Grafting Transformer on Automatically Designed Convolutional Neural Network for Hyperspectral Image Classification

Xizhe Xue, Haokui Zhang, Bei Fang, Zongwen Bai, and Ying Li

***Abstract*— Hyperspectral image (HSI) classification has been a hot topic for decides, as HSIs have rich spatial and spectral information, and provide a strong basis for distinguishing different land-cover objects. Benefiting from the development of deep learning technologies, deep learning-based HSI classification methods have achieved promising performance. Recently, several neural architecture search (NAS) algorithms have been proposed for HSI classification, which further improves the accuracy of HSI classification to a new level. In this article, NAS and transformer are combined for handling the HSI classification task for the first time. Compared with the previous work, the proposed method has two main differences. First, we revisit the search spaces designed in previous HSI classification NAS methods and propose a novel hybrid search space, consisting of the space-dominated cell and the spectrum-dominated cell. Compared with search spaces proposed in previous works, the proposed hybrid search space is more aligned with the characteristic of HSI data, that is, HSIs have a relatively low spatial resolution and an extremely high spectral resolution. Second, to further improve the classification accuracy, we attempt to graft the emerging transformer module on the automatically designed convolutional neural network (CNN) to add global information to local region focused features learned by CNN. Experimental results on three public HSI datasets show that the proposed method achieves much better performance than comparison approaches, including manually designed networks and NAS-based HSI classification methods. Especially on the most recently captured dataset Houston University, overall accuracy is improved by nearly 6 percentage points. Code is available at https://github.com/Cecilia-xue/HyT-NAS.**

***Index Terms*— Global information, hybrid search space, hyperspectral image (HSI) classification, transformer.**

Manuscript received April 17, 2022; accepted May 26, 2022. Date of publication June 8, 2022; date of current version June 20, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61871460 and Grant 62107027, in part by the Natural Science Foundation of Shaanxi Province under Grant 2020JM-556, and in part by the China Postdoctoral Science Foundation under Grant 2021M692006. *(Xizhe Xue and Haokui Zhang contributed equally to this work.) (Corresponding author: Ying Li.)*

Xizhe Xue and Ying Li are with the School of Computer Science, the National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, and the Shaanxi Provincial Key Laboratory of Speech and Image Information Processing, Northwestern Polytechnical University, Xi'an 710129, China (e-mail: xuexizhe@mail.nwpu.edu.cn; lybyp@nwpu.edu.cn).

Haokui Zhang is with Intellifusion, Shenzhen 518059, China (e-mail: hkzhang1991@mail.nwpu.edu.cn).

Bei Fang is with the Key Laboratory of Modern Teaching Technology, Ministry of Education, Shaanxi Normal University, Xi'an 710119, China (e-mail: beifang@snnu.deu.cn).

Zongwen Bai is with the School of Physics and Electronic Information, Yan'an University, Yan'an 716000, China (e-mail: ydbzw@yau.edu.cn).

Digital Object Identifier 10.1109/TGRS.2022.3180685

I. Introduction

REMOTE sensing observation plays an important role in earth observation and has many applications in agriculture and military [1], [2]. Among various remote sensing observation technologies, hyperspectral image (HSI) classification is a fundamental but essential technique. Captured by the amounts of hyperspectral remote sensing imagers, the HSIs of hundreds of bands contain much richer spectral information than ordinary remote sensing images, and the characteristic of containing both spatial and rich spectral information makes HSIs very useful for distinguishing ground-cover objects. Due to this, the HSI classification technology is widely applied in various scenes, e.g., mineral exploration [3], plant stress detection [4], and environmental science [5]. However, in HSIs, feature vectors containing thousands of bands can be extracted from each spatial pixel location. Such high-dimensional features, on the one hand, help classify the ground objects and, on the other hand, increase difficulty in feature extraction. Therefore, it is worth exploring how to efficiently extract features from HSIs. During past decades, various feature extractions have been applied and designed to extract robust features from HSIs [6]–[8]. Very recently, Luo *et al.* [9] proposed a multistructure unified discriminative embedding (MUDE) method, which overcomes the drawbacks of previous graph-based methods that only consider the individual information of each sample. In MUDE, the neighborhood, tangential, and statistical properties of each sample are introduced by using neighborhood structure graphs. Duan *et al.* [10] considered the manifold structure and multivariate relationship of samples from HSI in their proposed method geodesic-based sparse manifold hypergraph (GSMH). The nonlinear similarity of the distribution of the sample on the manifold space is measured with the geodesic distance to build a manifold neighborhood for each sample. The final method achieves promising performance.

Since 2012, deep learning has been developing rapidly and achieving remarkable results in various fields. Inspired by this, researchers have brought deep learning methods to solve the problem of HSI classification and gained impressive performance. In 2013, Lin *et al.* [11] utilized PCA to reduce the dimensionality of HSI from hundreds of spectral dimensions to dozens and then extract deep features from a neighborhood region via SAE. From 2014 to 2015, Chen *et al.* [12] introduced another spectral dimension channel based on this additional channel directly takes the spectral features

extracted from the pixel to be classified as input, and its output is integrated with the spatial spectrum channels to form a dual-channel structure [12], [13]. In the same period, some other methods tried to apply 1-D- and 2-D-convolutional neural networks (CNNs) in the HSI classification. Specifically, 1-D-CNNs are used to extract deep spectral features [14], [15], and 2-D-CNN are employed to extract deep spatial features from HSI blocks that have been compressed along the spectral dimensions [16], [17]. After 2017, deep HSI classification methods primarily focused on extracting spatial–spectral features. Some works construct a dual-channel network structure to obtain spectral features and spatial features separately and then merge them to form spatial–spectral features [18]. In addition, 3-D-CNN is also a popular choice to capture the spatial–spectral joint features directly [19], [20]. Since 2017, various optimized 3-D-CNN have been applied to the HSI classification task [21], [22], besides which some transfer learning methods have also been drawn into the classification of HSI images [22], [23]

The deep HSI classification methods give full play to the ability to extract robust features independently. These deep HSI classification approaches show a significant advantage in classification performance compared to traditional HSI classification algorithms. However, these deep HSI classification approaches face a problem. Specifically, the network architectures in these methods are manually designed. For deep learning methods, designing an efficient network architecture is difficult, time-consuming, labor-intensive, and requires a lot of verification experiments. This problem is even more serious in HSI classification. Because HSIs data are very different from each other in the number of bands, spectral range, and spatial resolution, the suitable architectures are also different for different HSIs data. Therefore, it is usually necessary to design different network architectures for different HSIs data.

Moving beyond manually designed network architectures, neural architecture search (NAS) techniques [24] seek to automate this process and find not only good architectures but also their associated weights for a given image classification task. NAS provides an ideal solution to liberate people from the heavy work of network architecture design. Chen *et al.* [25] first introduced DARTS into the HSI classification task. This work compressed the spectral dimension of HSIs to tens of dimensions through a pointwise convolution and then directly used DARTS to search for a 2-D-CNN that is suitable for a specific HSI dataset. Later, Zhang *et al.* [26] made an in-depth analysis of the structural characteristics of HSIs and proposed 3-D-ANAS. In their work, a 3-D asymmetric CNN is automatically designed under a pixel-to-pixel classification framework, which overcomes the problem of redundant operation existing in the previous classification framework and significantly improves the model inferring speed.

In this work, further improvements have been made to 3-D-ANAS from two aspects.

1) In 3-D-ANAS, an asymmetric decomposition convolution is introduced in the search space, considering the difference between the spatial resolution and the spectral resolution of the HSI. However, this distinction between space and spectrum is only reflected on the operation level and is not free enough on the search space level. To be more specific, the entire search space consists of a sequence of blocks, each of which contains a number of operations. 3-D-ANAS takes some asymmetric decomposition convolutions and other common convolutions as candidate operations; therefore, 3-D-ANAS can only separably process spatial and spectral information at the operation level. It is difficult for such an approach to incorporate some classic hand-designed experience. For example, in the classical manually designed HSI classification network SSRN [27], the operation is completely separated into spectral processing and spatial processing. Thus, in this article, we have constructed a new and more efficient search space with more freedom to process the differences between spatial and spectral information.
2) The pure CNN structure is good at capturing local information but ignores global information, which has been proven to be very important for a lot of vision tasks. Inspired by this, we attempt to further improve the performance of automatically designed networks by integrating global information through grafting transformer modules. Before classification, we captured the relative relationship of pixels in different spatial positions and used this relationship to fine-tune the spatial–spectral features to achieve better classification accuracy. The main contributions of this work include the following three aspects.

1) By analyzing the characteristics of HSI, we propose an NAS algorithm to automatically design CNN for HSI. Specifically, we proposed a novel hybrid search space, which contains two types of cells, including space-dominated cells and spectral-dominated cells. The entire search space is built on these two cells and can be divided into inner and outer spaces. The inner space determines the topology structure in the cell, and the outer space decides whether the space-dominated cell or the spectral-dominated cell is selected on the specific layer.
2) To further improve the classification accuracy, we attempt to graft the emerging transformer module on the automatically designed CNN to add global information to local region focused features learned by CNN. Benefiting from the pixel-to-pixel classification framework that we adopted here, the transformer module can be seamlessly grafted to the end layer of CNN. Such a grafted structure takes advantage of the ability of a transformer to capture the inner relationship of pixels while avoiding the difficulties of training a complete transformer.
3) Experimental results on three typical HSI classification datasets, including Pavia Center, Pavia University, and Houston University, have validated that the proposed approach obviously improves the classification accuracy of autodesigned HSI classification approaches.

The rest of this article is organized as follows. Section II reviews related work. Our approach is elaborated on in Section III. Section IV provides algorithm implementation

details and extensively evaluates and compares the proposed Hy-NAS and HyT-NAS approaches with state-of-the-art competitors. Finally, we conclude this work in Section V.

II. Related Work

A. Hyperspectral Image Classification via CNNs

Recent years have witnessed growing interest in using CNNs to deal with the HSI classification problem. The development of HSI classification based on CNNs has mainly gone through three stages.

From 2015 to early 2016, researchers focused primarily on HSI classification based on 1-D-CNNs and 2-D-CNNs. The methods based on 1-D-CNNs generally employ 1-D-CNNs to perform convolution along the spectral dimension to extract spectral features [14], [28]. Beyond methods based on 1-D-CNNs, a series of 2-D-CNN-based HSI classification approaches are with good prospects. Intuitively, regions surrounding the pixel can provide additional visual information facilitating the classification. After compressing HSIs to low-dimension, 2-D-CNN-based methods crop a neighborhood patch around the pixel to be classified. Then, this patch is fed to a 2-D-CNN to extract the spatial–spectral features [16], [17]. Compared with the 1-D-CNN-based approaches, the methods based on 2-D-CNNs achieve higher accuracy. However, the classification results of methods that only use 2-D-CNNs may not keep structural information very well. Their visual results are much smoother than those of the 1-D-CNNs methods. The second development stage mainly focuses on combining 1-D-CNN and 2-D-CNN to perform the HSI classification. Taking the advantages of 1-D-CNN and 2-D-CNN, the dual-channel CNN structure can further improve the accuracy of HSI classification [18], [23]. The third stage is the 3-D-CNN stage. Inspired by the 3-D structure of HSIs, 3-D-CNNs have been gradually used in HSI classification approaches. Such methods directly construct 3-D-CNNs to extract the spatial spectrum features. Compared with those of dual-channel CNNs, the structures of 3-D-CNNs are always more simple, intuitive, and powerful [19], [20].

