	98.58	98.83	98.14	92.61	92.23	91.40	91.70	92.16	89.63
SHTN-3-MV	**99.24**	**99.47**	**99.01**	**94.31**	**94.52**	**93.36**	**94.50**	**95.24**	**93.09**

The bold entities are the best results to facilitate reading.

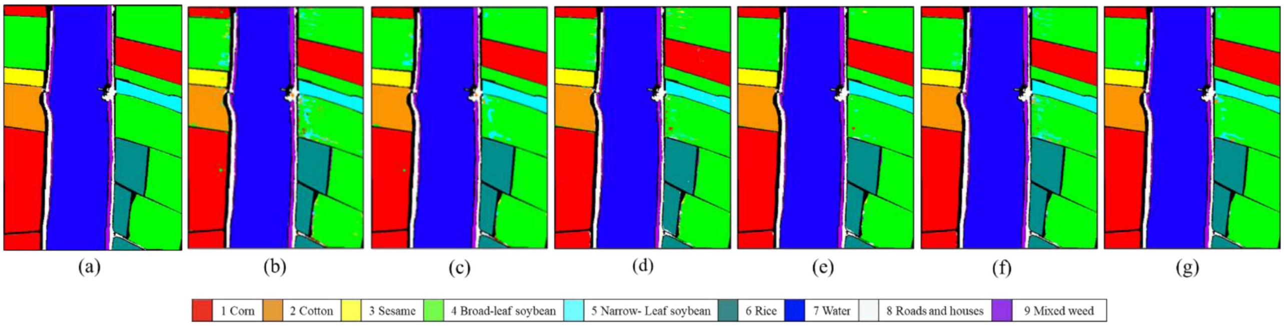


Fig. 11. Ground truth and classification maps achieved by our proposed models on the LongKou. (a) GT. (b) FHTN-1. (c) FHTN-1-MV. (d) SHTN-2. (e) SHTN-2-MV. (f) SHTN-3. (g) SHTN-3-MV.

proposed networks can also obtain good visual classification results. Compared with the SHTN-2, the SHTN-3 uses high-order TD layers to extract low-rank features, improving the category discrimination and reducing information redundancy.

For the new datasets, Table V and Figs. 11–13 show that our methods also have good classification results. As in the LongKou dataset, when randomly selecting 100 samples per class as labeled, due to the high imaging quality of this dataset, the classifier can be fully trained. The traditional methods, SVM and KNN, have achieved 94.12% and 83.21% in the OA, but they do not make full use of spatial information. Under the same condition, from the perspective of spectral–spatial correlation, our networks excavate the deep distinguishing features, alleviating the impact of the spectral variability, and achieve the OA at 98.58%. Although the scene of HanChuan dataset is complex, and some categories are seriously confused, which is very easy to cause classification errors, the average classification accuracy and Kappa coefficient obtained by SHTN-3 on HanChuan dataset are also higher than other methods. When SCNN selects 100 samples for each class on this dataset, class 6 and class 13 are seriously confused. Under the same condition, the average classification accuracy of SHTN-3 is 92.23%. It shows that SHTN-3

