Spectral–Spatial Transformer Network for Hyperspectral Image Classification: A Factorized Architecture Search Framework

Zilong Zhong, *Member, IEEE*, Ying Li, *Member, IEEE*, Lingfei Ma, Jonathan Li, *Senior Member, IEEE*, and Wei-Shi Zheng, *Member, IEEE*

***Abstract*— Neural networks have dominated the research of hyperspectral image classification, attributing to the feature learning capacity of convolution operations. However, the fixed geometric structure of convolution kernels hinders long-range interaction between features from distant locations. In this article, we propose a novel spectral–spatial transformer network (SSTN), which consists of spatial attention and spectral association modules, to overcome the constraints of convolution kernels. Also, we design a factorized architecture search (FAS) framework that involves two independent subprocedures to determine the layer-level operation choices and block-level orders of SSTN. Unlike conventional neural architecture search (NAS) that requires a bilevel optimization of both network parameters and architecture settings, the FAS focuses only on finding out optimal architecture settings to enable a stable and fast architecture search. Extensive experiments conducted on five popular HSI benchmarks demonstrate the versatility of SSTNs over other state-of-the-art (SOTA) methods and justify the FAS strategy. On the University of Houston dataset, SSTN obtains comparable overall accuracy to SOTA methods with a small fraction (1.2%) of multiply-and-accumulate operations compared to a strong baseline spectral–spatial residual network (SSRN). Most importantly, SSTNs outperform other SOTA networks using only 1.2% or fewer MACs of SSRNs on the Indian Pines, the Kennedy Space Center, the University of Pavia, and the Pavia Center datasets.**

***Index Terms*— Factorized architecture search (FAS), spatial attention, spectral association, spectral–spatial transformer network (SSTN).**

Manuscript received April 14, 2021; revised July 7, 2021, August 16, 2021, and September 13, 2021; accepted September 22, 2021. This work was supported in part by China Postdoctoral Science Foundation under Grant 2020TQ0372 and Grant 2020M672964 and in part by Guangdong Natural Science Foundation under Grant 2021A1515011843. *(Corresponding author: Wei-Shi Zheng.)*

Zilong Zhong and Wei-Shi Zheng are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China, and also with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Guangzhou 510006, China (e-mail: z26zhong@uwaterloo.ca; wszheng@ieee.org).

Ying Li is with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: ying.li@bit.edu.cn).

Lingfei Ma is with the Engineering Research Center of State Financial Security, Ministry of Education, Beijing 102206, China, and also with the School of Statistics and Mathematics, Central University of Finance and Economics, Beijing 102206, China (e-mail: l53ma@cufe.edu.cn).

Jonathan Li is with the Department of Geography and Environmental Management and the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Codes are available at: https://github.com/zilongzhong/SSTN

Digital Object Identifier 10.1109/TGRS.2021.3115699

I. Introduction

HYPERSPECTRAL image (HSI) classification requires labeling each pixel in the imagery as belonging to one of the predefined land cover categories. This challenging task forms the cornerstone of various remote sensing applications, including object detection, land-cover mapping, semantic segmentation, and anomaly detection [2]–[5]. The hundreds of contiguous spectral bands of HSIs separate them from typical images, and this distinctive property prevents machine learning methods from being directly transferred to HSI analysis. Meanwhile, the spatial contexts of HSI samples provide complementary information to their abundant spectral signatures for precise recognition. Considering each HSI dataset contains a limited number of classes, this task can be regarded as projecting samples from high-dimensional data space to a compact semantic space. The essence lies in integrating the characteristics of data into network design.

Traditional HSI classification models involve two independent steps: Feature engineering and classifier training [6]–[10]. Although this two-step paradigm has been adopted for HSI classification by many pioneering works, the conventional methods suffer from the disadvantages of low generalizability and limited representational capacity. In recent years, inspired by the burst of deep learning in addressing various vision problems [11]–[13], many research groups manage to transfer the remarkable feature learning capacity of neural networks to HSI recognition problems [14]–[16]. In these methods, the tasks of learning features and training classifiers are integrated in an end-to-end manner. Such advancement enables practitioners to focus on the design of neural networks or learning frameworks for achieving better recognition performance. However, two obstacles still hinder the development of neural networks for HSI classification.

The first shortcoming is the geometric constraints imposed by convolution kernels, the square structure of which limits their receptive fields to local contexts. Although conventional convolution extracts spatial features effectively, they seldom receive information from long-distance positions in feature maps. Meanwhile, various vision tasks involve spatial recognition have demonstrated the importance of long-range interactions between pixels from different locations. However, the transformer units applied in vision tasks are computed with

TABLE I

MULTIPLY–ACCUMULATE OPERATIONS (M), NUMBER OF PARAMETERS (K) IN SUBSCRIPT, AND THEIR RESPECTIVE FRACTIONS BETWEEN SSTNS AND SPECTRAL–SPATIAL RESIDUAL NETWORKS (SSRNS) IN DIFFERENT MODELS ON IN, UP, KSC, UH, AND PC DATASETS

Dataset	CONV	STN	SPA	SSAN	SSRN	AUTO	SSTN	Fraction
IN	$33.1_{1088.2}$	$16.2_{226.4}$	$39.1_{4085.0}$	$123.5_{713.5}$	$211.6_{725.9}$	$24.9_{316.6}$	$\mathbf{1.65_{20.5}}$	$\mathbf{0.8\%_{2.8\%}}$
KSC	$24.9_{842.5}$	$14.4_{204.4}$	$31.2_{3262.7}$	$108.3_{628.8}$	$189.6_{651.1}$	$19.3_{245.2}$	$\mathbf{2.26_{27.3}}$	$\mathbf{1.2\%_{4.2\%}}$
UP	$8.3_{289.2}$	$9.7_{146.3}$	$13.0_{1404.1}$	$62.8_{374.6}$	$107.6_{396.9}$	$6.5_{82.8}$	$\mathbf{1.3_{16.2}}$	$\mathbf{1.2\%_{4.1\%}}$
UH	$16.7_{565.5}$	$12.2_{177.5}$	$22.2_{2333.8}$	$88.1_{516.0}$	$150.9_{538.3}$	$12.9_{164.4}$	$\mathbf{5.8_{71.9}}$	$\mathbf{1.2\%_{4.2\%}}$
PC	$8.4_{283.7}$	$9.7_{145.6}$	$12.8_{1385.4}$	$61.5_{367.6}$	$105.4_{389.9}$	$6.5_{82.3}$	$\mathbf{1.3_{16.2}}$	$\mathbf{1.2\%_{4.2\%}}$

respect to all spatial locations and thus inherently overlook the ample spectral information of HSIs.

The second drawback originates from the design choices of network architecture, which is infinite in theory. Although neural networks have dominated the research of HSI classification, their architecture design still largely relies on domain knowledge from experts. To alleviate such a dilemma, neural architecture search (NAS) has attracted a lot of scholarly attention as a potential solution. Since NAS presents a promising alternative paradigm capable of designing networks automatically. Unfortunately, prohibitive computation expense and unstable training cost prevent the NAS strategy from being widely adopted.

Many preceding works have explored practical constraints in the remote sensing community to regularize HSI datasets for various tasks. For example, the subspace structure prior [17] and material-level data distribution [18] are introduced for hyperspectral unmixing and object detection, respectively. Also, designing a suitable sampling strategy is discussed for HSI analysis [19], [20]. Unlike these data-centric methods, this article focuses on introducing constraints on architecture spaces to avoid prohibitive costs of NAS.

Recently, the attention mechanism has been adopted rapidly for addressing remote sensing tasks [16], [21]. For instance, [22] designed an approach that emphasizes certain hyperspectral bands for improving HSI classification. However, these methods adopt attention units as an additional part to boost backbone models. Contemporarily, the work [23] proposed the HSI-BERT model that is composed of multiple attention layers. This method requires the input HSI samples to be flattened to a sequence of vectors to satisfy the requirements of natural language processing. Motivated by the pioneering research, we concentrate on designing transformer networks composed of novel attention units that account for the characteristics of HSIs.

To this end, we propose an efficient and effective spectral–spatial transformer network (SSTN), the configurations of which are searched by a novel factorized architecture search (FAS) strategy. First, we embed the attention mechanism into both spatial and spectral feature learning modules. These attention modules can capture long-range interactions by replacing the convolution operations with more flexible transformer units, thus representing spectral–spatial features with reduced computational cost. Second, we propose an FAS framework that uses the innovative transformer units as building blocks, upon which the architecture settings of SSTN are searched. Specifically, we introduce six combinations of four basic operators to explore the optimal layer-level configuration. Then, we search for the best block-level order using the selected operators. Finally, we adopt all network settings searched from the FAS to configure the SSTN. The finalized SSTN contains two spatial transformer and two spectral transformer units, outperforming state-of-the-art (SOTA) expert-designed or auto-searched models [1], [24] in four out of five HSI datasets with a much lower computational cost as shown in Table I.

The main contributions of this article are threefold and listed as follows.

1) We introduce the SSTN consisting of spectral and spatial transformer blocks to extract spectral–spatial HSI features, replacing convolution operations with spatial attention and spectral association modules.
2) We propose a novel FAS framework that only focuses on two crucial factors, the layer-level operation choices and the block-level sequential orders of SSTN, thus enabling a fast and practical architecture search.
3) The effectiveness and efficiency of SSTNs have been demonstrated on three challenging HSI benchmarks, outperforming human-designed as well as NAS-based networks.

We arrange the remaining part as follows. Section II summarizes related works from three perspectives. Section III introduces the detailed framework of FAS and describes the building elements of SSTN. Then, Section IV presents the hyperparameter configuration, experiment results, and corresponding analysis. We conclude this work in Section V.