In recent years, optimizing the structures of 3-D-CNNs for HSI classification has become mainstream, for example, the introduction of efficient residual structure [21], lightweight design, and so on [22], [29], [30]. Based on the classical residual structure, Zhong *et al.* [21] integrated the spectral residual and spatial residual modules and then constructed an HSI classification model SSRN based on the two residual modules. Zhang *et al.* [22] developed a lightweight 3-D-CNN to optimize the model structure and proposed two transfer learning strategies (cross-sensor and cross-modality) to handle the problem of small sample [22]. Meng *et al.* [29] proposed a lightweight spectral–spatial convolution HSI classification module (LS2CM) to reduce network parameters and computational complexity.

B. Neural Network Architecture Search

To overcome the heavy burden of manually designing network architecture, researchers turn their attention to NAS, which can automatically and efficiently discover the neural architectures that are suitable for certain tasks. Recent years have witnessed the success of NAS algorithms in plenty of general computer vision tasks, such as image classification [31], object detection [32], and semantic segmentation [33]. So far, the development of NAS always happened in three phases: architecture search based on the evolutionary algorithm (EA), architecture search based on reinforcement learning (RL), and architecture search based on gradient. RL-based methods [31], [34] often contain a recurrent neural network (RNN) to perform as a metacontroller, generating potential architectures. In the NAS methods enlightened by EA algorithms [35]–[37], a series of randomly constructed models are evolved into a better architecture through EA. However, most RL methods and EA methods suffer from heavy computational costs and are less efficient in the searching stage. The gradient-based NAS methods are proposed recently and can alleviate this problem to some extent. The first attempt DARTS is proposed in [24]. Unlike the EA and RL-based methods that train plenty of student networks, DARTS merely trains one super network in the searching phase, reducing the training workload significantly. Getting inspiration from DARTS, Chen *et al.* [25] proposed a 3-D Auto-CNN for HSI classification. In the preprocessing stage, 3-D Auto-CNN heavily compresses the spectral dimension of raw HSIs through pointwise convolution. The search space of 3-D Auto-CNN is made up of 2-D convolution operations in fact.

Very recently, Zhang *et al.* [26] proposed 3-D-ANAS, in which the pixel-to-pixel classification framework and the 3-D hierarchical search space are jointly used. In conventional patch-to-pixel classification frameworks, all information in a cropped patch is used to classify a single pixel. In a pixel-to-pixel framework, all pixels in a cropped patch are classified in one iteration. Adopting a pixel-to-pixel classification framework reduces repeat operations, speeding up inference efficiency significantly. In the 3-D hierarchical search space, all operations are in the 3-D structure, and the widths of networks can be adjusted adaptively in this work according to the characteristics of different HSIs. Benefiting from these two points, 3-D-ANAS achieves promising performance. Unfortunately, 3-D-ANAS still has two shortcomings.

1) Previous work has indicated that learning the spectral and spatial representations separately is beneficial to extracting more discriminative features, such as SSRN. Although various asymmetric convolutions in the search space of 3-D-ANAS allow the fine-tuning of the convolution kernel size and receptive field along spectral and spatial dimensions, this adjustment is limited inside a cell. Adjusting the proportions of spectral and spatial convolutions across the entire network is infeasible in this framework.
2) The pure convolutional structure mainly focuses on local neighborhood information while ignoring the global relationship information among the whole input patch, which is often critical for the high-level classification task.

To overcome these two issues mentioned above, we propose a new NAS method for HSI classification. Specifically,

to address the first issue, we design a hybrid search space that consists of two kinds of cells. One is a space-dominated cell, and another is a spectrum-dominated cell. The hybrid search space has more flexible structures in selecting spatial or spectral convolution than the search space proposed in 3-D-ANAS. Aiming to solve the second problem, a light transformer structure is grafted to the end of CNN, playing a similar role as a CRF to dig out the relationship between pixels.

C. Vision Transformer

By in-depth analysis of the attention mechanism, Vaswani *et al.* [38] proposed the transformer model. Compared with the RNN model previously applied to the NLP problem, the transformer improves the computational efficiency significantly because its structure can handle the elements in a sequence in parallel. Besides, the transformer inherits and further expands the ability to capture the relationship between elements in the sequence, in comprehension with RNN. As a result, the introduction of the transformer has greatly promoted the development of NLP fields.

In recent years, transformer models have been adopted in image processing and achieved very promising performance. Dosovitskiy *et al.* [39] proposed ViT, where the image is cut into patches, and then, the patches are arranged into the input sequence for feature extraction. In order to keep sensitive to the position information of the patches, position embedding is introduced in the ViT. Besides, an additional class token is designed to perform the final classification. ViT's success in the fundamental visual tasks has greatly inspired the field of CV. Although the performance of ViT is relatively good, there still exist some problems; for instance, ViT has low computational efficiency and is hard to train. To alleviate the problem that the ViT is hard to train, Touvron *et al.* [40] proposed to use knowledge distillation to train ViT models and achieved competitive accuracy with the less pretraining data. From the perspective of reducing computational cost and improving inference speed, Liu *et al.* [41] proposed the Swin transformer. The Swin transformer limits the calculation of attention to pixels within a small window, which reduces the amount of calculation. Moreover, a shifted window-based MSA is proposed, which makes the attention cross different windows. The Swin transformer has achieved higher accuracy than previous CNN models on tasks such as dense prediction. Very recently, after conducting a detailed analysis of the working principle of CNN and transformer, Graham *et al.* [42] mixed CNN and transformer in their LeVit model, which significantly outperforms previous CNNs and ViT models with respect to the speed/accuracy tradeoff.

Very recently, there are several methods adopting transformer models to classify HSIs [43]–[45]. He *et al.* [43] designed a spatial–spectral transformer, where a CNN is used to capture spatial information and a ViT is introduced to extract spectral relationship. Similarly, two parallel works also adopt transformers to extract the spectral relationship. The network proposed in [44] starts with a spectral relationship extraction transformer and ends with several decoders. In SpectralFormer [45], a sequence of patches extracted from the input HSI is fed into the transformer.

Relevant to fusing the strength of CNN and the transformer model, our work is closely related to Levit. The difference is that the main body of our network still relies on an automatically designed CNN. In Levit, the transformer part is also the main part of feature extraction. The structure of the high-level CNN is equivalently replaced with the transformer structure. In our work, the transformer model is just to further capture the spatial relationship based on the features extracted by CNN. Compared to works that also adopt transformer models, our proposed HyT-NAS is a hybrid structure, which inherits the advantages of NAS and the strengths of the transformer. Such structure makes it more stable and easier to train while gaining better performance. For example, on the Houston University dataset, with only 450 training samples, our proposed HyT-NAS achieves 91.14% overall accuracy, which is 3.13 percentage points higher than 88.01%, and the overall accuracy of SpectralFormer is trained with 2823 training samples. In fact, such a phenomenon is consistent with the discovery presented in recent research works [42], [46]–[48]. Hybrid structures generally achieve better performance than pure CNNs or ViTs as hybrid structures combine both advantages of CNN and ViT. In addition, previous transformer structures adopted in HSI classification methods always focus on capturing spectral relationships, while, in our work, the transformer is responsible for capturing the relationship from all input space.

III. Proposed Method

In this section, the proposed method is introduced in detail. First, as the proposed method contains more steps than previous deep learning-based HSI classification approaches, we introduce the overall workflow briefly. Next, we elaborate on the proposed hybrid search space and compare it with the search space proposed in 3-D-ANAS [26]. Then, we explain the reason for grafting the transformer module to the searched CNN and present the architecture of the grafted transformer module. Finally, we will briefly introduce our training process.

A. Overall Workflow

As shown in Fig. 1, the workflow of the overall classification framework can be divided into the following steps.

1) Samples Extraction: Some pixels are randomly extracted from the whole HSIs according to certain proportions and rules. The collected sample pixels are divided into the training set and the validation set. The rest are reserved as the test set.

2) Searching: The collected training samples are fed into the CNN super network stacked by the space-dominated cell and spectrum-dominated cell. The training loss aims to minimize the loss between the prediction label and the ground truth. The prediction accuracy of the network is validated on the validation set at a certain interval, and the loss and verification accuracy are recorded.

3) Deducing the Final Network and Grafting Transformer: The weight of the super network model with the highest validation accuracy is used to deduce the final component

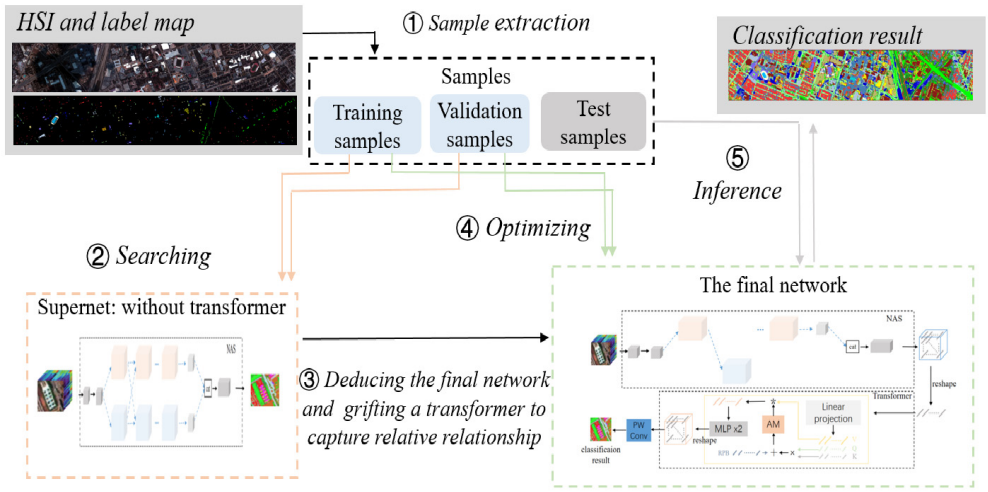

Fig. 1. Workflow of the proposed method.