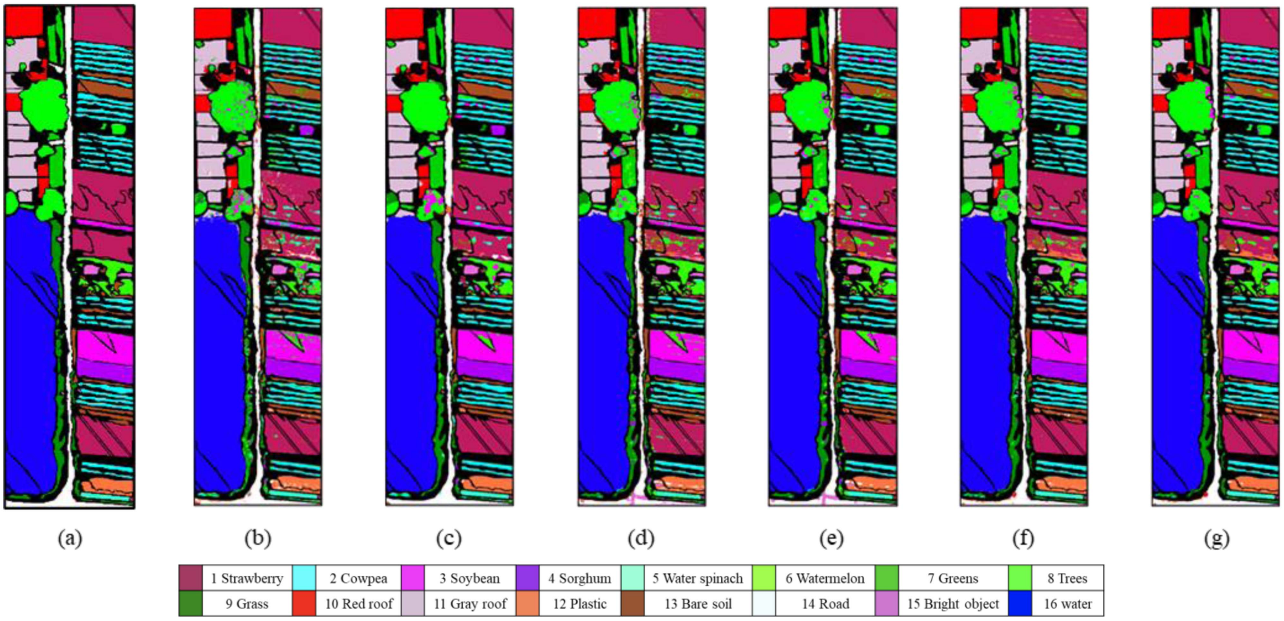


Fig. 12. Ground truth and classification maps achieved by our proposed models on the HanChuan. (a) GT. (b) FHTN-1. (c) FHTN-1-MV. (d) SHTN-2. (e) SHTN-2-MV. (f) SHTN-3. (g) SHTN-3-MV.

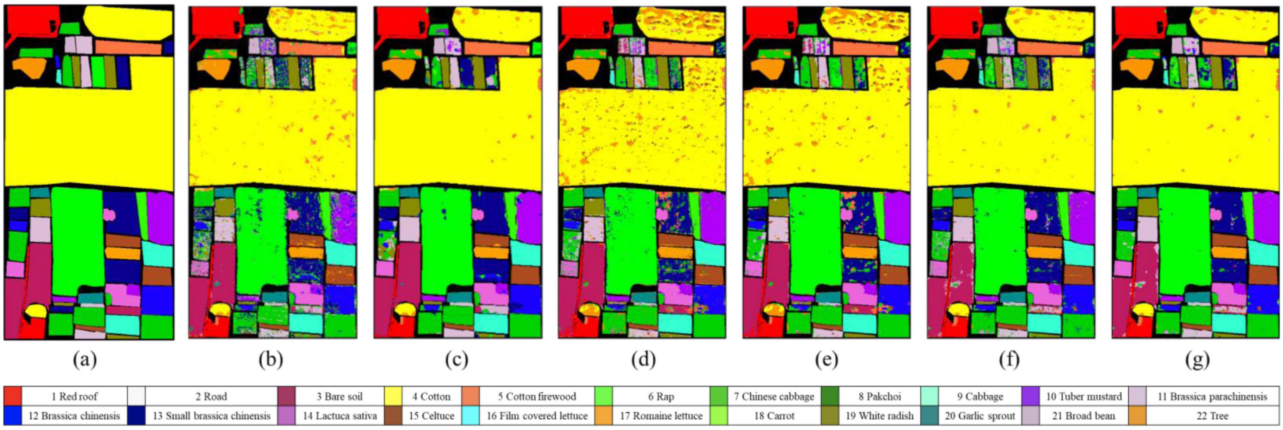


Fig. 13. Ground truth and classification maps achieved by our proposed models on the HongHu. (a) GT. (b) FHTN-1. (c) FHTN-1-MV. (d) SHTN-2. (e) SHTN-2-MV. (f) SHTN-3. (g) SHTN-3-MV.