II. RELATED WORK

A. Attention Mechanism

Motivated by the attention mechanism used in modeling various sequential/data [25], [26], multiple works introduce novel attention modules to overcome the geometric limitations of convolution kernels for HSI classification [16], [27], [28]. For example, spectral–spatial attention network (SSAN) first incorporates attention mechanism in spectral and spatial feature learning units [21]. Spectral gates generated by global convolution are designed to determine the importance of different spectral bands [29]. Also, an embedded attention module is introduced to deprecate interfering HSI pixels [16]. Recently, consecutive channel and spatial attention blocks are adopted to improve their spectral–spatial residual counterparts [30]. Although obtaining promising mapping results, these attention-based networks calculate spectral attention as

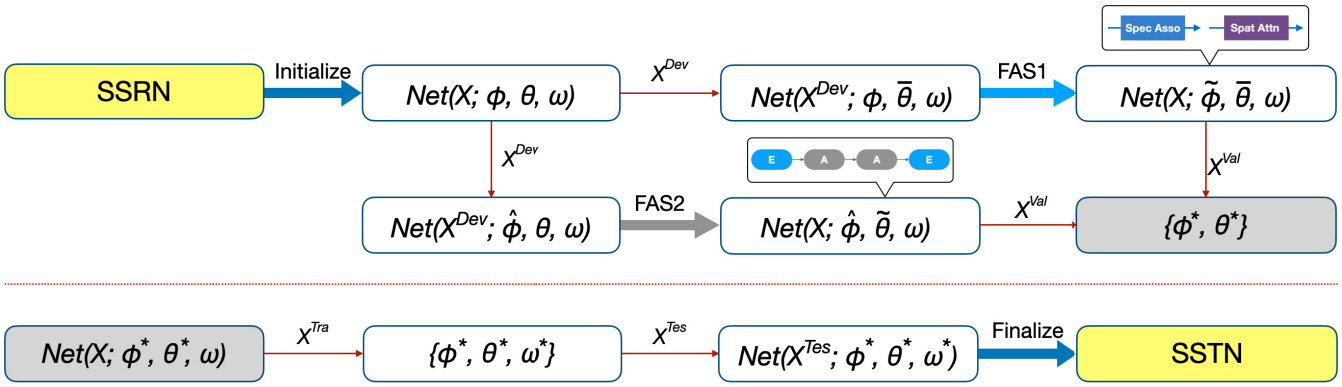


Fig. 1. Framework of FAS framework. The super network Net($\boldsymbol{X}; \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\omega}$) built for search is partially initialized by the hyperparameters of SSRN. The upper part shows the searching stage that aims to figure out the two optimal factors $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ of network architecture on $\boldsymbol{X}^{\text{Dev}}$. The lower part presents the training stage that follows a standard network training procedure using the hyperparameters $\boldsymbol{\phi}^*$ and $\boldsymbol{\theta}^*$ searched in the previous step on $\boldsymbol{X}^{\text{Tra}}$. Finally, we discover an SSTN that consists of spatial attention and spectral association modules using the FAS strategy.

the correlation between feature channels. Instead, we propose a novel spectral transformer unit that models the association between a set of spectral kernels and spatial locations with marginal computational resources.

B. Neural Architecture Search

NAS has been studied in growing numbers of literature for automatically designing networks for vision tasks. Zoph and Le first introduce reinforcement learning to NAS via adopting an RNN controller, which produces hyperparameters to search neural networks [31]. Then, differentiable architecture search (Darts) advocates establishing a differentiable search space, resulting in efficient network searching [32]. Several recent articles employ the NAS strategy to analyze remotely sensed data. For instance, Peng *et al.* [33] put forward a gradient-based NAS method to search for optimal convolutional networks for classifying remote sensing scenes. Dong *et al.* [34] designed a data-specific search space equipped with a one-shot strategy for differentiable NAS. Also, Chen *et al.* [24] introduced automatic network design using 1-D or 3-D CNNs for HSI classification. However, these NAS-based methods suffer from heavy computational burden and memory footprint. In this work, we explore an FAS framework that enables efficient network searching for HSI analysis.

C. Expert-Designed Networks

Neural networks have been proven to be more effective for HSI feature representation compared to conventional machine learning methods [35]–[37]. For instance, autoencoder and convolution neural networks (CNNs) are among the earliest spectral approaches used for HSI classification and outperform traditional classifiers like SVM [38], [39]. Zhong *et al.* [1] proposed spectral convolution to boost the spectral feature learning capacity of CNN backbones. Interestingly, Mou *et al.* [40] introduced recurrent neural networks (RNNs) to learn spectral features regarding each pixel in an HSI as sequences. Many recent articles have demonstrated that spatial features play a crucial role in achieving excellent HSI classification performance. CNN and its variants are employed to capture spectral–spatial features for achieving discriminative HSI representation [15], [41]. Besides, other learning methods usually exploit inductive biases, such as sparseness or smoothness constraints. For example, graph models, such as Markov random fields, are widely used to refine the classification maps to boost pixel-wise HSI recognition accuracy [42], [43]. Motivated by these enlightening works, we aim to integrate the edges of NAS and experts' intuition by designing an FAS strategy using novel spectral–spatial modules that account for the characteristics of HSI.

III. Proposed Framework

Fig. 1 shows the FAS framework used to search the settings of SSTN for HSI classification. Suppose that an HSI dataset contains a train set $\{\boldsymbol{X}^{\text{Tra}}, \boldsymbol{y}^{\text{Tra}}\}$, a validate set $\{\boldsymbol{X}^{\text{Val}}, \boldsymbol{y}^{\text{Val}}\}$, and a test set $\{\boldsymbol{X}^{\text{Tes}}, \boldsymbol{y}^{\text{Tes}}\}$. Standard learning-based methods approach the classification task by training a network Net$(\cdot; \boldsymbol{\eta}, \boldsymbol{\omega})$ to fit the training set such that the trained model can generalize to the unseen test set and make decent predictions. This objective can be formulated as follows:

$$\{\boldsymbol{\eta}^*, \boldsymbol{\omega}^*\} = \underset{\boldsymbol{\eta}, \boldsymbol{\omega}}{\operatorname{argmin}} \{\mathcal{L}_{\text{train}}(\boldsymbol{\eta}, \boldsymbol{\omega}), \mathcal{L}_{\text{val}}(\boldsymbol{\eta}, \boldsymbol{\omega})\} \tag{1}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\omega}$ denote the architecture settings and network parameters to be learned, respectively. The train loss $\mathcal{L}_{\text{train}}$ and the validate loss $\mathcal{L}_{\text{val}}$ are defined as follows:

$$\mathcal{L}_{\text{train}}(\boldsymbol{\eta}, \boldsymbol{\omega}) = \mathcal{L}\big(\boldsymbol{y}^{\text{Tra}}, \text{Net}\big(\boldsymbol{X}^{\text{Tra}}; \boldsymbol{\eta}, \boldsymbol{\omega}\big) \tag{2}$$

$$\mathcal{L}_{\text{val}}(\boldsymbol{\eta}, \boldsymbol{\omega}) = \mathcal{L}\big(\boldsymbol{y}^{\text{Val}}, \text{Net}\big(\boldsymbol{X}^{\text{Val}}; \boldsymbol{\eta}, \boldsymbol{\omega}\big). \tag{3}$$

The analytical solutions of (1) are nontrivial to obtain due to the hierarchical architecture of neural networks and the high dimension of HSI samples, while the settings of these models are proposed by expert. Thus, NAS is introduced to automatically design network architectures for HSI classification [24]. The NAS strategy involves a bilevel optimization procedure, which means that it alternately runs both optimization steps via gradient descent in each training epoch. Therefore, the optimal

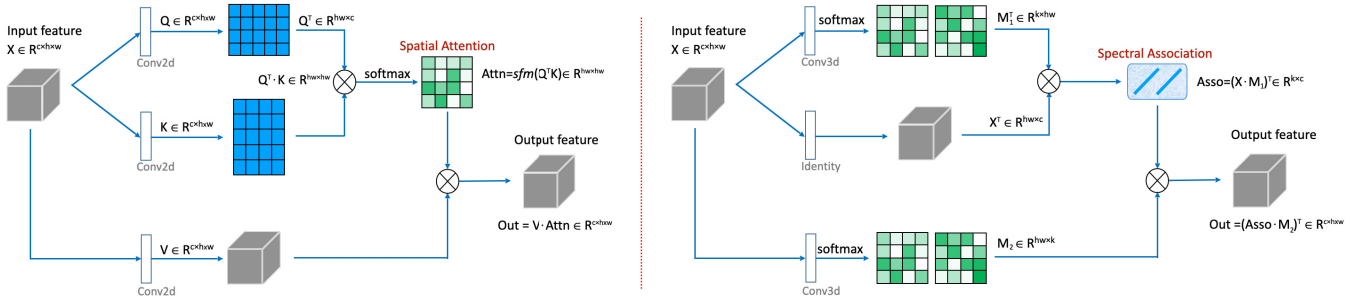


Fig. 2. Illustration of spatial attention module (left) and spectral association module (right). The attention maps Attn $\in \mathbb{R}^{hw\times hw}$ in the spatial attention module is produced by multiplying two reshaped tensors $\boldsymbol{Q}$ and $\boldsymbol{K}$. Instead, the attention maps $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$ in the spectral association module are the direct output of a convolution operation. The spectral association kernels Asso $\in \mathbb{R}^{k\times c}$ represent a compact set of spectral vectors used to reconstruct input feature $\boldsymbol{X}$.