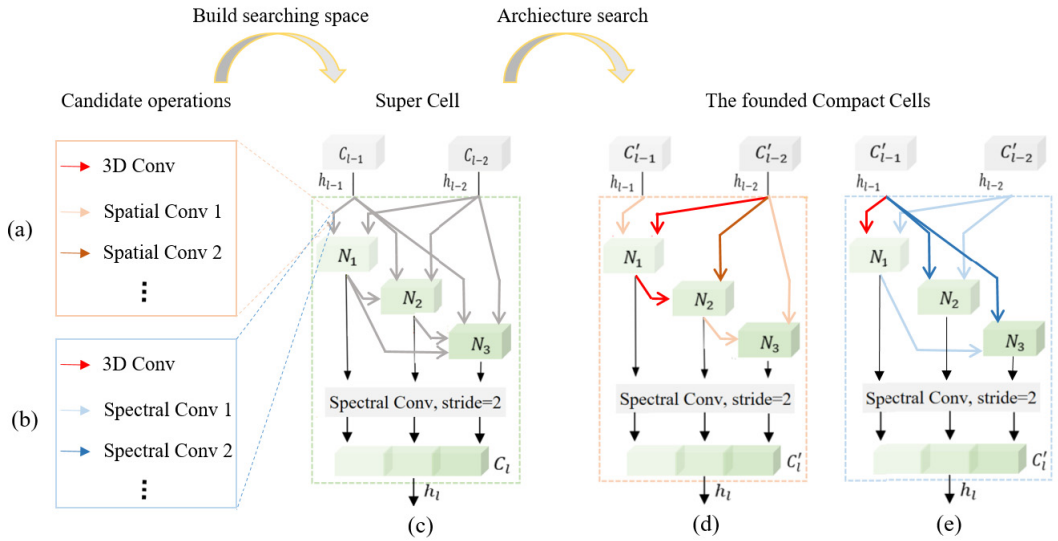

Fig. 2. Architectures of supercells and the founded compact cells. (a) Candidate operations in the space-dominated supercell. (b) Candidate operations in the spectrum-dominated supercell. (c) Super Cell. (d) Founded space-dominated compact cell. (e) Founded spectrum-dominated compact cell. In (a) and (b), different types of possible operations are represented as arrows with different colors. The topological architecture searching strategy aims to find a compact cell, in which each node only keeps the two most valuable inputs, each of which is fed to the selected operation. For a more clear and concise presentation, we only show three nodes in the cell and three convolution operations in the search space.

network. According to the weight of the search model, the type of cell and the topology inside the cell are fixed in each layer. Besides, a flexible transformer structure is grafted at the end of the CNN network to capture the relationship between pixels.

4) Optimizing the Final Compact Network: The training set is taken to optimize the grafted CNN-transformer network structure, using the same loss as the searching stage.

5) Inference: After training, the compact model with the highest verification accuracy and the smallest loss is tested on the test set.

B. Hybrid Search Space

1) Cell Structure: In 3-D-ANAS, the authors have already noticed that processing spatial and spectral information separately has a better performance than using 3-D convolution. In their work, classification accuracy is improved by introducing asymmetric search space. In this work, we further extend this discovery and propose a hybrid search space, which consists of space-dominated cells and spectrum-dominated cells. As shown in Fig. 2, the space-dominated cell only contains spatial convolutions and 3-D convolutions (instead of using the standard 3-D convolution, we adopt separable 3-D convolution as it has fewer parameters; in the following paragraphs, we call separable 3-D convolution as 3-D convolution for short), and the spectrum-dominated cell includes some spectral convolutions and 3-D convolutions. After searching, each layer can only keep one space cell or spectrum cell, and different layers do not share the cell structure.

Compared with the search space proposed in 3-D-ANAS, our designed hybrid search space is more flexible in selecting different operations to process spatial and spectral information.

Note that this is relatively important for HSI datasets, as HSI datasets have a special characteristic, that is, HSI datasets have different relatively low spatial resolutions and extremely high spectral resolutions. Roughly processing spatial and spectral information usually generates inferior classification accuracy.

In specific, the space-dominated cell includes the following operations.

1) *acon_3-1:* LReLU-Conv($1 \times 3 \times 3$)-BN.
2) *acon_5-1:* LReLU-Conv($1 \times 5 \times 5$)-BN.
3) *asep_3-1:* LReLU-Sep($1 \times 3 \times 3$)-BN.
4) *asep_5-1:* LReLU-Sep($1 \times 5 \times 5$)-BN.
5) *con_3-3:* LReLU-Conv($1 \times 3 \times 3$)-Conv($3 \times 1 \times 1$)-BN.
6) *con_3-5:* LReLU-Conv($1 \times 3 \times 3$)-Conv($5 \times 1 \times 1$)-BN.
7) *skip_connection:* $f(x) = x$.
8) *discarding:* $f(x) = 0$.

The spectrum-dominated cell includes the following operations.

1) *econ_3-1:* LReLU-Conv($3 \times 1 \times 1$)-BN.
2) *econ_5-1:* LReLU-Conv($3 \times 1 \times 1$)-BN.
3) *esep_3-1:* LReLU-Sep($3 \times 1 \times 1$)-BN.
4) *esep_5-1:* LReLU-Sep($5 \times 1 \times 1$)-BN.
5) *con_3-3:* LReLU-Conv($1 \times 3 \times 3$)-Conv($3 \times 1 \times 1$)-BN.
6) *con_3-5:* LReLU-Conv($1 \times 3 \times 3$)-Conv($5 \times 1 \times 1$)-BN.
7) *skip_connection:* $f(x) = x$.
8) *Discarding:* $f(x) = 0$.

Here, LReLU, BN, Conv, and Sep represent the LeakyReLU activation function, batch normalization, common convolution, and separable convolution.

2) Architecture Searching Strategy: The network architecture search process can be divided into inner and outer search parts. The outer search part determines the cell type of this layer, and the inner search strategy decides the cell's internal topology structure. The finally searched L-layer network may contain L different cell structures, and every cell contains a sequence of N nodes.

The inputs for each node consist of the outputs of all previous nodes and two inputs of the current cell. Assuming that each path in a cell contains all the P candidate operations, the output of node x_i is

$$x_i = \sum_{j=1}^{j=P} \left(\omega_i^j \cdot o_i^j \right) \tag{1}$$

where o and ω represent the different convolution operations and their corresponding weights, respectively. This weight is learned through inner search according to the backpropagation gradient. The output of a cell h_l^k is obtained by

$$h_l^k = \text{concat}\left(x_i^k \mid i \in \{1, 2, \ldots, N\} \right) \tag{2}$$

where k denotes the cell type and l represents the layer number.

When optimizing the internal topology of a cell, the outer selection on cell types is also ongoing. Specifically, two types of cells are provided for each layer, focusing on spatial information and spectral information, respectively. In each layer, the outputs corresponding to two types of cells are weighted via learnable weights α_i and β_i and then combined to form the cell output h_l^k. The output of layer l can be expressed as

$$h_l = \text{concat}\left(\alpha_l \cdot h_l^{spa} + \beta_l \cdot h_l^{spe} \right). \tag{3}$$

After the stage of searching for network architectures, we build a compact network according to the learned structure parameters ω, α, and β. Specifically, for the inner topology structure, we keep the two operations corresponding to the top two weights and prune the rest in each cell. For outer structure, we compare α and β and then reserve the cell whose weight is bigger.

C. Structure of Transformer

So far, the proposed architecture has been a pure convolution structure. The final compact network founded by Hy-NAS is also a pure CNN. The Hy-NAS algorithm not only improves the external structure but also reserves the inductive bias of convolution operations. In other words, the final compact network founded by Hy-NAS also inherits the disadvantage of pure CNN. Pure CNN is good at extracting local features and ignores global relationships. Adding global information to local features always leads to much better performance, which has been verified by previous nonlocal related works [49] and the emerging transformer models [40]. Especially, for some dense prediction tasks, using global information may bring a significant improvement [50].

Therefore, to further improve the performance of Hy-NAS, we attempt to integrate global information with features learned by pure CNN. A natural idea is to add some nonlocal modules into the search space. As Hy-NAS is an NAS algorithm, adding nonlocal modules into search space does integrate information, but it also increases the complexity of search space and the difficulty of training super net. Here, we make a tradeoff. Specifically, we graft a flexible transformer module at the end of the final compact network founded by Hy-NAS. Finally, we obtain a new HSI classification method, HyT-NAS.

Such a grafting operation integrates global information to features learned by CNN while avoiding introducing a complex search space which may improve the workload of architecture search. Inspired by the promising performance of transformer models, we choose to graft a transformer module to integrate global information, as shown in the bottom half of Fig. 3.

Before being split to sequence and sent to the transformer unit, the feature map f of size(B, C, W, H) from the encoder is reshaped and transposed to $f \in (B, N, C)$. Q, K, and V are calculated through a linear layer and batch normalization layer (refers to linear projection in Fig. 3). Here, linear projection is responsible for mapping input vectors to three different feature spaces Q, K, and V, which plays different roles in the following computational procedure. The definitions and functions of Q, K, and V can be found in [38]. f is the input of the attention layer, and f_{attn} is computed according to the following equation:

$$f_{\text{attn}} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + P \right) V \tag{4}$$

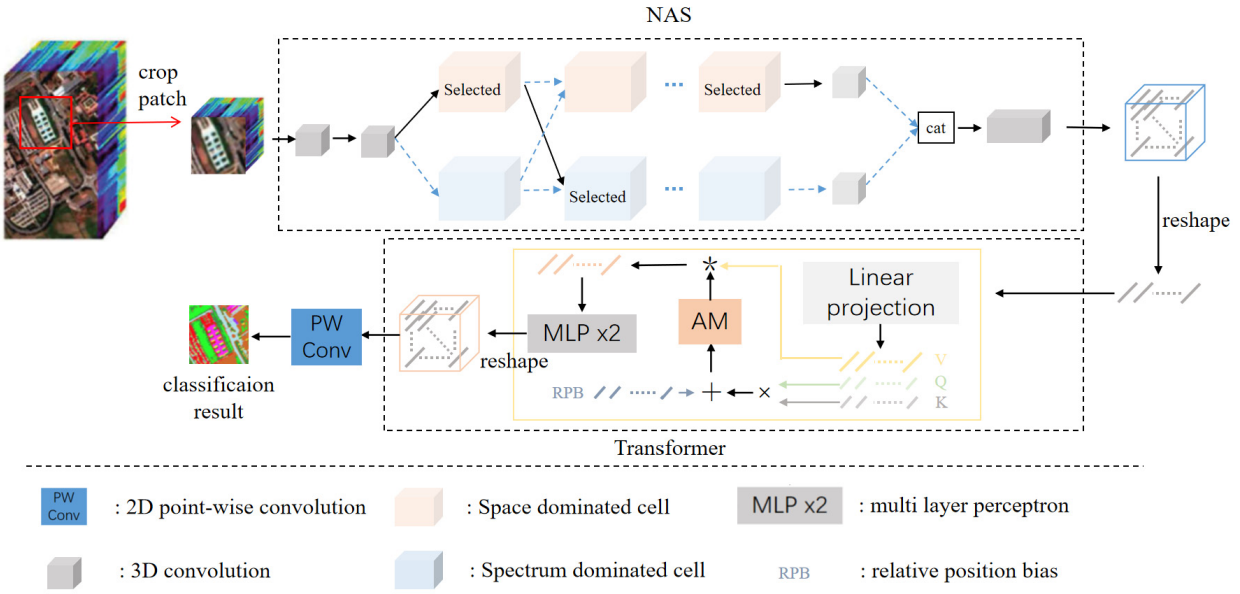

Fig. 3. Architecture of our final classification framework. (Top) Searched compact network. The final selected blocks are marked as Selected. (Bottom) Structure of the grafted transformer.

where d_k denotes the dimensionality of V, and P means the relative position embedding (RPB)

$$P^h_{(x,y),(x',y')} = Q_{(x,y),:} \cdot K_{(x',y'),:} + B^h_{|x-x'|,|y-y'|} \tag{5}$$

where B^h represents the translation-invariant attention bias. The output f_{out} of the transformer can be generated as (6) and then reshaped to the same dimensionality as the input f

$$f_{\text{out}} = \text{MLP}(\text{MLP}(AF(BN(f_{\text{att}})) + f)) \tag{6}$$

in which MLP and BN denote multilayer perception and the batch normalization layer, respectively. AF means the activation function. Specifically, a Hardswish function is employed in this work.