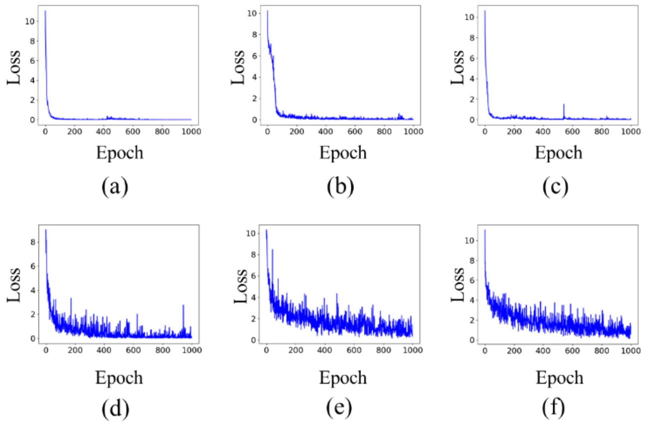
can combine high-order TD layer and 3D-conv layer to extract the multiscale spectral–spatial features and low-rank features between different classes to accurately distinguish different categories. In the HongHu scene, it has 22 classes, which are difficult to distinguish accurately only from the spectral dimension. Especially, class 12 is seriously confused with other categories. Compared with the other latest methods (i.e., WUFL and SGL), the SHTN-3-MV gains an average improvement of OA + 4.77 and 6.69%. In summary, it shows that our proposed networks have strong generalization and robustness and can achieve good classification results for different datasets with different difficulties.

3) Ablation Study: The ablation study has been done on the six datasets. Different forms of TD layers can extract low-rank features from different dimensions of HSIs. Therefore, to analyze the effectiveness of our proposed HTNs, SHTN-3 is compared with SCNN, SHTN-Spa, and SHTN-2. The first model does not contain the TD layer and the remaining two models only extract spectral or spatial features, respectively. Table VI presents the OA of SHTN-3 and the ablation methods. It can be seen that on the six datasets, the results obtained by SHTN-3 are better than the other three models, indicating that high-order TD layers can extract spectral–spatial low-rank features for HSI classification. It is proved that high-order TD can further improve the classification accuracy on the basis of other forms of TD. Also, the models with TD layers can achieve better results than those without TD, which shows that TD can combine with the convolutional layers to extract the multiscale spectral–spatial features and low-rank features of the HSIs that are helpful for classification. From Tables II–V and

TABLE VI
OA of SHTN-3 and Ablation Methods on Six Datasets

Data Sets	SCNN	SHTN-Spa	SHTN-2	SHTN-3
Indian Pines	86.38	93.88	97.49	**97.75**
PaviaU	94.76	95.78	97.74	**97.78**
Salinas	86.62	90.77	97.42	**97.68**
LongKou	94.69	97.08	98.09	**98.58**
HanChuan	91.20	92.30	92.52	**92.61**
HongHu	85.05	87.49	90.93	**91.70**

The bold entities are the best results to facilitate reading.



(a) (b) (c)

(d) (e) (f)

Fig. 14. Convergence curve of SHTN-3 on the six datasets. (a) Indian Pines. (b) PaviaU. (c) Salinas. (d) LongKou. (e) HanChuan. (f) HongHu.

Figs. 8–13, we can get that, through the postprocessing technique, the efficacy of the classification task has been further improved. Due to the characteristic that the ground object distribution has local spatial continuity, the noise on the classification map is smoothed, and it has good visual effect.

D. *Efficiency*

1) Convergence Curves: Fig. 14 shows the convergence curve of SHTN-3 on the six datasets. For classic datasets and new datasets, we randomly select 3 and 100 samples per class, respectively. All the losses decrease as the training epoch moves on and then it is less than 1e-4. Fig. 15 shows the convergence curves and OA of the training set at the 3000th epoch of SHTN-3 compared with the classic fully supervised CNN method, and three other advanced semisupervised algorithms, SSAD, SCNN, and WUFL algorithms, on the Indian Pines dataset. The Indian Pines dataset contains 16 land-cover classes. For the fully supervised competing model, CNN, we randomly select 500 samples per class (total 3635 samples as some classes having less than 500 samples, we choose 15 samples for these classes) for training. For semisupervised methods, we randomly selected 3 samples per class, totaling 48 samples used for training. We can observe that the convergence curve of our proposed SHTN-3 converges quickly with small fluctuations and a relatively stable pattern, which demonstrates the feasibility and stability of our proposed models.