settings $\boldsymbol{\eta}^*$ and parameters $\boldsymbol{\omega}^*$ of a given super network are approximated by a bilevel optimization as follows:

$$\boldsymbol{\omega} := \boldsymbol{\omega} - \xi_1 \nabla_{\boldsymbol{\omega}} \mathcal{L}\big(\boldsymbol{y}^{\mathrm{Tr}}, \mathrm{Net}\big(\boldsymbol{X}^{\mathrm{Tr}}; \boldsymbol{\eta}, \boldsymbol{\omega}\big)\big) \tag{4}$$

$$\boldsymbol{\eta} := \boldsymbol{\eta} - \xi_2 \nabla_{\boldsymbol{\eta}} \mathcal{L}\big(\boldsymbol{y}^{\mathrm{Val}}, \mathrm{Net}\big(\boldsymbol{X}^{\mathrm{Val}}; \boldsymbol{\eta}, \boldsymbol{\omega}(\boldsymbol{\eta})\big)\big) \tag{5}$$

where ξ_1 and ξ_2 are the learning rates of dual optimization levels, respectively. However, this bilevel optimization process usually leads to unstable training and expensive computational costs. To this end, we design a novel and effective FAS framework that replaces network settings $\boldsymbol{\eta}^*$ with two independent architecture factors $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. Then, we build a super network $\mathrm{Net}(\boldsymbol{X}; \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\omega})$ that is partially initialized by the architecture settings of SSRN [1]. The upper part of Fig. 1 shows the searching stage that aims to figure out the optimal settings of the two factors $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ on a develop set $\{\boldsymbol{X}^{\mathrm{Dev}}, \boldsymbol{y}^{\mathrm{Dev}}\}$ and validate set. The lower part presents the training stage that follows a standard network training procedure using the architecture settings $\boldsymbol{\phi}^*$ and $\boldsymbol{\theta}^*$ searched in the previous step on the train set. Finally, we discover an SSTN that consists of spatial attention and spectral association modules using the FAS strategy, which is stabilized by searching disentangled architecture factors progressively while eliminating other unstable aspects.

A. Spatial Attention

Spatial attention mechanism aims to model the interactions between different locations of HSI samples [44], [45]. The left part of Fig. 2 demonstrates the structure of a spatial attention module. The spatial attention module overcomes the constraint imposed by the grid structure of convolution kernels and reduces the high computational costs for processing images of large spatial sizes. Following previous works [16], [21], we use the 2-D convolution with 1×1 kernels to implement the spatial attention module. First, we calculate query, key, and value tensors as follows:

$$\boldsymbol{Q} = \mathcal{F}\big(\boldsymbol{X}; \boldsymbol{W}_Q\big) \in \mathbb{R}^{c'\times h\times w} \tag{6}$$

$$\boldsymbol{K} = \mathcal{F}(\boldsymbol{X}; \boldsymbol{W}_K) \in \mathbb{R}^{c'\times h\times w} \tag{7}$$

$$\boldsymbol{V} = \mathcal{F}(\boldsymbol{X}; \boldsymbol{W}_V) \in \mathbb{R}^{c\times h\times w} \tag{8}$$

where $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$ represent trainable parameters in the convolution operations of query, key, and value tensors, respectively. Also, c, w, h, and c' are channel size, height, width of input feature $\boldsymbol{X}$, and the channel size of $\boldsymbol{Q}$ and $\boldsymbol{K}$, respectively. $\mathcal{F}(\cdot)$ denotes a 2-D convolution with the kernel size of $c' \times c \times 1 \times 1$. Then, we reshape $\boldsymbol{Q}$ and $\boldsymbol{K}$ to the size of $c' \times hw$ for the following matrix multiplication. The attention map is produced as follows:

$$\mathrm{Attn} = \mathrm{sfm}_{-1}\big(\boldsymbol{Q}^T \cdot \boldsymbol{K}\big) \in \mathbb{R}^{hw\times hw} \tag{9}$$

where $\mathrm{sfm}_{-1}(\cdot)$ denotes the softmax function along the last dimension of input tensor. Finally, the output of the spatial attention module can be calculated as follows:

$$\mathrm{Out} = \boldsymbol{V} \cdot \mathrm{Attn} = \boldsymbol{V} \cdot \mathrm{sfm}_{-1}\big(\boldsymbol{Q}^T \cdot \boldsymbol{K}\big) \in \mathbb{R}^{c\times h\times w} \tag{10}$$

where each location of output feature Out is a reconstruction of $\boldsymbol{V}$ by summarizing a weighted value tensor $\boldsymbol{V} \cdot \mathrm{Attn}_i$, where $1 \le i \le hw$. We adopt a skip connection to regularize the spatial attention module and thus enable smooth training.

B. Spectral Association

The spatial attention module introduced in the above subsection establishes the connections between different locations in an HSI cuboid. However, all positions in the cuboid are used for reconstructing the input tensor of the spatial attention module. Therefore, as shown in the right part of Fig. 2, we design a novel spectral association module that integrates out and builds back spatial information using masks generated by 3-D convolution operations. First, we calculate the spectral association kernels as follows:

$$\boldsymbol{M}_1 = \mathrm{sfm}_{-1}(\mathcal{G}\big(\boldsymbol{X}; \boldsymbol{W}_{M_1}\big)) \in \mathbb{R}^{hw\times k} \tag{11}$$

$$\mathrm{Asso} = \boldsymbol{M}_1^T \cdot \boldsymbol{X}^T = (\boldsymbol{X} \cdot \boldsymbol{M}_1)^T \in \mathbb{R}^{k\times c} \tag{12}$$

where $\mathcal{G}(\cdot)$ denotes a 3-D convolution using the kernel size of $k \times 1 \times c \times 1 \times 1$ to produce a tensor of size $k \times h \times w$, on which we impose a softmax function. Then, we adopt the generated mask $\boldsymbol{W}_{M_1}$ to integrate out the spatial information of input feature $\boldsymbol{X}$, resulting in spectral association kernels of size $k \times c$. Finally, the output of the spectral association module can be calculated as follows:

$$\boldsymbol{M}_2 = \mathrm{sfm}_{-1}(\mathcal{G}\big(\boldsymbol{X}; \boldsymbol{W}_{M_2}\big)) \in \mathbb{R}^{hw\times k} \tag{13}$$

$$\mathrm{Out} = \mathrm{Asso}^T \cdot \boldsymbol{M}_2^T = (\boldsymbol{M}_2 \cdot \mathrm{Asso})^T \in \mathbb{R}^{c\times h\times w} \tag{14}$$

where $\boldsymbol{M}_2$ is the normalized output of another 3-D convolution $\mathcal{G}(\cdot)$. In practice, we set $\mathcal{G}(\cdot; \boldsymbol{W}_{M_1})$

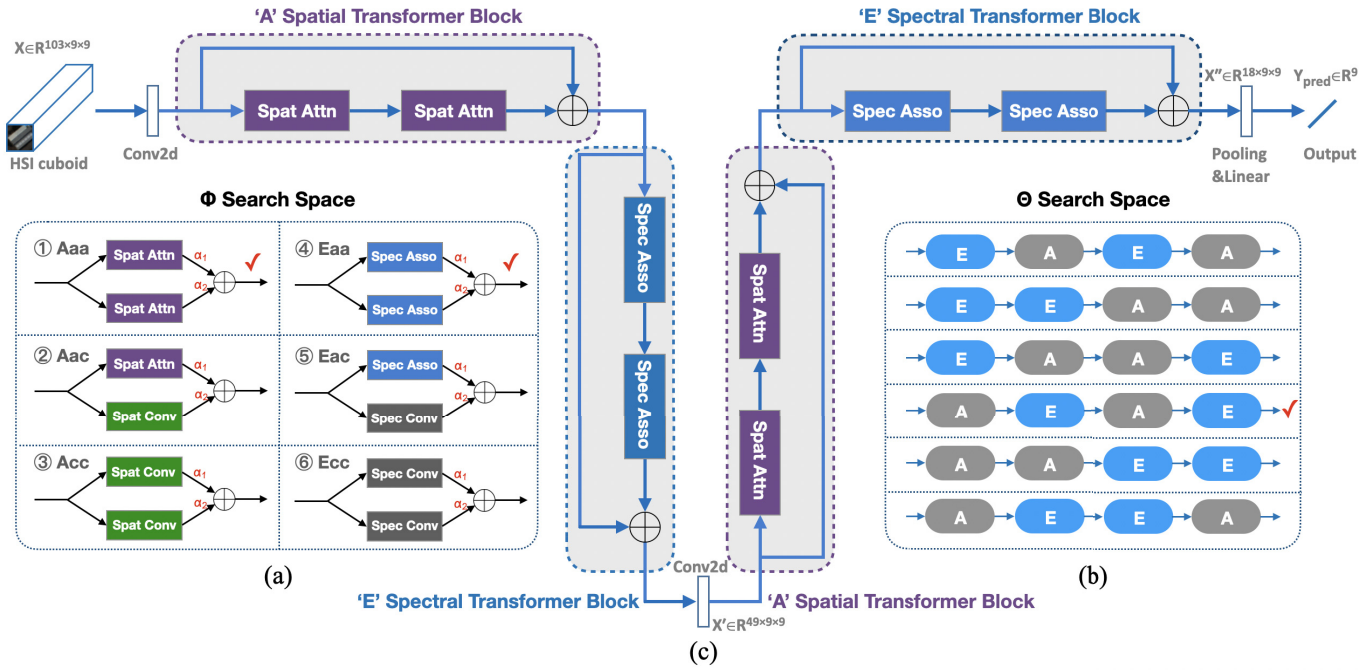


Fig. 3. SSTN and two spaces for searching network hyperparameters. (a) $\mathbf{\Phi}$ search space used to determine layer-level units of spectral and spatial feature learning blocks, where six combinations are presented. (b) $\mathbf{\Theta}$ search space adopted to decide block-level sequential order, in which "A" and "E" denote spatial and spectral blocks, respectively. (c) Final architecture of SSTN has the setting of "AEAE" and is composed of two consecutive pairs of spectral and spatial transformer blocks. The red checkmarks indicate the searched optimal settings in two factorized spaces. The spatial transformer unit is composed of two spatial attention modules, and the spectral counterpart contains two spectral association modules.

and $\mathcal{G}(\cdot; \boldsymbol{W}_{M_2})$ to be the same, and thus, they share trainable parameters. We multiply the transformed spectral association kernels Asso^T and the transformed mask $\boldsymbol{M}_2^T$ to enable a sparse representation of the input feature $\boldsymbol{X}$. Considering that usually, $k \times c \ll hw \times hw$, the sparsity derives from the compact set of spectral vectors $\text{Asso} \in \mathbb{R}^{k \times c}$. It is noteworthy that the spectral association module builds the correlation between spectral kernels and spatial positions, which complements spatial feature learning modules.