D. Training Process

In this work, we have followed the pixel-to-pixel classification framework of 3-D-ANAS. Therefore, to fairly verify the effectiveness of the proposed contributions, we apply the same sampling rules, searching, and training strategies as those in 3-D-ANAS. After taking a 3-D image cube from raw HSI and predicting the class of each 2-D position in the cube, the cross-entropy loss has been calculated according to the sparse training label map.

IV. Experiments

Experiments are conducted on a server with an Intel[1] Xeon[1] Gold 6230 CPU @ 2.10 GHz, 512 GB of memory, and Nvidia Tesla V100 32-GB graphics card. The training and testing experiments were implemented by using the open-source framework Pytorch 1.8.[2]

[1]Registered trademark.
[2]https://pytorch.org/docs/1.8.0/

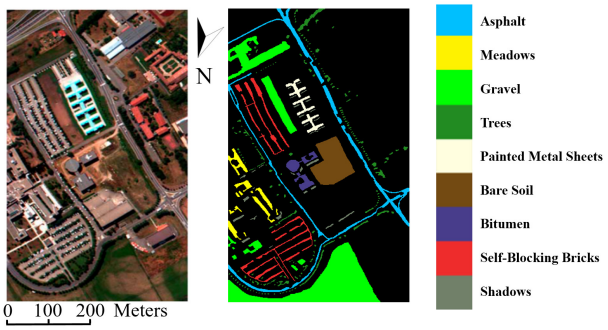

Fig. 4. False color composites and ground-truth maps of Pavia University.

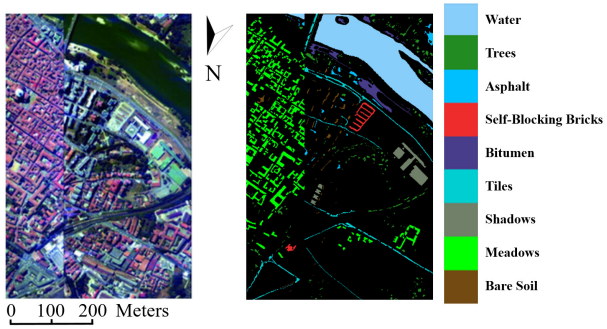

Fig. 5. False color composites and ground-truth maps of Pavia Center.

A. Data Description

To evaluate the effectiveness of the proposed NAS algorithm, we conduct comparison experiments on three representative HSI datasets, namely, Pavia University, Pavia Center, and Houston University. In turn, the false color composites and ground-truth maps of these three HSIs are presented in Figs. 4–6. The corresponding sample distribution information is listed in Table I.

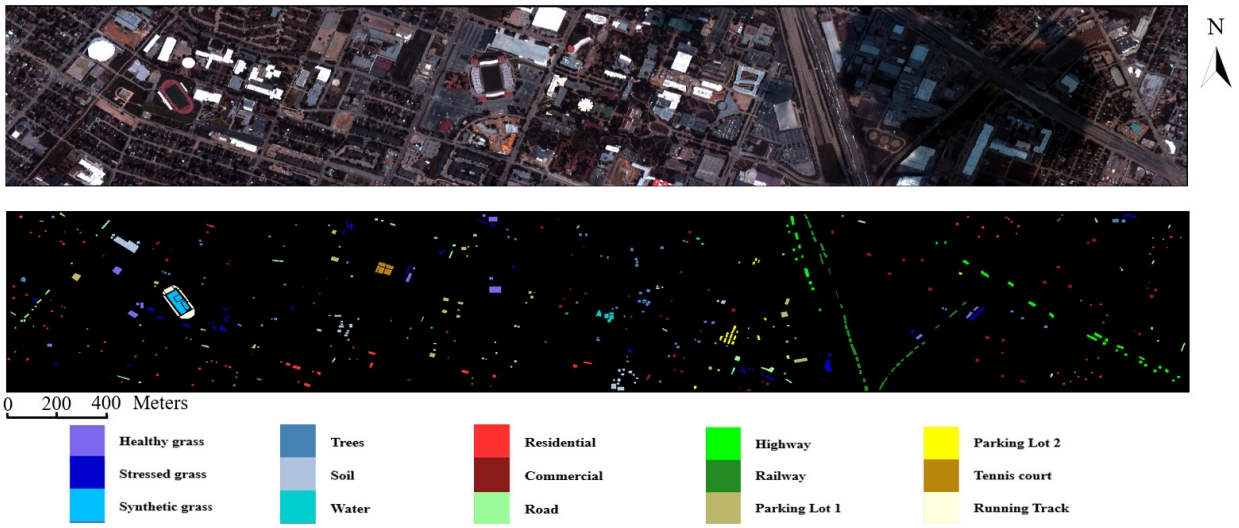

Fig. 6. False color composites and ground-truth maps of Houston University.

TABLE I
SAMPLE DISTRIBUTION INFORMATION OF DATASETS

Pavia University			Pavia Center		Houston University	
Class	Land Cover Type	No.of Samples	Land Cover Type	No.of Samples	Land Cover Type	No.of Samples
1	Asphalt	6631	Water	824	Healthy Grass	1251
2	Meadows	18649	Trees	820	Stressed Grass	1254
3	Gravel	2099	Asphalt	816	Synthetic Grass	697
4	Trees	3064	Self-Blocking Bricks	808	Trees	1244
5	Painted Metal Sheets	1345	Bitumen	808	Soil	1242
6	Bare Soil	5029	Tiles	1260	Water	325
7	Bitumen	1330	Shadows	476	Residential	1268
8	Self-Blocking Bricks	3682	Meadows	824	Commercial	1244
9	Shadows	947	Bare Soil	820	Road	1252
10	-	-	-	-	Highway	1227
11	-	-	-	-	Railway	1235
12	-	-	-	-	Parking Lot 1	1233
13	-	-	-	-	Parking Lot 2	469
14	-	-	-	-	Tennis Court	428
15	-	-	-	-	Running Track	660
	Total	42776	Total	7456	Total	15029

Pavia University and Pavia Center were captured by the ROSIS-3 sensor in 2001 during a flight campaign over Pavia, Northern Italy. Due to low SNR, some frequency bands were removed. The remaining 103 channels are used for classification. These datasets have the same geometric resolution, that is, 1.3 m. Each dataset covers nine different land cover categories. Part of the categories is overlapped. Please find more details in Fig. 4 and 5. Pavia University consists of 610×340 pixels, and Pavia Center covers 1096×715 pixels.

Houston University was captured by the ITRES-CASI 1500 hyperspectral Imager over the University of Houston campus and the neighboring urban area. Compared with the aforementioned two datasets, Houston University has lower spatial resolution but much higher spectral resolution. Its spatial resolution is 2.5 m, and it contains 144 spectral bands, covering the wavelength range of 360–1050 μm. This dataset also covers a wider area and more abundant land cover objects. In specific, the Houston University dataset consists of 349×1905 pixels and includes 15 land-cover classes of interest.

B. Experiment Design

In order to validate the effectiveness of the proposed algorithm, we conduct experiments in two different settings. In setting one, 20 and ten labeled pixels are randomly extracted from each class to build a training set and a validation set. The rest is used as a test set. In setting two, the number of training samples for each class is increased to 30. Others keep the same with that in setting one. More details about the sample distribution are listed in Table II. To ensure the fairness and stability of the comparison, we repeat each experiment five times and take the average values as the final results.

C. Implementation Details

Similar to 3-D-ANAS [26], the proposed method also has two optimizing stages and one inference stage. In this section, we introduce the different settings in the aforementioned stages on three different datasets. For brevity, the settings

TABLE II

DISTRIBUTION INFORMATION OF TRAINING, VALIDATION, AND TEST SETS

Setting	Dataset	Training	Validation	Test	Training%
20 pixels/class	Pavia U	180	90	42506	0.42%
	Pavia C	180	90	7186	2.41%
	Houston U	300	150	14579	2.00%
30 pixels/class	Pavia U	270	90	42416	0.63%
	Pavia C	270	90	7096	3.62%
	Houston U	450	150	14429	2.99%

that are consistent with the baseline 3-D-ANAS would not be mentioned here.

1) Searching: For three different datasets, we construct three different super nets that share the same outline structure. Specifically, in the outer structure, each super net consists of four layers of supercells, and each layer is made up of two different supercells: the space-dominated supercell and the spectrum-dominated supercell. In the inner structure, each cell has a sequence of three nodes. The entire searching process is carried out on an Nvidia V100 card with 32-GB memory. For Pavia University and Pavia Center, we crop the patches with a spatial resolution of 24×24 as searching samples, and the batch size is set to 6. In Houston University, the crop size of patches is set to 14×14, and the batch size is 5. On all three datasets, the Adam optimizer with both learning rate and weight attenuation of 0.001 is used to optimize the architecture parameters (α, β, and ω). The standard SGD optimizer is applied to update the super net parameters (learnable kernels in candidate operations), where momentum and weight decay are set to 0.9 and 0.0003, respectively. The learning rate decays from 0.025 to 0.001 according to the cosine annealing strategy. For Pavia University and Pavia Center, the first 15 epochs are the warm-up stage, in which we only optimize super net parameters. Because Houston University is more challenging, we set 30 epochs for warming up. After the warming-up stage, we alternately update the architecture parameters and super network parameters in each iteration.