2) Time Costs: Table VII presents the training and testing time on the Salinas and HongHu datasets with 1000 epochs of the proposed methods. The proposed FHTN-1 costs less training time on the Salinas dataset but yields lower classification accuracy than the other two semisupervised models. Both training and testing time of SHTN-3 is longer than SHTN-2, while it achieves higher classification accuracy. The main reason is that the high-order TD layers extract spectral and spatial features simultaneously. Thus, it requires more computation and takes longer time costs.

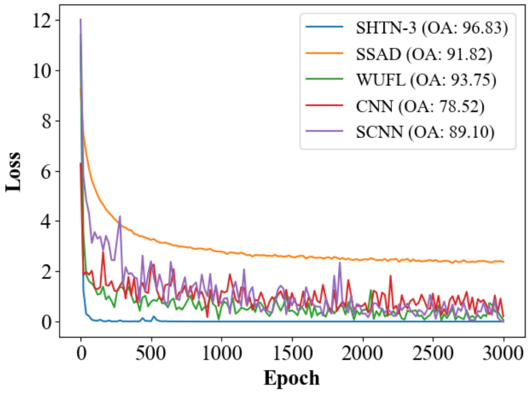


Fig. 15. Convergence curve and classification OA of the training set at the 3000th epoch of the proposed SHTN-3, compared with the classic fully supervised CNN model (500 labeled samples per class for training) and three other semisupervised methods SSAD, WUFL, and SCNN (3 labeled samples per class for training) on the Indian Pines dataset.

TABLE VII
Training and Testing Time of the Proposed Models (s)

Data Sets	Salinas		HongHu	
	Train	Test	Train	Test
FHTN-1	68.62	42.22	455.13	374.97
FHTN-1-MV	68.62	44.87	455.13	389.81
SHTN-2	211.65	40.23	451.37	80.38
SHTN-2-MV	211.65	43.24	451.37	93.41
SHTN-3	277.79	74.86	567.02	169.94
SHTN-3-MV	277.79	77.51	567.02	184.49

V. Discussions

HSIs have lots of redundant information, which has a great impact on classification. Our proposed HTNs combine tensor analysis and convolutional operation for multiscale spectral–spatial features and low-rank features representation, which can reduce the computational complexity of the model and the use of high-order TD can obtain the spectral–spatial features with strong correlation. Moreover, the scarcity in manual annotations is a very important problem for HSI classification. Therefore, we embed the HTNs into the semisupervised network, which can obtain the higher classification accuracy. Through the MV strategy, it can smooth the classification results graph. Although HTNs have achieved good classification results compared with the other state-of-the-art algorithm, there is still a room for improvement in its training and testing efficiency.

VI. Conclusion

In this article, several novel HTNs that represent multiscale spectral–spatial patterns and multidomain low-rank features for accurate classification have been proposed. Also, the HTNs combine both fully supervised and semisupervised frameworks, leveraging the labeled and unlabeled sample features and improving the classification accuracy through the postprocessing techniques. Experiments show that, for different datasets, the proposed networks are stable, and the efficacy and efficiency are better than some classic methods and a variety of advanced algorithms. The future direction of our work is to further improve the efficiency of the semisupervised network. Also, we can introduce the pooling layer into the HTNs for higher classification accuracy.

References

[1] S. Li, Q. Hao, G. Gao, and X. Kang, "The effect of ground truth on performance evaluation of hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7195–7206, Dec. 2018.

[2] C.-I. Chang and H. Ren, "An experiment-based quantitative and comparative analysis of target detection and image classification algorithms for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 2, pp. 1044–1063, Mar. 2000.

[3] S. Prasad and L. M. Bruce, "Limitations of principal components analysis for hyperspectral target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 625–629, Oct. 2008.

[4] Q. Zhang, L. Zhang, Y. Yang, Y. Tian, and L. Weng, "Local patch discriminative metric learning for hyperspectral image feature extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 612–616, Mar. 2014.

[5] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral–spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3232–3245, May 2020.

[6] Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, and L. Zhang, "WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF," *Remote Sens. Environ.*, vol. 250, 2020, Art. no. 112012.

[7] Y. Zhong, X. Lin, and L. Zhang, "A support vector conditional random fields classifier with a Mahalanobis distance boundary constraint for high spatial resolution remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1314–1330, Apr. 2014.