C. Disentangled Search Space

NAS strategy [31], [32] is widely used to search the holistic network by searching the best combination of basic building blocks, including various convolution layers, pooling operation, and skip connection. However, such a granular search setting inevitably demands high computation costs and large memory footprints. To this end, we design a novel search framework to overcome these problems and therefore impetus a stable training process. Before introducing the proposed FAS framework, we describe two search spaces as $\mathbf{\Phi}$ and $\mathbf{\Theta}$ in the following two paragraphs.

As shown in Fig. 3(a), the $\mathbf{\Phi}$ search space is used to determine layer-level units of spectral and spatial feature learning blocks. For the spatial block, the left part of Fig. 3(a) demonstrates three combinations of spatial modules, including spatial convolution and spatial attention. As to the spectral block, the right part also shows three compositions of spectral units, including spectral convolution and spectral association. We employ "Aac" to indicate a spatial layer consisting of spatial attention and a spatial convolution for simplicity's sake. Similarly, we adopt "Eac" to represent a spectral layer containing a spectral association and a spectral convolution. In each combined module, the weighted output feature can be calculated as follows:

$$\boldsymbol{X}_{\text{out}} = \sum_{\phi \in \Phi} \frac{\exp\{\alpha^{\phi}\}}{\sum_{\phi' \in \Phi} \exp\{\alpha^{\phi'}\}} \cdot \phi(\boldsymbol{X}_{\text{in}}) \tag{15}$$

where ϕ denotes an operation in $\mathbf{\Phi}$ and α^{ϕ} represents the trainable weight of each unit in a combined module. In (15), $\mathbf{\Phi}$ represents the operator space shown in Fig. 3(a), including all possible candidate operators. Then, the output tensor X_{out} of a layer-level unit in a transformer block is calculated by the weighted sum of intermediate tensors over all operators in $\mathbf{\Phi}$. The weight for each operator equals $\exp\{\alpha^{\phi}\}$ in which α is a trainable architecture parameter and is divided by a normalized term equals $\sum_{\phi'} \exp\{\alpha^{\phi'}\}$. In this way, we are able to compare two families of arch settings instead of just two fixed models.

As shown in Fig. 3(b), the $\mathbf{\Theta}$ search space is adopted to decide block-level sequential order, in which "A" and "E" denote spatial and spectral blocks, respectively. For example, SSRN [1] takes the sequential order of "EEAA." We limit the block-level search of FAS to operating only on the six candidates shown in $\mathbf{\Theta}$ space. In addition, two baselines ("AAAA" and "EEEE") are used in the ablation study for validating the block-level network settings. In Section III-D, we will describe the proposed FAS strategy in detail.

D. Factorized Architecture Search

Previous works have demonstrated that NAS suffers from unstable training caused by the gap between the super networks used in the bilevel optimization and their derived networks. To avoid this gap, as shown in Fig. 1, we introduce an FAS strategy that limits architecture search only in the searching stage, focusing on finding out the optimal setting of one factor in an FAS subprocedure by fixing other factors. Specifically, the proposed FAS framework involves two subprocesses in two factorized spaces introduced in the above subsection, thus enabling an effective and efficient search in each space.

First, we use the sequential order of SSRN as the initial architecture ("EEAA" in $\mathbf{\Phi}$) of super network. Following [1], we use two consecutive spectral and spatial blocks to learn discriminative spectral–spatial features. Then, we conduct an FAS process (FAS1 in Fig. 1) to discover the optimal combination of operations in $\mathbf{\Phi}$ space, which determines the layer-level setting. The optimization step and the objective of FAS1 can be formulated as follows:

$$\boldsymbol{\omega} := \boldsymbol{\omega} - \gamma_1 \nabla_{\boldsymbol{\omega}} \mathcal{L}\big(\boldsymbol{y}^{\text{Dev}}, \text{Net}\big(\boldsymbol{X}^{\text{Dev}}; \boldsymbol{\phi}, \bar{\boldsymbol{\theta}}, \boldsymbol{\omega}\big)\big) \tag{16}$$

$$\boldsymbol{\phi}^* = \underset{\tilde{\boldsymbol{\phi}}}{\text{argmin}}\, \mathcal{L}\big(\boldsymbol{y}^{\text{Val}}, \text{Net}\big(\boldsymbol{X}^{\text{Val}}; \tilde{\boldsymbol{\phi}}, \bar{\boldsymbol{\theta}}, \boldsymbol{\omega}\big)\big) \tag{17}$$

where γ_1 represents the learning rate of the FAS1 process, in which we use a fixed $\bar{\boldsymbol{\theta}}$ which is the same as SSRN. We search for the optimal spectral and spatial units in $\mathbf{\Phi}$ space to build spectral and spatial blocks, respectively.

Then, we adopt another FAS process (FAS2 in Fig. 1) to figure out the optimal setting in $\mathbf{\Theta}$ space, which decides the sequential orders of spectral and spatial blocks in SSTN. Similarly, the optimization step and the objective of FAS2 can be formulated as follows:

$$\boldsymbol{\omega} := \boldsymbol{\omega} - \gamma_2 \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{y}^{\text{Dev}}, \big(\text{Net}\big(\boldsymbol{X}^{\text{Dev}}; \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}, \boldsymbol{\omega}\big)\big) \tag{18}$$

$$\boldsymbol{\theta}^* = \underset{\tilde{\boldsymbol{\theta}}}{\text{argmin}}\, \mathcal{L}\big(\boldsymbol{y}^{\text{Val}}, \text{Net}\big(\boldsymbol{X}^{\text{Val}}; \hat{\boldsymbol{\phi}}, \tilde{\boldsymbol{\theta}}, \boldsymbol{\omega}\big)\big) \tag{19}$$

where γ_2 represents the learning rate of the FAS2 procedure, in which we use a fixed $\hat{\boldsymbol{\phi}}$ found in (17).

As shown in Fig. 1, both FAS1 and FAS2 processes are trained on $\boldsymbol{X}^{\text{Dev}}$ and validated on $\boldsymbol{X}^{\text{Val}}$. We combine the searched optimal architecture settings $\boldsymbol{\phi}^*$ and $\boldsymbol{\theta}^*$ in these two disentangled spaces. Using these searched settings, we finally train a network from scratch on $\boldsymbol{X}^{\text{Tra}}$ and test the trained model on $\boldsymbol{X}^{\text{Tes}}$. Compared to various NAS methods, the FAS framework presents a stable searching process by imposing constraints on the two factorized search spaces and decoupling the stages of searching and training.

E. Spectral–Spatial Transformer Network

We employ the progressive FAS framework in $\mathbf{\Phi}$ and $\mathbf{\Theta}$ spaces to determine network architectures, rather than searching the whole network via a bilevel optimization like NAS. We determine the settings of SSTN according to empirical evidence of FAS on three HSI datasets. In the $\mathbf{\Phi}$ space, the spectral and spatial transformer blocks employ spectral association and spatial attention modules as fundamental building operations, respectively. Also, we set the block sequential order of SSTN to be "AEAE" in the $\mathbf{\Theta}$ space. This progressive FAS framework conducted in factorized search spaces avoids the unstable training of previous NAS methods.

1) Final Architecture: The final architecture follows a block sequence of "AEAE" as shown in Fig. 3(c), where "A" and "E" denote spatial transformer and spectral transformer blocks, respectively. The spatial transformer block comprises two spatial attention modules with a skip connection that follows the classic design of the residual block in ResNet [11]. Similarly, the spectral counterpart contains two spectral association units. We use an HSI cuboid from the University of Pavia (UP) dataset as an input to demonstrate the SSTN structure. As shown in Fig. 3(c), the input HSI sample has a size of $103 \times 9 \times 9$. The first Conv layer is used to reduce the spectral dimensions of input feature X from 103 to 49. The first pair of spatial–spectral ("AE") transformer blocks keep the spectral dimension unchanged and output an intermediate feature tensor X'. Then, the second Conv layer reduces the spectral size from 49 to 18. The second pair of spatial–spectral transformer blocks function similar to the first one. As for the hyperparameters, we set the ratio of c/c' in (7) and (8) in the spatial attention module to be 8 and spectral association dimension k in (11) to be 18. Finally, an average pooling layer and a fully connected layer generate the classification logits Y_{pred}. In the following section, we describe the implementation details of FAS and report the qualitative and quantitative outcomes of experiments.

IV. Results and Discussion

In this section, we first describe the three HSI benchmark datasets, then introduce the configurations for both the searching and training stages of FAS, and finally assess the SSTN using qualitative metrics, including overall accuracy (OA), average accuracy (AA), kappa coefficient ($\mathcal{K}$), training time, and testing time. We carry out network searching on two factorized architecture spaces $\mathbf{\Phi}$ and $\mathbf{\Theta}$ to demonstrate the effectiveness of the FAS strategy. In addition, we report the parameter numbers of SSTN and other SOTA networks used in comparison experiments to evaluate their computational expenses.