2) Grafted Network Optimization: We crop patches with spatial resolution 32×32 to train the final grafted network. Random cropping, flipping, and rotation are introduced as data enhancement strategies. Batch sizes on Pavia University and Pavia Center are set to 12. The batch size on Houston University is set to 16. At this stage, we use the SGD optimizer. The initial learning rate is set to 0.1, decayed according to the poly learning rate policy with power of 0.9 ($\text{lr} = \text{init\_lr} \cdot (1 - (\text{iter}/\text{max\_iter})^{\text{power}})$). The performance of the network is validated every 100 iterations.

3) Inference: For the grafted framework based on hybrid CNN and transformer, we introduced an overlap inference (OV) strategy to further improve the performance. Specifically, we use a sliding window to crop small blocks (the stride is half of the window size) and input the cropped blocks into the compact network. The average result of the overlapping area is considered the final prediction result. As the number of tokens in our transformer module is fixed, the multiscale verification method [multi-scale inference (MS)] is not adopted here, while using the OV strategy alone already achieves promising performance. The structure that we designed requires the input sequence to be a fixed length. Therefore, the image blocks should be on the same scale during training and verification. Relaxing this restriction is considered one of our future work.

TABLE III

COMPARISON EXPERIMENTAL RESULTS ON PAVIA UNIVERSITY USING 20 TRAINING SAMPLES EACH CLASS

Models	3D-LWNet	SSRN	1-D Auto-CNN	3-D Auto-CNN	3D-ANAS†	Hy-NAS	HyT-NAS	HyT-NAS +OV
1	82.43	99.54	69.69	88.24	92.72	97.77	**98.88**	98.79
2	84.76	99.31	76.37	90.72	96.18	98.88	97.86	**99.16**
3	76.88	94.36	73.43	92.11	97.32	98.07	**98.44**	98.12
4	91.45	94.95	90.2	81.27	95.49	95.81	97.45	**98.06**
5	96.23	99.25	96.54	93.12	**100**	99.92	99.31	99.77
6	92.50	71.76	75.48	98.47	96.89	94.48	**99.56**	98.92
7	93.56	73.64	88.83	96.14	97.19	99.92	**100**	**100**
8	96.01	86.27	77.59	**96.84**	93.64	92.03	95.19	96.44
9	89.16	98.72	96.66	79.62	**100**	**100**	**100**	**100**
OA	87.10	91.99	77.65	91.16	95.74	97.43	98.03	**98.77**
AA	89.22	90.87	82.75	90.72	96.60	97.43	98.41	**98.81**
K	83.35	89.60	71.51	88.51	94.37	96.59	97.39	**98.37**

TABLE IV

COMPARISON EXPERIMENTAL RESULTS ON PAVIA UNIVERSITY USING 30 TRAINING SAMPLES EACH CLASS

Models	3D-LWNet	SSRN	1-D Auto-CNN	3-D Auto-CNN	3D-ANAS†	Hy-NAS	HyT-NAS	HyT-NAS +OV
1	82.32	99.90	76.43	89.70	95.24	94.43	**99.07**	99.03
2	88.84	99.53	81.9	97.92	98.16	99.74	99.26	**99.80**
3	83.74	88.26	74.04	92.31	97.44	99.32	99.17	**99.81**
4	94.11	93.16	93.38	71.22	98.52	**98.52**	97.45	97.82
5	96.90	100	96.13	95.12	99.77	**100**	**100**	**100**
6	92.10	66.97	81.81	96.96	99.46	**100**	99.74	**100**
7	95.46	99.69	88.23	95.99	99.92	**100**	**100**	99.92
8	93.19	89.77	74.05	94.98	98.85	96.19	98.90	**99.18**
9	92.43	99.24	95.65	80.64	99.89	**100**	**100**	**100**
OA	89.25	91.95	81.75	93.36	98.05	98.55	99.18	**99.52**
AA	91.01	92.95	84.62	90.54	98.58	98.69	99.29	**99.51**
K	86.05	89.59	76.55	91.50	97.42	98.08	98.92	**99.37**

D. Comparison With State-of-the-Art Methods

In this section, we compare the proposed Hy-NAS and HyT-NAS with other four recent CNN-based HSI classification methods. The codes for all comparison methods are derived from the official codes: 3-D-LWNet,[3] 1-D Auto-CNN and 3-D Auto-CNN,[4] and 3-D-ANAS.[5] Tables III–VIII list the results of the comparative experiment, and Figs. 7–9 show the corresponding visual results. Here, we do not compare the inference speeds, as these methods are implemented with different frameworks, which may introduce biases. The inference of the pixel-to-pixel framework has higher efficiency than that of the patch-to-pixel framework because the former framework does not have repeat operations, as explained in [26]. The method

[3]https://github.com/hkzhang91/LWNet

[4]https://github.com/YushiChen/Auto-CNN-HSI-Classification

[5]https://github.com/hkzhang91/3D-ANAS

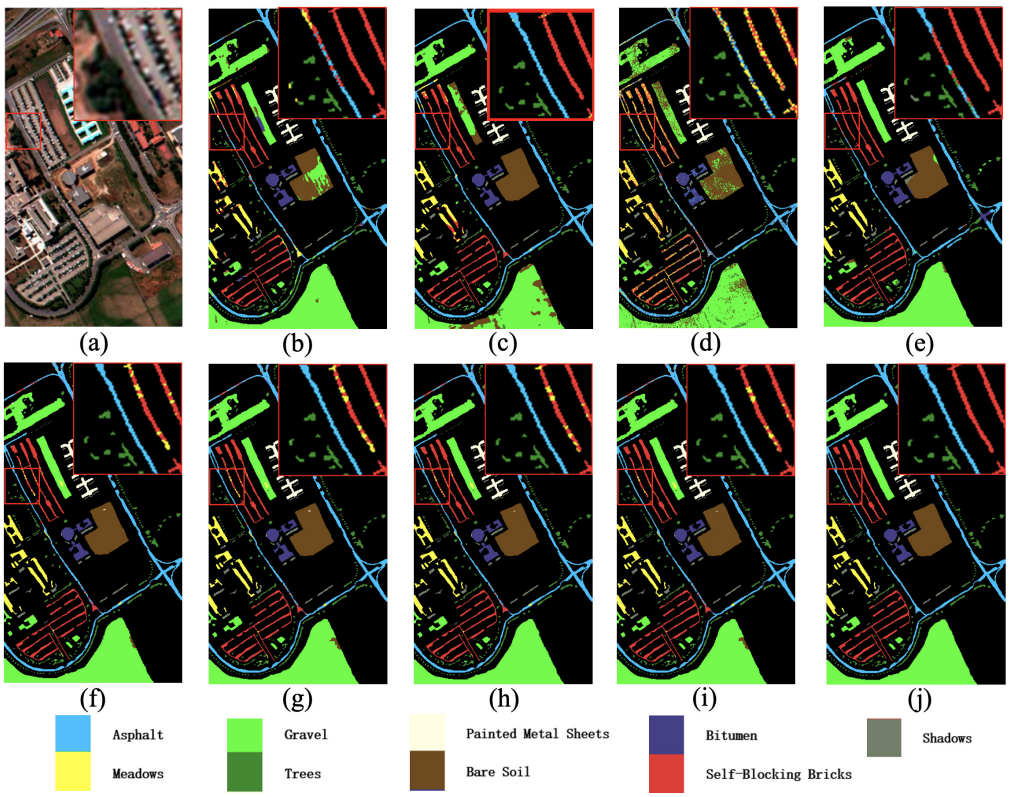

Fig. 7. Comparison experimental results on Pavia University using 30 training samples in each class. (a) False color composite. (b) 3-D-LWNet, OA = 89.25%. (c) SSRN, OA = 91.95. (d) 1-D Auto-CNN, OA = 81.75%. (e) 3-D Auto-CNN, OA = 93.36%. (f) 3-D-ANAS, OA = 98.05%. (g) Hy-NAS, OA = 98.55%. (h) HyT-NAS, OA = 99.18%. (i) HyT-NAS + OV, OA = 99.52%. (j) Ground-truth map.

TABLE V
COMPARISON EXPERIMENTAL RESULTS ON PAVIA CENTER USING 20 TRAINING SAMPLES EACH CLASS

Models	3D-LWNet	SSRN	1-D Auto-CNN	3-D Auto-CNN	3D-ANAS†	Hy-NAS	HyT-NAS	HyT-NAS +OV
1	99.61	**100**	99.81	99.56	99.61	99.62	99.71	99.69
2	91.85	98.68	80.9	87.79	93.16	95.18	**96.61**	96.43
3	86.77	87.39	89.04	82.98	**96.67**	93.43	89.77	89.28
4	92.85	87.79	73.53	98.85	99.81	96.80	**100**	**100**
5	95.53	99.75	90.05	95.83	98.14	99.80	99.99	**100**
6	80.97	88.58	95.53	93.22	98.70	**99.23**	97.06	97.97
7	85.66	99.28	84.23	94.94	94.72	95.48	98.85	**99.21**
8	85.06	**100**	98.17	95.12	99.89	99.77	99.95	99.96
9	91.34	97.79	98.89	85.29	99.82	**100**	99.75	99.89
OA	92.42	98.51	96.18	96.25	98.95	99.05	99.23	**99.28**
AA	89.96	95.47	90.02	92.62	97.84	97.70	97.98	**98.05**
K	89.41	97.89	94.6	94.71	98.51	98.65	98.81	**98.98**

TABLE VI
COMPARISON EXPERIMENTAL RESULTS ON PAVIA CENTER USING 30 TRAINING SAMPLES EACH CLASS

Models	3D-LWNet	SSRN	1-D Auto-CNN	3-D Auto-CNN	3D-ANAS†	Hy-NAS	HyT-NAS	HyT-NAS +OV
1	99.52	100	99.76	99.78	**100.0**	**100.0**	99.99	**100.0**
2	93.92	99.34	87.9	92.48	95.83	96.75	96.40	**96.98**
3	89.26	86.05	91.16	85.81	93.34	95.28	**96.43**	96.20
4	91.1	79.70	79.72	97.32	97.77	99.85	99.96	**100.0**
5	96.09	99.93	92.1	97.02	98.36	96.82	99.95	**99.97**
6	90.73	93.62	96.47	95.91	99.57	98.97	99.69	**99.90**
7	93.24	99.51	85.43	95.29	97.97	97.92	**98.65**	98.58
8	87.32	**99.99**	97.88	95.13	99.40	99.69	99.27	99.31
9	93.7	99.23	98.58	91.84	**100.0**	99.89	99.79	**100.0**
OA	94.22	98.72	96.79	96.99	99.24	99.33	99.44	**99.49**
AA	92.76	95.26	92.11	94.51	98.03	98.34	98.90	**98.99**
K	91.92	98.18	95.47	95.76	98.92	99.05	99.20	**99.28**

proposed in this article adopts a pixel-to-pixel framework and inherits the advantage in inference efficiency.