[8] W. Wei, S. Xu, L. Zhang, J. Zhang, and Y. Zhang, "Boosting hyperspectral image classification with unsupervised feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Mar. 2022, Art. no. 5502315.

[9] H. Yu et al., "Global spatial and local spectral similarity-based manifold learning group sparse representation for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3043–3056, May 2020.

[10] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[11] M. Imani and H. Ghassemian, "An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges," *Inf. Fusion*, vol. 59, pp. 59–83, 2020.

[12] M. Imani and H. Ghassemian, "Attribute profile based feature space discriminant analysis for spectral-spatial classification of hyperspectral images," *Comput. Elect. Eng.*, vol. 62, pp. 555–569, 2016.

[13] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.

[14] S. Yang, J. Hou, Y. Jia, S. Mei, and Q. Du, "Pseudolabel guided kernel learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 3, pp. 1000–1011, Mar. 2019.

[15] T. C. Bau, S. Sarkar, and G. Healey, "Hyperspectral region classification using a three-dimensional Gabor filterbank," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 9, pp. 3457–3464, Sep. 2010.

[16] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.

[17] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

[18] D. Letexier and S. Bourennane, "Estimation of N-mode ranks of hyperspectral images for tensor denoising," in *Proc. 17th Eur. Signal Process. Conf.*, 2009, pp. 2594–2597.

[19] N. Renard and S. Bourennane, "Dimensionality reduction based on tensor modeling for classification methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1123–1131, Apr. 2009.

[20] Y. Xu, B. Du, and F. Zhang, "Hyperspectral image classification via a random patches network," *ISPRS J. Photogramm. Remote Sens.*, vol. 142, no. 8, pp. 344–357, 2018.

[21] Y. Feng, R. Song, W. Ni, J. Zhu, and X. Wang, "A novel semi-supervised long-tailed learning framework with spatial neighborhood information for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, Feb. 2023, Art. no. 5501305.

[22] Y. Su, L. Gao, M. Jiang, A. Plaza, X. Sun, and B. Zhang, "NSCKL: Normalized spectral clustering with kernel-based learning for semisupervised hyperspectral image classification," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2022.3219855.

[23] Y. Yang et al., "Semi-supervised multiscale dynamic graph convolution network for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2022.3212985.

[24] W. Wei, S. Zhao, S. Xu, L. Zhang, and Y. Zhang, "Semi-supervised neural architecture search for hyperspectral imagery classification method with dynamic feature clustering," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, May 2023, Art. no. 5513314.

[25] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.

[26] L. Liu, W. Huang, B. Liu, L. Shen, and C. Wang, "Semisupervised hyperspectral image classification via Laplacian least squares support vector machine in sum space and random sampling," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4086–4100, Nov. 2018.

[27] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[28] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, Nov. 2010.

[29] D. Tuia, J. Muñoz-Marí, and G. Camps-Valls, "Remote sensing image segmentation by active queries," *Pattern Recognit.*, vol. 45, no. 6, pp. 2180–2192, 2012.

[30] J. Munoz-Mari, D. Tuia, and G. Camps-Valls, "Semisupervised classification of remote sensing images with active queries," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3751–3763, Oct. 2012.

[31] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, Nov. 2018.

[32] 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.

[33] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.

[34] Z.-H. Zhou and M. Li, "Semisupervised regression with cotraining-style algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 11, pp. 1479–1493, Nov. 2007.

[35] H. Wang, Y. Cheng, C. L. P. Chen, and X. Wang, "Hyperspectral image classification based on domain adversarial broad adaptation network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2022, Art. no. 5517813.

[36] Q. Xu, Y. Shi, X. Yuan, and X. X. Zhu, "Universal domain adaptation for remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Feb. 2023, Art. no. 4700515.

[37] X. Yu, Z. Chen, G. Chen, H. Zhang, and J. Zhou, "A tensor network for tropical cyclone wind speed estimation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 10007–10010.

[38] 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.

[39] S. Dong, Y. Quan, W. Feng, G. Dauphin, L. Gao, and M. Xing, "A pixel cluster CNN and spectral-spatial fusion algorithm for hyperspectral image classification with small-size training samples," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4101–4114, Mar. 2021.

[40] H. Gao, Y. Yang, C. Li, L. Gao, and B. Zhang, "Multiscale residual network with mixed depthwise convolution for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3396–3408, Apr. 2021.