A. HSI Datasets

We evaluate the proposed SSTN and the FAS framework using three challenging HSI benchmarks, including the Indian Pines (IN), the Kennedy Space Center (KSC), UP, the University of Houston (UH), and the Pavia Center (PC) datasets. Fig. 8 shows the imagery of IN dataset, which includes 16 vegetation categories and has 145×145 pixels with 200 hyperspectral bands. Fig. 9 shows the imagery of KSC dataset that involves 13 wetland classes and has 512×614 pixels with 176 hyperspectral bands. Fig. 10 shows the imagery of the UP dataset that contains nine urban land cover classes and has 610×340 pixels with 103 bands. Fig. 11 shows the imagery of UH dataset that contains 15 urban land cover classes and has 349×1905 pixels with 144 bands [46]. Fig. 12 shows

the imagery of PC dataset that contains nine classes and has 1096 × 715 pixels with 102 bands. As for the IN dataset, we ensure that each land cover category contains at least one sample for all sets to avoid the case that no HSI cuboids are sampled for rare classes.

We randomly sample 200 develop and 400 validate HSI cuboids with their annotations in each dataset for architecture searching. Also, we randomly select 200 samples for network training and use the remaining cuboids for testing. Besides, all HSI cuboids of three datasets are normalized by subtracting mean values and then being divided by max values. Tables II–IV list the sampled numbers of train, develop, and validate groups in three HSI datasets, respectively.

B. Framework Setting

1) Implementation Details: In the architecture searching stage, as shown in Fig. 3(a), we set the layer-level contains two options for each composition setting in the $\mathbf{\Phi}$ space. Then, we employ the warm-up mode for the first 15 epochs and search for 100 epochs in total. We use the Adam optimizer. We set the learning rates to 0.01, 0.02, 0.02 in the searching stage for IN, KSC, and UP datasets. The momentum and weight decay are set to 0.9 and $3e^{-4}$, respectively. In the network training stage, we adopt the optimal architecture settings shown in Fig. 3(a) and (b). We also use the Adam optimizer in this stage with a learning rate equals to 0.002 on four HSI datasets, except 0.001 for the PC dataset. We train all methods in comparison experiments for 300 epochs. We set the batch size to 50 for both the searching and training stages. We run five times for all experiments and report their mean values of different metrics.

As shown in Fig. 1, the block-level sequential order of super network $\text{Net}(\boldsymbol{X}; \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\omega})$ is internalized to "EEAA" that is the same as that of SSRN [1]. We use the super network to explore the optimal network settings for learning spectral and spatial features in the $\mathbf{\Phi}$ space. Then, we search for the optimal block-level sequential order of SSTN in the $\mathbf{\Theta}$ space. Fig. 3 takes an HSI cuboid as input and shows an SSTN with a sequential order of "AEAE" for HSI classification. To make a fair comparison, the spatial size of HSI cuboids is 9 × 9 for all networks on different datasets. In the following paragraphs, we studied three aspects that affect the FAS framework and the recognition performance of SSTNs.

First, we focus on the layer-level searching in the $\mathbf{\Phi}$ space, as shown in Fig. 3(a). We restrict the search space to containing three spatial and three spectral layer-level combinations to avoid the high computational costs of NAS. Using the block-level sequential order of SSRN, we consider five combinations of spectral and spatial operations: "AccEcc," "AacEcc," "AaaEcc," "AaaEac," and "AaaEaa." We compare the searching curves of networks with the five settings on three HSI datasets in the next subsection. These results are obtained during a 100-epoch architecture search in a subprocedure of the FAS framework (FAS1) on randomly sampled 200 develop cuboids $\boldsymbol{X}^{\text{Dev}}$ and 400 validate samples $\boldsymbol{X}^{\text{Val}}$. Thus, we attain a more robust evaluation of the network settings than the traditional grid-search strategy by decoupling architecture searching and network training.

Second, we explore the block-level sequential orders of SSTN in the $\mathbf{\Theta}$ space, as shown in Fig. 3(b). We constrain the search space to enclose only six sequential settings, including "AEAE," "AAEE," "AEEA," "EAEA," "EEAA," and "EAAE." In this way, we extend the search space of conventional NAS by incorporating the expert knowledge that block-level sequential order matters for classification performance. We report the OA of the six block-level configurations to determine the architecture of SSTN. The classification outcomes are produced from a 100-epoch architecture search in another FAS subprocedure (FAS2) using the same hyperparameters as FAS1. Therefore, we employ FAS as a method to justify the layer- and block-level settings of SSTN, rather than searching for network settings without imposing constraints on the search space.

Third, we carry out an ablation study by comparing networks with all spectral or all spatial blocks ("EEEE" or "AAAA") to validate the effectiveness of the FAS strategy and the searched sequential order of SSTN. Also, we adopt the sequential order of SSRN ("EEAA") as a strong baseline. According to the two subprocedures of the FAS framework introduced in the above two paragraphs, we chose the two best performing settings of SSTN as our candidates for comparison. In the following subsection, we reported the OA of networks with five different block orders on three HSI datasets to assess the generalizability of SSTN and also validate the FAS framework.

C. Ablation Studies

We first presented the experiments of two factorized architecture spaces $\mathbf{\Phi}$ and $\mathbf{\Theta}$ in the FAS framework on three datasets to determine the layer- and block-level settings for SSTN. Based on the searched settings, we then conducted an ablation study regarding the sequential orders of SSTN to validate the efficacy of FAS. Next, we compared the SSTN using searched settings to expert-designed networks, including CONV [47], spatial transformation network (STN) [48], spectral attention module-based convolutional network (SPA) [29], SSAN [16], and SSRN [1]. Also, we compared SSTN to an NAS approach automatic convolutional neural network (AUTO) [24] for HSI classification to show the generalizability of FAS. We evaluated these networks qualitatively using their classification maps. For a fair comparison, we ensure all models to have two spatial and two spectral blocks. Then, we trained 200 epochs for all networks and set the input HSI cuboids with the size of $L \times 9 \times 9$, where L represents the number of hyperspectral bands.

To decide the operations used for spectral and spatial blocks, we recorded the OA of networks with the same sequential order as SSRN using different layer-level settings in the $\mathbf{\Phi}$ space. The classification outcomes are calculated on the validate HSI set from 50 to 100 epochs in an interval of ten epochs during the first searching stage of FAS (FAS1). As shown in Fig. 4, the layer setting of "AaaEaa" obtains comparable accuracy to that of "AaaEcc" in IN dataset and outperforms other layer-level settings in KSC and UP datasets. The setting of "AaaEaa" represents that the network adopts

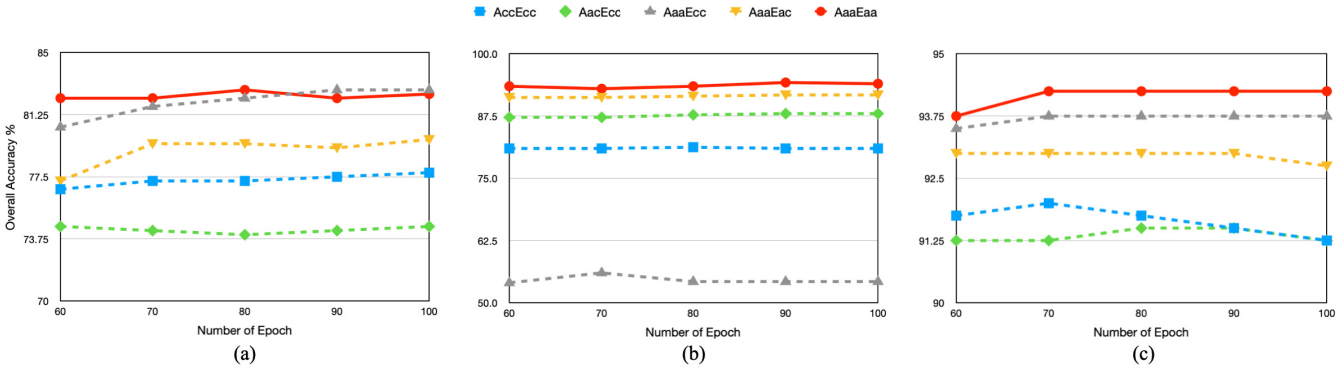


Fig. 4. OA of networks that have a sequential order of "EEAA" with different layer-level settings in the $\mathbf{\Phi}$ space on the validate set from 50 to 100 epochs in an interval of ten epochs during the FAS1 searching stage. (a) IN dataset. (b) KSC dataset. (c) UP dataset.

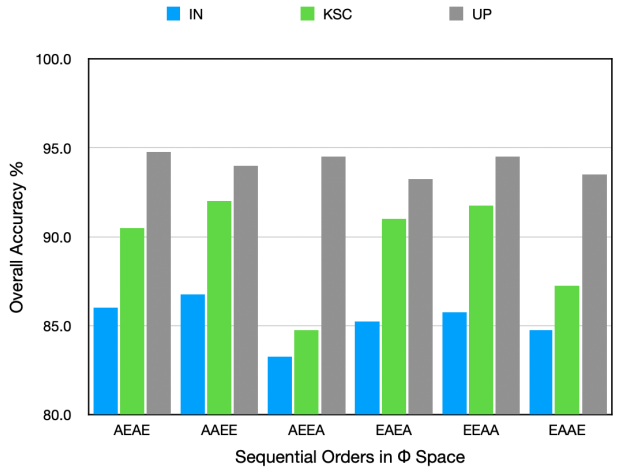


Fig. 5. OA of different architecture of SSTN in the $\mathbf{\Theta}$ Space on the validate samples in the FAS2 searching stage on IN, KSC, and UP datasets.