The performance on Pavia University is listed in Tables III and IV. The corresponding visual comparison results are shown in Fig. 7. From the comparison results, we can draw the following conclusions.

1) Compared with the method based on 1-D-CNN, the method based on 3-D-CNN usually gains better performance because jointly using the spectral and spatial information is beneficial to improve the classification accuracy. Compared to 3-D-ANAS,[6] the proposed method adopts hybrid search space, which improves the flexibility in processing spectral and spatial with different operations, achieving higher classification accuracy.

[6]For fairness, we rerun the code of 3-D-ANAS on the same device (with Nvidia V100 GPUs) with the proposed methods and reported the results as 3-D-ANAS†. For 3-D-ANAS†, the performance is a little bit different from that in the original paper; we conjecture that this is caused by the reimplementation on different devices.

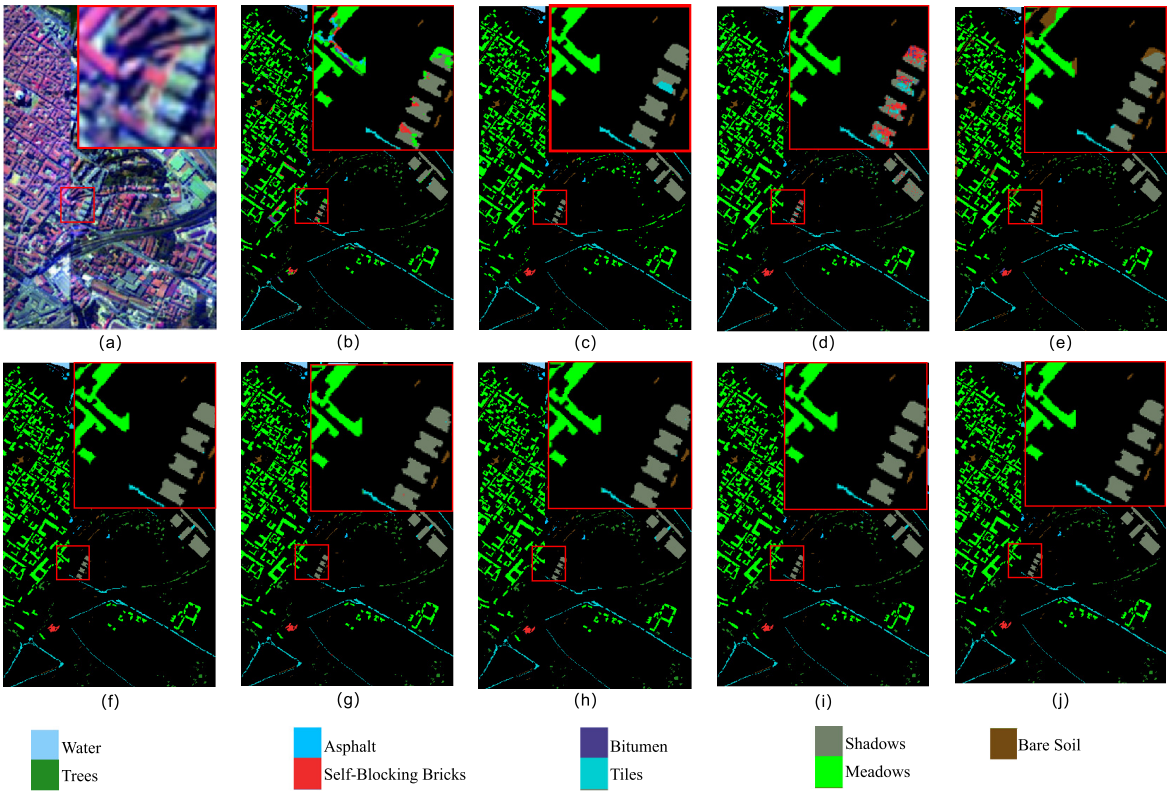

Fig. 8. Comparison experimental results on Pavia Center using 30 training samples each class. (a) False color composite. (b) 3-D-LWNet, OA = 94.22%. (c) SSRN, OA = 98.72. (d) 1-D Auto-CNN, OA = 96.79%. (e) 3-D Auto-CNN, OA = 96.99%. (f) 3-D-ANAS, OA = 99.24%. (g) Hy-NAS, OA = 99.33%. (h) HyT-NAS, OA = 99.44%. (i) HyT-NAS + OV, OA = 99.49%. (j) Ground-truth map.

TABLE VII

COMPARISON EXPERIMENTAL RESULTS ON HOUSTON UNIVERSITY USING 20 TRAINING SAMPLES EACH CLASS

Models	3D-LWNet	SSRN	1-D Auto-CNN	3-D Auto-CNN	3D-ANAS†	Hy-NAS	HyT-NAS	HyT-NAS +OV
1	79.16	69.69	69.98	84.01	77.56	83.62	**87.37**	86.00
2	71.61	96.81	60.19	85.65	82.27	79.66	88.30	**88.48**
3	94.4	97.81	77.30	93.86	86.96	82.01	**87.67**	86.96
4	74.25	85.79	49.02	66.77	**82.78**	70.92	71.76	72.41
5	90.37	95.82	83.09	93.83	91.17	93.15	**97.00**	96.70
6	84.92	81.71	50.46	80.43	96.27	94.58	98.95	**98.98**
7	**84.73**	64.64	30.93	72.21	72.29	80.45	83.39	**84.09**
8	52.01	96.53	50.53	70.96	60.79	78.83	80.15	**80.56**
9	70.47	76.11	46.58	**71.18**	69.72	68.74	70.38	**72.67**
10	92.15	83.35	69.73	96.43	86.55	92.23	97.14	**99.00**
11	89.39	85.10	50.06	**93.73**	91.95	86.14	92.72	91.70
12	60.04	74.15	63.44	87.95	84.37	85.12	**88.60**	86.95
13	86.99	84.25	53.35	84.90	**97.27**	91.57	95.57	95.44
14	92.99	93.14	76.73	92.99	99.25	98.24	**100**	**100**
15	92.83	96.01	54.00	88.30	91.43	94.13	92.58	**92.70**
OA	78.95	82.55	58.35	83.37	82.10	83.39	86.97	**87.11**
AA	81.09	85.39	59.03	84.21	84.71	85.29	88.77	**88.84**
K	77.32	81.13	55.18	82.07	80.67	82.06	85.92	**86.07**

TABLE VIII

COMPARISON EXPERIMENTAL RESULTS ON HOUSTON UNIVERSITY USING 30 TRAINING SAMPLES EACH CLASS

Models	3D-LWNet	SSRN	1-D Auto-CNN	3-D Auto-CNN	3D-ANAS†	Hy-NAS	HyT-NAS	HyT-NAS +OV
1	84.81	90.91	72.25	87.50	**90.01**	**90.01**	78.94	80.59
2	80.22	98.61	63.96	77.91	79.49	85.75	92.42	**92.75**
3	93.45	97.75	76.61	92.74	98.78	85.08	98.17	**99.85**
4	79.74	82.11	51.29	72.65	85.22	86.63	91.53	**91.94**
5	90.90	91.51	82.25	96.14	99.00	**99.50**	96.84	98.34
6	81.44	65.83	55.32	84.86	91.93	97.89	**99.30**	**99.30**
7	87.83	81.52	34.20	73.03	66.53	76.30	83.06	**91.43**
8	63.69	98.15	62.11	76.64	75.50	72.43	77.24	**77.99**
9	77.50	67.71	49.84	71.10	81.93	74.34	86.72	**88.45**
10	93.26	91.49	64.48	96.67	90.14	89.81	**97.72**	96.88
11	91.04	93.38	50.23	92.31	87.36	87.03	94.73	**96.23**
12	79.40	90.44	62.47	91.48	89.52	87.18	92.20	**92.29**
13	90.41	69.84	50.92	85.8	88.81	94.64	98.60	**99.07**
14	90.65	84.09	76.07	90.65	**99.23**	96.13	96.91	96.39
15	92.47	97.27	62.94	88.85	85.97	96.29	94.03	**98.39**
OA	84.17	86.60	60.36	84.45	85.81	86.22	90.04	**91.14**
AA	85.12	86.71	61.00	85.22	87.29	87.93	91.89	**92.66**
K	82.96	85.53	57.35	83.25	84.66	85.10	89.64	**90.42**

2) After grafting the transformer structure, the proposed HyT-NAS achieves better performance than other autodesigned methods. For example, HyT-NAS achieves 98.03% OA, 98.41% AA, and 97.39% K when 20 training samples are extracted from each class, which are 2.29%, 1.81%, and 3.02% higher than 3-D-ANAS, respectively.
3) The OV enhancement strategy can further improve the performance. As shown in Table III, using OV increases OA, AA, and K by 0.74%, 0.40%, and 0.98%, respectively.

To save space, we only present the visual results using 30 training samples per class in Fig. 7. In order to clearly illustrate the difference, we placed a partially enlarged patch in the upper right corner of each result map. It can be easily found from the partially enlarged patch that there are fewer misclassified pixels in the results of a series of HyT-NAS. Some Asphalt pixels (class 1, cyan) are incorrectly classified as Self-Blocking Bricks (class 8, red) by 3-D-LWNet and 3-D Auto-CNN. A lot of pixels belonging to Self-Blocking Bricks are incorrectly classified as Meadows (Class 2, green) by 3-D-ANAS. However, in the results of a series of HyT-NAS, all pixels belonging to Asphalt and Self-Blocking Bricks are correctly classified.

Tables V and VI collect the comparison results on Pavia center, and Fig. 8 shows the visual results of qualitative analysis. Compared with the results on Pavia University, the accuracy of these seven methods all improved to certain extents, and the proposed HyT-NAS still attains the best performance. Observing from Fig. 8, the number of bitumen pixels that a series of HyT-NAS approaches incorrectly classified into self-blocking Bricks is significantly less than that of other methods. Although Hy-NAS with only improved hybrid spatial-spectrum search space still makes some false predictions on the bitumen class, the introduction of the transformer finally handles the problems very well.