[41] Z. Chen and B. Wang, "Semisupervised spectral-spatial classification of hyperspectral imagery with affinity scoring," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 8, pp. 1710–1714, Aug. 2015.

[42] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[43] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.

[44] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 105–114.

[45] J.-T. Chien and Y.-T. Bao, "Tensor-factorized neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1998–2011, May 2018.

[46] J. Yue, L. Fang, H. Rahmani, and P. Ghamisi, "Self-supervised learning with adaptive distillation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Feb. 2022, Art. no. 5501813.

[47] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1422–1430.

[48] B. Tu, S. Huang, L. Fang, G. Zhang, J. Wang, and B. Zheng, "Hyperspectral image classification via weighted joint nearest neighbor and sparse representation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4063–4075, Nov. 2018.

[49] X. Zhang, Q. Song, R. Liu, W. Wang, and L. Jiao, "Modified co-training with spectral and spatial views for semisupervised hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2044–2055, Jun. 2014.

[50] Z. Zheng, Y. Zhong, A. Ma, and L. Zhang, "FPGA: Fast patch-free global learning network for fully end-to-end hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5612–5626, Aug. 2020.

[51] X. Hu, X. Wang, Y. Zhong, J. Zhao, C. Luo, and L. Wei, "SPNet: A spectral patching network for end-to-end hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 963–966.

[52] J. Zhang, W. Wei, L. Zhang, and Y. Zhang, "Improving hyperspectral image classification with unsupervised knowledge learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 2722–2725.

[53] H.-C. Li, W.-Y. Wang, L. Pan, W. Li, Q. Du, and R. Tao, "Robust capsule network based on maximum correntropy criterion for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 738–751, Feb. 2020.

[54] L. Gao, D. Gu, L. Zhuang, J. Ren, D. Yang, and B. Zhang, "Combining T-distributed stochastic neighbor embedding with convolutional neural networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1368–1372, Aug. 2020.

[55] Z. Tian, Z. Zhang, S. Mei, R. Jiang, S. Wan, and Q. Du, "Discriminative CNN via metric learning for hyperspectral classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 580–583.

[56] P. Sellars, A. I. Aviles-Rivero, and C.-B. Schönlieb, "Superpixel contracted graph-based learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4180–4193, Jun. 2020.

[57] X. Zhang, Y. Wei, H. Yao, Z. Ye, Y. Zhou, and Y. Zhao, "Locally homogeneous covariance matrix representation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9396–9407, Sep. 2021.

[58] W. Hu, H. Li, L. Pan, W. Li, R. Tao, and Q. Du, "Spatial–spectral feature extraction via deep ConvLSTM neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4237–4250, Jun. 2020.

[59] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[60] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[61] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 539–546.

[62] Z. Chen and X. Yu, "A novel tensor network for tropical cyclone intensity estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3226–3243, Apr. 2021.

[63] T. Celik, "Unsupervised change detection in satellite images using principal component analysis and *k*-means clustering," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 772–776, Oct. 2009.

[64] 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.

[65] P. Ramachandran, B. Zoph, and Q. Le, "Swish: A self-gated activation function," *Neural Evol. Comput.*, 2017.

Yahui Xiu is currently working toward the M.S. degree in computer science with the School of Computer Science and Technology, Donghua University, Shanghai, China, supervised by Dr. Zhao Chen.

His research interests include remote sensing image analysis, deep learning, and tensor analysis.

Fuyin Ye received the M.S. degree in computer science from the School of Computer Science and Technology, Donghua University, Shanghai, China, in 2022, supervised by Dr. Zhao Chen.

His research interests include remote sensing image analysis, deep learning, and tensor analysis.

Zhao Chen received the B.S. and Ph.D. degrees in electronic engineering from the School of Information Science and Technology, Fudan University, Shanghai, China, in 2011 and 2016, respectively.

She is currently an Associate Professor with the School of Computer Science and Technology, Donghua University, Shanghai. Her research interests include remote sensing image analysis, medical image processing, pattern recognition, deep learning, and tensor analysis.

Yuxuan Liu is currently working toward the B.S. degree in computer science with the School of Computer Science and Technology, Donghua University, Shanghai, China.

His research interests include image processing and deep learning.