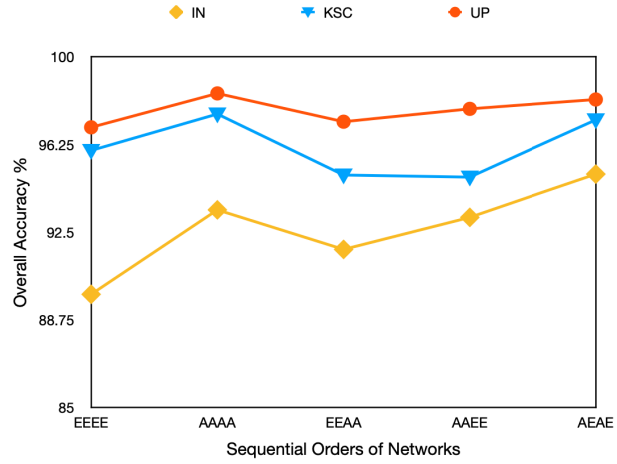


Fig. 6. Ablation study on three HSI benchmarks with different sequential order settings of SSTN, including "EEEE," "AAAA," "EEAA," "AAEE," and "AEAE," where "E" represents a spectral block and "A" denotes a spatial block.

spectral association and spatial attention for building spectral and spatial transformer blocks, respectively. Also, these results demonstrate the effectiveness of spatial attention and spectral

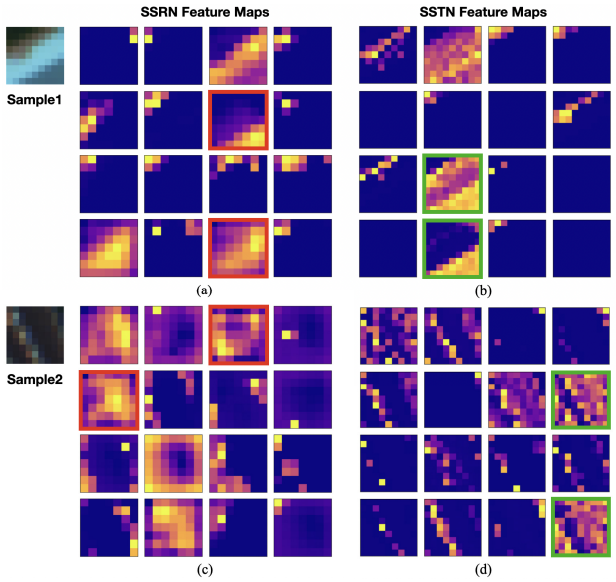


Fig. 7. Activation maps before the final linear layers the trained SSRN and SSTN generated by two HSI samples from the UP dataset. (a)–(b) Activation maps of SSRN and SSTN generated by the first sample. (c)–(d) Activation maps of SSRN and SSTN generated by the second sample. We highlight typical maps with green and red boxes.

association modules compared to their convolution counterparts for learning discriminant spectral–spatial features.

To determine the block-level settings of SSTN, we reported the OA of networks using different sequential orders in the $\mathbf{\Theta}$ space. The classification results are also computed using the validate HSI samples in the second searching stage of FAS (FAS2). As shown in Fig. 5, the block-level sequential order of "AEAE" achieves the highest OA (94.75%) on the validate set of UP dataset, while the "AAEE" setting obtains the best OA (86.75% and 92.00%) on the validate groups of IN and KSC datasets. Given that these two block-level settings clearly outperform other competitors on all three benchmarks, we adopt both as the network setting candidates of SSTN in the following ablation study.

Next, we conducted an ablation experiment using different block-level sequential orders of SSTN on three datasets. We compared our two candidates ("AAEE" and "AEAE")

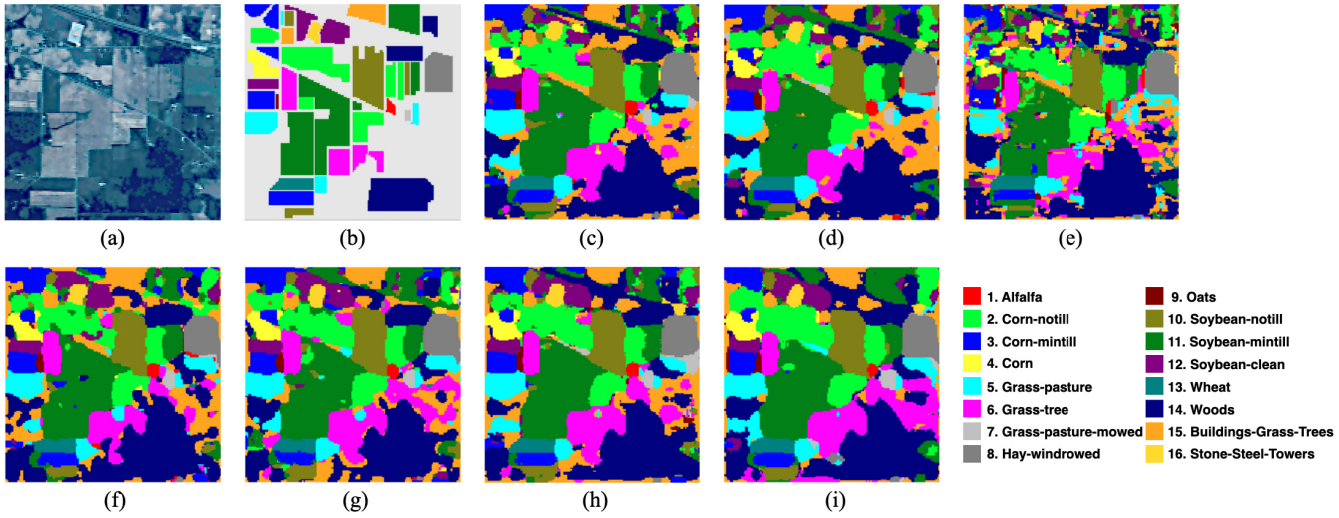


Fig. 8. Classification results of different models on the IN dataset. (a) False-color map. (b) Ground truth map. (c)–(i) Classification maps of CONV, STN, SPA, SSAN, SSRN, AUTO, and SSTN.

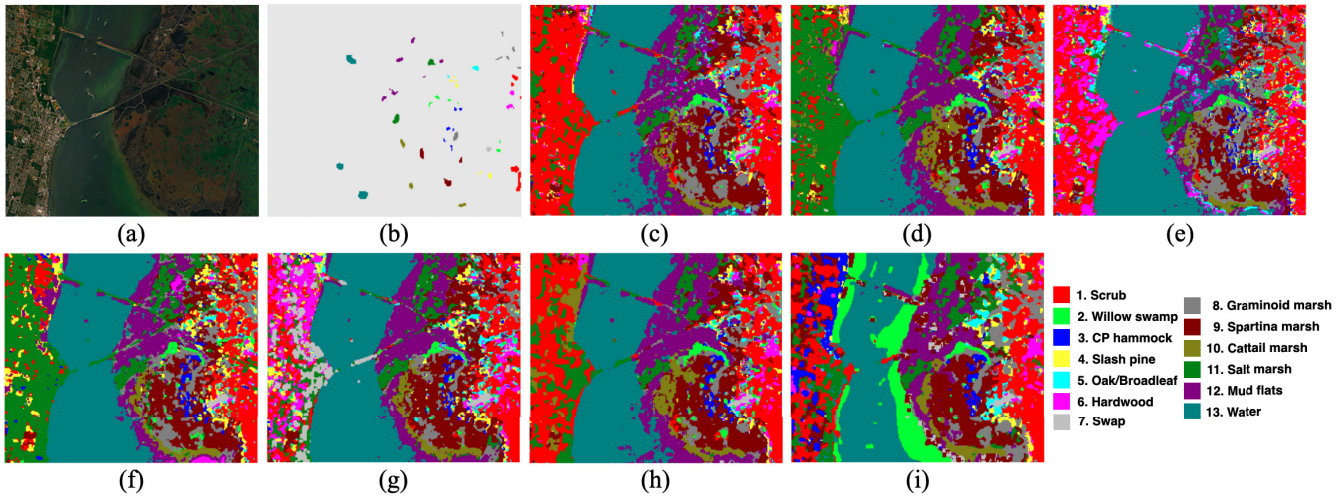


Fig. 9. Classification results of different models on the KSC dataset. (a) False-color map. (b) Ground truth map. (c)–(i) Classification maps of CONV, STN, SPA, SSAN, SSRN, AUTO, and SSTN.

searched by the FAS framework to three baselines ("EEEE," "AAAA," and "EEAA"), in which "E" denotes a spectral transformer block consisting of two spectral association units and "A" represents a spatial transformer block containing two spatial attention modules. As shown in Fig. 6, the block-level setting of "AEAE" yields similar classification performance compared to the top ones produced by the baseline setting "AAAA" on both KSC and UP datasets while generating the best OA on the IN dataset. Therefore, we finalize the SSTN using the sequential order of "AEAE" because the spatial attention unit consumes much more computational expense than its spectral association counterpart.

1) Activation Maps: We add a qualitative experiment to visualize the activation maps of SSTN and those of SSRN for justifying the long-range interaction between learned features. As shown in Fig. 7, we use two different HSI samples from the UP dataset as inputs to compare the activations maps before the final linear layers of the trained SSTN and SSRN. In both cases, we can see relatively bright positions distribute evenly across the activation maps of SSTN (green boxes) rather than concentrate on small areas in those of SSRN (red boxes), which is caused by the grid structure of convolutional kernels. Also, these findings are in line with the illustrations in Fig. 2, in which the attention maps (left) are the matrix multiplication outcome of, and association kernels (right) connect to all positions in feature maps. Therefore, our proposed modules can largely address the limitation of standard convolution via emphasizing the interactions between features regardless of their spatial distance.