The comparison results on Houston University are shown in Tables VII and VIII and Fig. 9. Compared with the first two datasets, Houston University contains more spectral bands and more object categories. Therefore, the classification accuracy of all methods on this dataset is relatively low. The classification performance of different methods is quite different. As shown in Fig. 9, the result map of 1-D Auto-CNN clearly shows the structural outlines of different buildings, for example, the dark red part of the partially enlarged area (commercial, level 8). However, many misclassified pixels are distributed throughout the result image and look like salt and pepper noise, resulting in relatively poor visual effects. On the contrary, 3-D Auto-CNN showed very smooth results, in which the outline of the structure was almost lost. 3-D-ANAS and HyT-NAS have kept a relatively good balance between displaying good visual effects and maintaining the contour structure, and gained better performance than other algorithms. As shown in the enlarged image in Fig. 9, 3-D-ANAS misclassifies some pixels classified as land into stressed grass and highway, while HyT-NAS has very few misclassified pixels. From Tables VII and VIII, it is obvious that the results of HyT-NAS are better than those of 3-D-ANAS regardless of whether the training samples are 20 or 30. Specifically, when there are 20 training samples for each class, the OA, AA, and K of HyT-NAS are 86.97%, 88.77%, and 85.92%, respectively, which are significantly higher than that of 3-D-ANAS. When the training samples of each class increase to 30, the advantages of HyT-NAS and 3-D-ANAS are more obvious, increasing by 4.23%, 4.60%, and 4.98% on OA, AA, and K, respectively.

TABLE IX
COMPARISONS BETWEEN DIFFERENT SEARCH SPACES ON HOUSTON UNIVERSITY

Search Space	Model Size	Transformer	OA	AA	K
Spectral	1.41 MB	✗	83.04	85.25	81.68
Spatial	1.48 MB	✗	84.85	86.90	83.63
Spectral + Spatial	1.44 MB	✗	86.22	87.93	85.10
Spectral	9.98 MB	✓	88.18	89.75	87.22
Spatial	10.04 MB	✓	89.53	90.96	88.67
Spectral + Spatial	10.00 MB	✓	90.04	91.89	89.64

E. Ablation Study

HSI has different spatial and spectral resolutions. During the searching stage, different layers tend to select different types of cells. We speculate that merely maintaining a space-dominated cell or a spectrum-dominated cell would affect the performance of the algorithm although both kinds of cells contain the 3-D convolution. Here, ablative experiments are conducted to verify the effectiveness of hybrid search space. Besides, we also compared the classification accuracy of the model with and without the transformer unit. The experiments are carried out on the most challenging dataset Houston University with 30 training samples per class.

As shown in Table IX, the proposed method with hybrid search space achieves the highest accuracy. When only the spectral search space is retained in the model, the classification accuracy is the lowest. Importing different types of cells can dig out the spatial and spectral information jointly and freely in HSI classification tasks. In addition, the introduction of the transformer has brought about a 5% improvement in accuracy in all different search space settings. This illustrates the importance of fully mining the associated information between pixels in the classification of HSIs.

F. Architecture Analyze

In this section, the architectures searched by HyT-NAS are shown in Fig. 10 and analyzed. Since the three datasets have different spectral and spatial resolutions, and land covers, we searched for the architecture on each dataset separately. Although these three architectures are different in topology and operations, they also have some common characteristics.

1) 2-D spatial convolution and 2-D spectral convolution play important roles in the final selected operations. As introduced in Section III-B, the search space that we construct for searching the internal topology includes not only 2-D spatial convolution and 2-D spectral convolution but also 3-D convolution. Even so, HyT-NAS tends to build a network with both 2-D convolution operations and 3-D convolution operations. In most cases, 2-D convolution operations are the main operation, and 3-D convolution operations play the part of the complementary operation. The proportions of 2-D convolution operations in the final network designed on Pavia Center and Pavia University are 41.67% and 52.78%, respectively. Under the architecture searched for Houston University,

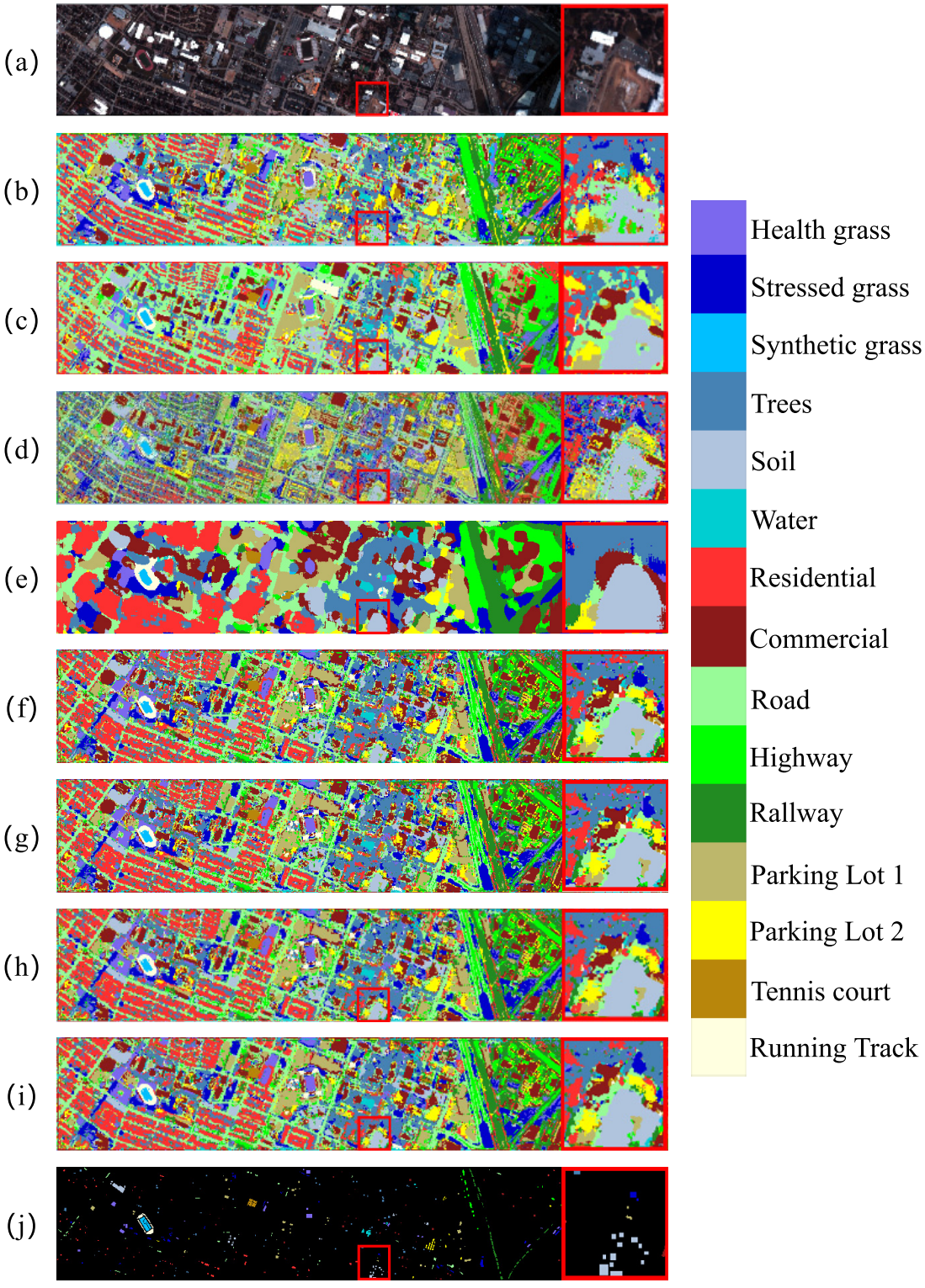

Fig. 9. Comparison experimental results on Houston University using 30 training samples in each class. (a) False color composite. (b) 3-D-LWNet, OA = 84.17%. (c) SSRN, OA = 86.60. (d) 1-D Auto-CNN, OA = 60.36%. (e) 3-D Auto-CNN, OA = 84.45%. (f) 3-D-ANAS, OA = 85.81%. (g) Hy-NAS + MS, OA = 86.22%. (h) HyT-NAS, OA = 90.04%. (i) HyT-NAS + OV, OA = 91.14%. (j) Ground-truth map.

2-D convolution operations occupied 44.44% of all operations. The proportions of 3-D convolution operations are 13.89%, 8.33%, and 16.67% on Pavia Center, Pavia University, and Houston University. This shows that, although 3-D convolution fits the data characteristics of HSIs, widely utilizing it as in traditional algorithms is not necessary. The 2-D–3-D mixed network architecture that we searched has fewer parameters and higher parameter utilization compared with models under the same scale.

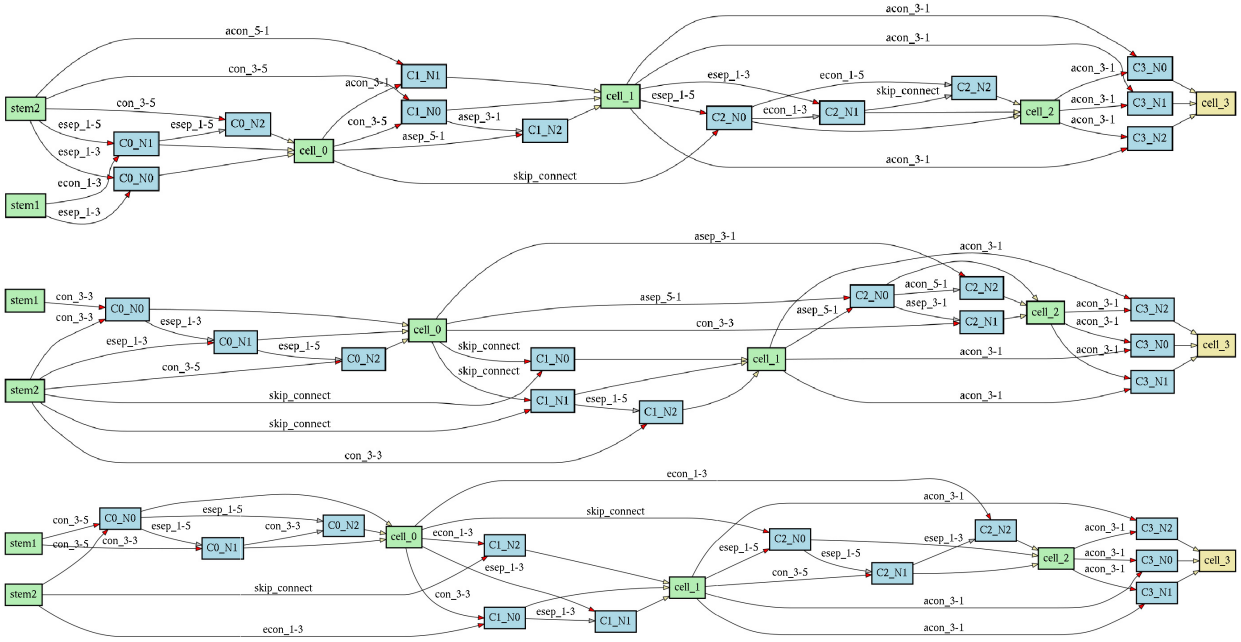

Fig. 10. Final architectures found for three datasets. From (Top) to (Bottom): Pavia University, Pavia Center, and Houston University.