In addition, we have tested SSTNs with different numbers of training samples, ranging from 200 to 400, on three HSI datasets and presented the results in Fig. 13. This figure shows that SSTNs deliver better performance when using more training samples on all three datasets while consistently

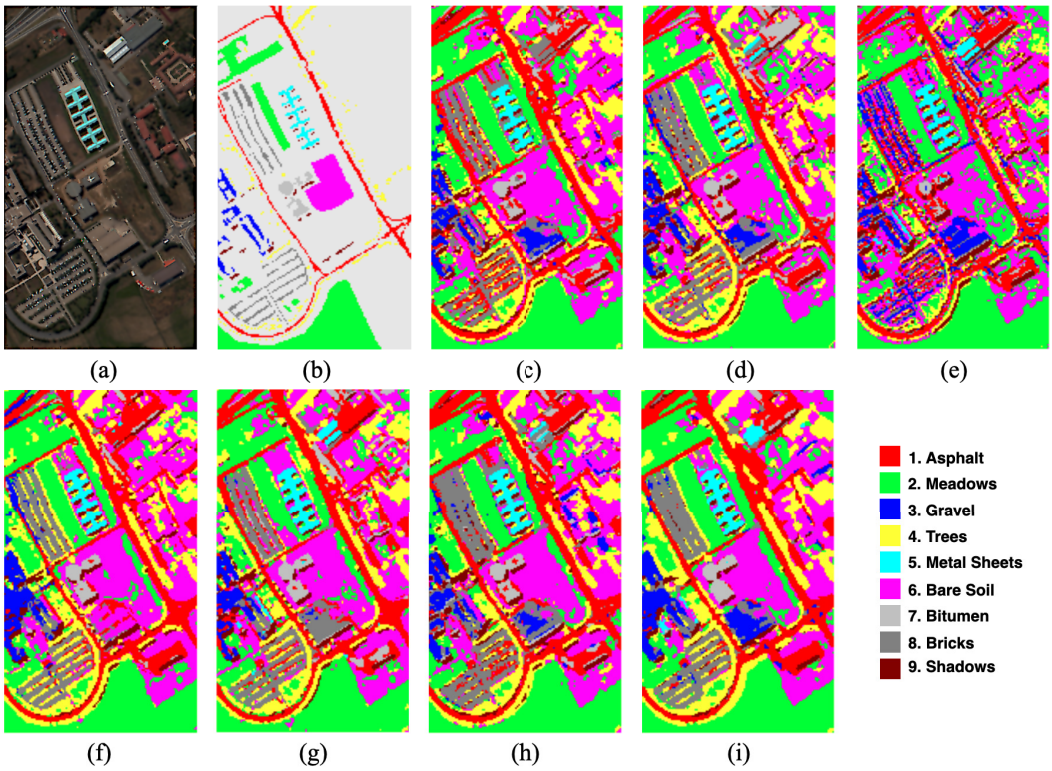


Fig. 10. Classification results of different models on the UP dataset. (a) False-color map. (b) Ground truth map. (c)–(i) Classification maps of CONV, STN, SPA, SSAN, SSRN, AUTO, and SSTN.

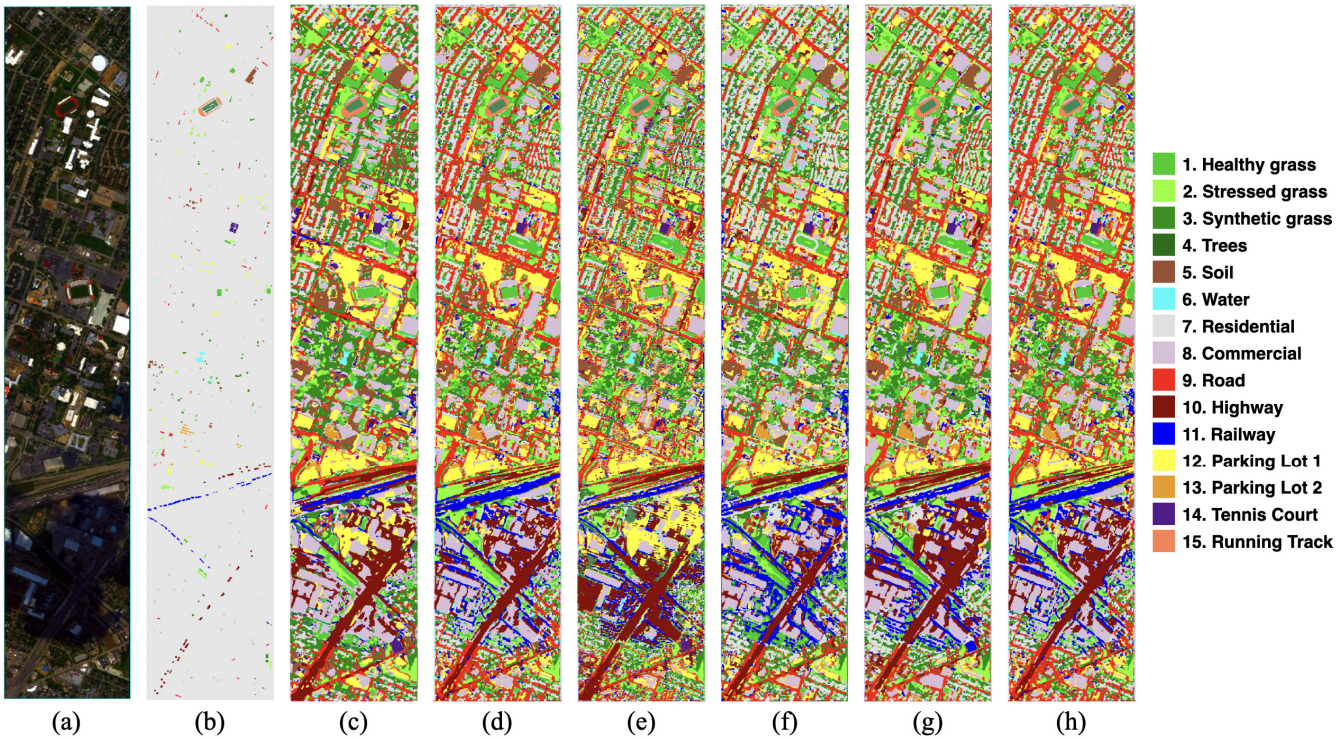


Fig. 11. Classification results with OA in parenthesis of different models on the UH dataset. (a) False-color map. (b) Ground truth map. (c)–(h) Classification maps of CONV (87.41%$_{\pm 1.3\%}$), STN (91.93%$_{\pm 0.8\%}$), SPA (87.41%$_{\pm 2.2\%}$), SSAN (90.70%$_{\pm 1.3\%}$), SSRN (92.33%$_{\pm 1.4\%}$), and SSTN (91.95%$_{\pm 1.3\%}$).

outperforming SSRNs. These results justify the efficiency and robustness of SSTNs, which possesses a small model size compared to strong baselines. Therefore, according to empirical results shown in Fig. 13 and Table I, it is reasonable to extrapolate that SSTNs can be extended to other challenging scenarios.

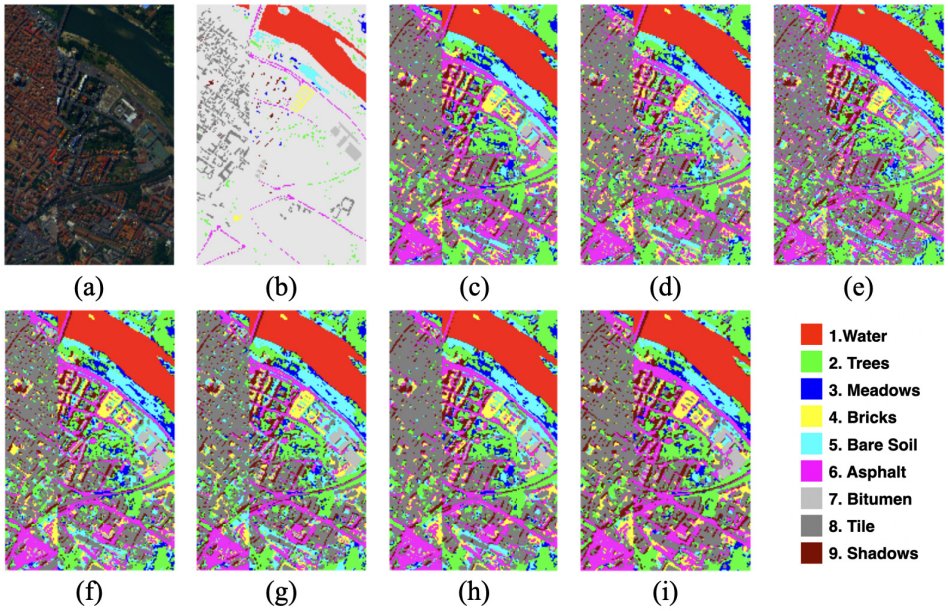


Fig. 12. Classification results with OA in parenthesis of different models on the PC dataset. (a) False-color map. (b) Ground truth map. (c)–(i) Classification maps of CONV (97.42%$_{\pm 0.3\%}$), STN (98.20%$_{\pm 0.5\%}$), SPA (97.87%$_{\pm 0.4\%}$), SSAN (98.42%$_{\pm 0.3\%}$), SSRN (98.155%$_{\pm 0.2\%}$), AUTO (98.35%$_{\pm 0.2\%}$), and SSTN (98.96%$_{\pm 0.2\%}$).