2) In the final network, 3-D convolution operations are distributed from the beginning to the end. However, 2-D spectral convolutions dominate in the shallow layer, while the number of 2-D spatial convolutions is relatively large in the deep layer, as shown in Fig. 10. In the architectures of the Pavia Center and Pavia University, the spectral convolutions account for the majority in the first two layers, but the spatial convolutions account for the highest proportion in the last two layers. In the final architecture for Houston University, the first three layers are almost all spectral convolutions, and only the last layer of the network is mainly spatial convolution. In classic HSI classification networks, such as SSRN [21], spectral convolution is always performed first, followed by the spatial convolution. Our experimental results are consistent with the manual design experience.
3) With the enrichment of spectral information, the proportion of spectral convolution in the final network gradually increases. In different HSIs, the richness of spectral information and spatial information is quite different. For example, both Pavia University and Pavia Center have only 102 bands, while Houston University has 144 bands. Traditional 3-D convolution pays the same attention to spatial information and spectral information. The hybrid spatial–spectral search space that we proposed can flexibly adjust the ratio of spatial and spectral convolutions freely according to the proportion of the spatial–spectral information of the data itself. As shown in Fig. 10, although the search space is exactly the same, 2-D-spectral convolution and 3-D convolution account for 33.33% and 25.00% of operations in Pavia University and Pavia Center, respectively. In the architecture for Houston University, this proportion has risen to 44.44%. This may be because Houston University has a larger number of spectra bands and requires more spectral convolutions to extract rich spectral information. The experimental results prove that the hybrid search space can adapt well to the characteristics of different data.

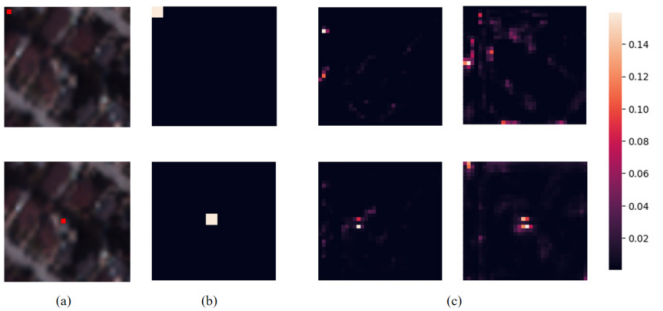

Fig. 11. Attention map analyses. (a) Input patch. Two pixels marked in red color are pixels to be classified. (b) Region of interesting of CNN. (c) Attention maps of different heads of the transformer unit.

G. Attention Map Analyze

In this section, we analyze the difference between using and not using the transformer unit on Houston University. A visualized result is presented in Fig. 11. After introducing the transformer unit, the receptive field of the feature map has expanded from a small range to a wider global one. In addition, the CNN structure pays the same level of attention to each pixel in a local receptive field, while the transformer unit can capture the relationship between global pixels in HSI adaptively.

Fig. 11(c) shows that the attention maps produced by different heads inside the transformer are also different.

This observation indicates the effectiveness of the multiattention head mechanism in HSI classification tasks. Various heads inside the transformer focus on different types of information, according to the spatial position and neighborhood of the pixel itself. The multiattention mechanism fuses the information of different heads, resulting in a more comprehensive and robust feature map.

V. Conclusion

In this article, we have proposed an autodesigned HSI classification method based on the hybrid CNN-transformer framework. The proposed HyT-NAS has been compared with other manual designed CNN-based HSI classification methods (3-D-LWNet) and automatic design CNN-based methods (1-D Auto-CNN, 3-D Auto-CNN, and 3-D-ANAS) comprehensively on three typical public HSI datasets. The experimental results show that the HyT-NAS outperforms other state-of-the-art DL-based algorithms. In addition, abundant ablation studies have been carried out to verify the effectiveness of the proposed hybrid spatial–spectral search space and the grafted transformer. The results of the ablation study demonstrated that the HyT-NAS does find a local optimum architecture in the architecture search space. Compared with the pure CNN-based HSI classification framework, the hybrid CNN-transformer framework captures the global relationship between pixels. In future work, we will focus on designing a more efficient NAS approach to automatically design a full transformer architecture for HSI classification.

References

[1] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.

[2] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, "Multiple Kernel learning for hyperspectral image classification: A review," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6547–6565, Nov. 2017.

[3] T. A. Carrino, A. P. Crósta, C. L. B. Toledo, and A. M. Silva, "Hyperspectral remote sensing applied to mineral exploration in southern Peru: A multiple data integration approach in the Chapi Chiara gold prospect," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 64, pp. 287–300, Feb. 2018.

[4] J. Behmann, J. Steinrücken, and L. Plümer, "Detection of early plant stress responses in hyperspectral images," *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 98–111, Jul. 2014.

[5] J. Transon, R. D'Andrimont, A. Maugnard, and P. Defourny, "Survey of hyperspectral earth observation applications from space in the Sentinel-2 context," *Remote Sens.*, vol. 10, p. 157, Jan. 2018.

[6] W. Sun, J. Peng, G. Yang, and Q. Du, "Fast and latent low-rank subspace clustering for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3906–3915, Jun. 2020.

[7] X. Zheng, Y. Yuan, and X. Lu, "Dimensionality reduction by spatial–spectral preservation in selected bands," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5185–5197, Sep. 2017.

[8] R. Hang and Q. Liu, "Dimensionality reduction of hyperspectral image using spatial regularized local graph discriminant embedding," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 9, pp. 3262–3271, Sep. 2018.

[9] F. Luo, Z. Zou, J. Liu, and Z. Lin, "Dimensionality reduction and classification of hyperspectral image via multistructure unified discriminative embedding," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.

[10] Y. Duan, H. Huang, and T. Wang, "Semisupervised feature extraction of hyperspectral image using nonlinear geodesic sparse hypergraphs," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.

[11] Z. Lin, Y. Chen, X. Zhao, and G. Wang, "Spectral–spatial classification of hyperspectral image using autoencoders," in *Proc. 9th Int. Conf. Inf., Commun. Signal Process.*, Dec. 2013, pp. 1–5.

[12] 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.

[13] 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.

[14] H. Zhang and Y. Li, "Spectral–spatial classification of hyperspectral imagery based on deep convolutional network," in *Proc. Int. Conf. Orange Technol. (ICOT)*, Dec. 2016, pp. 44–47.

[15] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, Jan. 2015.

[16] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2015, pp. 4959–4962.

[17] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 6, pp. 468–477, Jun. 2015.

[18] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, 2017.

[19] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, Jan. 2017.

[20] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[21] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Aug. 2018.

[22] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.

[23] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.

[24] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*. New Orleans, LA, USA: OpenReview.net, May 2019.

[25] Y. Chen, K. Zhu, L. Zhu, X. He, P. Ghamisi, and J. A. Benediktsson, "Automatic design of convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7048–7066, Apr. 2019.

[26] H. Zhang, C. Gong, Y. Bai, Z. Bai, and Y. Li, "3D-ANAS: 3D asymmetric neural architecture search for fast hyperspectral image classification," 2021, *arXiv:2101.04287*.

[27] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[28] S. Mei, J. Ji, Q. Bi, J. Hou, Q. Du, and W. Li, "Integrating spectral and spatial information into deep convolutional neural networks for hyperspectral classification," in *Proc. Int. Geosci. Remote Sens. Symp.*, Jul. 2016, pp. 5067–5070.

[29] Z. Meng, L. Jiao, M. Liang, and F. Zhao, "A lightweight spectral-spatial convolution module for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[30] S. Jia *et al.*, "A lightweight convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4150–4163, May 2021.

[31] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[32] N. Wang *et al.*, "NAS-FCOS: Fast neural architecture search for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11943–11951.

[33] C. Liu *et al.*, "Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 82–92.

[34] Z. Zhong, J. Yan, W. Wu, J. Shao, and C.-L. Liu, "Practical block-wise neural network architecture generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2423–2432.

[35] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4780–4789.
[36] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*. Vancouver, BC, Canada: OpenReview.net, Apr./May 2018.
[37] D. Song, C. Xu, X. Jia, Y. Chen, C. Xu, and Y. Wang, "Efficient residual dense block search for image super-resolution," in *Proc. AAAI*, 2020, pp. 12007–12014.
[38] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[39] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[40] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.
[41] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021, *arXiv:2103.14030*.
[42] B. Graham *et al.*, "LeViT: A vision transformer in ConvNet's clothing for faster inference," 2021, *arXiv:2104.01136*.
[43] X. He, Y. Chen, and Z. Lin, "Spatial–spectral transformer for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 3, p. 498, Jan. 2021.
[44] Y. Qing, W. Liu, L. Feng, and W. Gao, "Improved transformer net for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 11, p. 2216, Jun. 2021.
[45] D. Hong *et al.*, "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.
[46] Z. Dai, H. Liu, Q. Le, and M. Tan, "CoAtNet: Marrying convolution and attention for all data sizes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1–13.
[47] H. Zhang, W. Hu, and X. Wang, "EdgeFormer: Improving light-weight ConvNets by learning from vision transformers," 2022, *arXiv:2203.03952*.
[48] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.
[49] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
[50] W. Wang *et al.*, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," 2021, *arXiv:2102.12122*.

Xizhe Xue received the B.E. degree from Northwestern Polytechnical University, Xi'an, China, in 2018, where she is currently pursuing the Ph.D. degree with the School of Computer Science.

Her research interests include visual object tracking, hyperspectral image (HSI) image processing, and image segmentation techniques.

Haokui Zhang received the M.S. and Ph.D. degrees in computer application technology from the Shaanxi Provincial Key Laboratory of Speech and Image Information Processing, Northwestern Polytechnical University, Xi'an, China, in 2021 and 2016, respectively.

He is currently an Algorithm Researcher with Intellifusion, Shenzhen, China. His research interests cover information retrieval, image restoration, and hyperspectral image classification.

Bei Fang received the Ph.D. degree in computer science and technology from the School of Computer Science, Northwestern Polytechnical University, Xi'an, China, in 2019.

She is currently a Post-Doctoral Research Associate with the Key Laboratory of Modern Teaching Technology, Ministry of Education, Shaanxi Normal University, Xi'an. Her research interests include computer vision, hyperspectral images (HSIs), image processing, and deep learning techniques.

Zongwen Bai received the M.S. degree from Yan'an University, Yan'an, China, in 2008. He is currently pursuing the Ph.D. degree with the School of Computer Science, Northwestern Polytechnical University, Xi'an, China.

He is currently an Associate Professor with the School of Physics and Electronic Information, Yan'an University. His research interests include computer vision, natural language processing, deep learning, hyperspectral image super-resolution, and image fusion.

Ying Li received the Ph.D. degree in electrical circuits and systems from the National Key Laboratory of Radar Signal Processing, Xidian University, Xi'an, China, in 2002.

She is currently a Professor with the School of Computer Science, Northwestern Polytechnical University, Xi'an. Her research interests cover image processing, computation intelligence, and signal processing.