TABLE II
CLASSIFICATION RESULTS OF DEEP LEARNING MODELS USING 200 TRAIN, 200 DEVELOP, AND 400 VALIDATE HSI SAMPLES ON THE IN DATASET

Class	Train	Dev	Val	CONV	STN	SPA	SSAN	SSRN	AUTO	SSTN
1	2	1	3	95.00	**97.50**	42.50	95.00	92.50	35.24	72.50
2	27	28	55	85.36	88.77	78.45	80.12	89.61	85.80	**96.05**
3	19	16	35	85.66	84.47	69.21	85.79	84.87	76.16	**87.76**
4	4	5	9	84.93	78.08	47.95	89.95	93.61	53.60	**90.87**
5	9	9	18	88.59	89.04	76.29	86.58	**88.91**	73.48	85.46
6	14	14	28	96.74	88.28	83.38	90.80	90.80	94.98	**97.18**
7	2	1	6	**100.0**	**100.0**	45.45	**100.0**	**100.0**	54.62	**100.0**
8	10	9	19	99.54	98.41	91.82	95.91	98.64	99.53	**100.0**
9	3	1	4	**100.0**	**100.0**	83.33	**100.0**	**100.0**	10.67	**100.0**
10	24	19	43	84.20	83.18	75.40	86.23	89.95	91.98	**92.66**
11	41	47	88	88.20	90.26	78.46	91.75	91.14	92.11	**97.41**
12	14	12	26	56.19	52.31	42.33	79.11	87.43	71.97	**87.80**
13	4	4	8	**100.0**	**100.0**	**100.0**	98.94	97.35	93.64	**100.0**
14	18	24	42	96.87	99.75	93.99	97.80	**99.15**	97.15	98.81
15	7	8	15	83.15	71.63	52.81	**91.29**	86.51	51.11	82.58
16	2	2	4	84.71	91.76	90.59	91.76	**98.82**	72.40	95.29
All	OA (%)			87.64$_{\pm 1.4}$	87.41$_{\pm 0.8}$	76.81$_{\pm 0.8}$	89.17$_{\pm 1.6}$	91.43$_{\pm 2.0}$	87.79$_{\pm 0.9}$	**94.39$_{\pm 0.5}$**
	AA (%)			89.32$_{\pm 2.0}$	88.34$_{\pm 1.3}$	72.00$_{\pm 3.0}$	91.32$_{\pm 2.5}$	**93.08$_{\pm 1.3}$**	72.15$_{\pm 1.4}$	92.77$_{\pm 0.9}$
	$\mathcal{K} \times 100$			85.88$_{\pm 1.6}$	85.57$_{\pm 1.0}$	73.51$_{\pm 1.0}$	87.62$_{\pm 1.8}$	90.21$_{\pm 2.4}$	85.73$_{\pm 1.0}$	**93.58$_{\pm 0.6}$**
	Training time (s)			558.9$_{\pm 95.1}$	512.7$_{\pm 40.1}$	591.5$_{\pm 40.8}$	659.4$_{\pm 66.9}$	623.5$_{\pm 17.1}$	758.7$_{\pm 83.1}$	587.9$_{\pm 60.0}$
	Testing time (s)			1.99$_{\pm 0.3}$	1.82$_{\pm 0.1}$	1.98$_{\pm 7.9}$	2.04$_{\pm 0.2}$	2.14$_{\pm 0.1}$	2.79$_{\pm 0.3}$	2.07$_{\pm 0.1}$

D. Comparison With State of the Art

Tables II–IV record the quantitative classification results, including OA for each class, OA, AA, and Kappa coefficients for all classes, generated by different networks on three HSI datasets. On the IN dataset, SSTN achieved the highest OA (94.39%) that outperforms SSRN by (2.96%). SSTN also yielded the best OA (97.30%) that surpasses SSRN by (0.56%) on the KSC dataset. On the UP dataset, SSTN obtained superior performance to other competing methods in all three metrics. Figs. 8–10 show the classification maps generated by all employed networks on three datasets. These qualitative outcomes are in line with quantitative ones. Given only 200 samples for training, it is worth noting that SSTN generates much clearer boundaries for the water class in the river than other networks, which sheds light on the possible application of SSTN to other related vision tasks. These results also demonstrate the effectiveness and generalizability of SSTNs, depending on the spectral–spatial feature learning capacity of spectral association and spatial attention modules.

TABLE III
CLASSIFICATION RESULTS OF DEEP LEARNING MODELS USING 200 TRAIN, 200 DEVELOP, AND 400 VALIDATE HSI SAMPLES ON THE KSC DATASET

Class	Train	Dev	Val	CONV	STN	SPA	SSAN	SSRN	AUTO	SSTN
1	29	29	58	**100.0**	**100.0**	97.83	98.91	**100.0**	98.77	**100.0**
2	9	10	19	75.61	77.56	69.76	75.12	80.49	88.80	**93.72**
3	10	10	20	93.98	**100.0**	75.00	99.54	99.07	86.78	99.07
4	10	9	19	49.53	84.11	70.09	97.66	**99.07**	89.18	80.19
5	6	6	12	78.10	15.33	59.85	80.29	**99.27**	95.40	81.02
6	9	9	18	91.19	92.22	72.54	66.32	88.08	**98.22**	94.82
7	4	4	8	98.88	**100.0**	**100.0**	**100.0**	**100.0**	98.63	89.89
8	17	16	33	94.25	91.78	93.70	98.63	92.05	94.99	**99.45**
9	20	20	40	**100.0**	**100.0**	99.31	99.77	**100.0**	**100.0**	99.32
10	16	15	31	82.46	99.12	88.01	96.78	96.78	**100.0**	**100.0**
11	16	16	32	**100.0**	98.03	98.31	99.44	100.0	**100.0**	**100.0**
12	19	20	39	84.94	**100.0**	92.24	88.00	91.53	99.23	97.19
13	35	36	71	98.22	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
All	OA (%)			$91.43_{\pm 1.6}$	$94.31_{\pm 1.0}$	$90.75_{\pm 2.4}$	$94.88_{\pm 0.1}$	$96.74_{\pm 1.1}$	$96.90_{\pm 0.7}$	$\mathbf{97.30_{\pm 1.1}}$
	AA (%)			$88.24_{\pm 2.4}$	$89.09_{\pm 1.6}$	$85.90_{\pm 3.5}$	$92.34_{\pm 1.7}$	$95.87_{\pm 1.5}$	$\mathbf{96.29_{\pm 1.4}}$	$94.97_{\pm 2.0}$
	$\mathcal{K} \times 100$			$90.46_{\pm 1.8}$	$93.66_{\pm 1.1}$	$89.70_{\pm 2.7}$	$94.29_{\pm 1.0}$	$96.37_{\pm 1.2}$	$96.54_{\pm 0.8}$	$\mathbf{96.99_{\pm 1.2}}$
	Training time (s)			$336.9_{\pm 15.0}$	$348.7_{\pm 3.65}$	$347.3_{\pm 5.0}$	$405.6_{\pm 67.8}$	$439.7_{\pm 11.5}$	$524.8_{\pm 38.1}$	$401.5_{\pm 57.4}$
	Testing time (s)			$1.06_{\pm 0.1}$	$1.16_{\pm 0.1}$	$1.10_{\pm 0.1}$	$1.17_{\pm 0.1}$	$1.22_{\pm 0.1}$	$1.99_{\pm 0.1}$	$1.30_{\pm 1.7}$

TABLE IV
CLASSIFICATION RESULTS OF DEEP LEARNING MODELS USING 200 TRAIN, 200 DEVELOP, AND 400 VALIDATE HSI SAMPLES ON THE UP DATASET

Class	Train	Dev	Val	CONV	STN	SPA	SSAN	SSRN	AUTO	SSTN
1	31	31	62	93.64	91.82	87.49	94.74	98.46	94.14	**98.17**
2	88	86	174	99.34	99.33	95.96	99.77	98.52	92.78	**99.96**
3	10	10	20	82.61	90.04	82.42	82.22	74.60	80.60	**89.02**
4	13	16	29	88.26	92.51	77.98	92.48	**95.18**	83.42	92.95
5	8	5	26	99.92	99.85	100.0	100.0	**100.0**	99.13	**100.0**
6	23	24	47	94.51	93.62	87.78	96.92	97.93	95.62	**98.97**
7	9	3	24	85.38	95.02	66.00	98.70	99.54	87.31	**100.0**
8	14	20	68	93.83	95.52	41.92	93.80	96.90	**98.39**	95.10
9	4	5	18	99.25	**100.0**	98.71	99.78	99.57	45.45	95.80
All	OA (%)			$95.38_{\pm 0.9}$	$96.11_{\pm 0.7}$	$86.33_{\pm 1.6}$	$96.73_{\pm 0.3}$	$96.99_{\pm 0.9}$	$93.88_{\pm 0.8}$	$\mathbf{98.02_{\pm 0.5}}$
	AA (%)			$92.97_{\pm 1.5}$	$95.30_{\pm 1.3}$	$82.03_{\pm 1.5}$	$95.38_{\pm 0.7}$	$95.63_{\pm 1.8}$	$87.17_{\pm 1.1}$	$\mathbf{96.67_{\pm 1.0}}$
	$\mathcal{K} \times 100$			$93.85_{\pm 1.2}$	$94.84_{\pm 1.0}$	$77.98_{\pm 2.0}$	$95.66_{\pm 0.4}$	$96.01_{\pm 1.2}$	$92.15_{\pm 1.1}$	$\mathbf{97.37_{\pm 0.6}}$
	Training time (s)			$1012.3_{\pm 101.3}$	$1139.9_{\pm 115.4}$	$1146.3_{\pm 99.1}$	$1311.5_{\pm 74.9}$	$1368.6_{\pm 23.5}$	$1363.0_{\pm 37.9}$	$1449.4_{\pm 60.1}$
	Testing time (s)			$3.79_{\pm 0.3}$	$4.77_{\pm 0.3}$	$4.67_{\pm 0.5}$	$5.17_{\pm 0.5}$	$4.59_{\pm 0.3}$	$5.90_{\pm 0.5}$	$6.08_{\pm 0.3}$

It is worth noting that SSTN outperforms AUTO in three HSI datasets, which validates the efficacy of the FAS framework.

Furthermore, we use the UH dataset collected in 2013 [46] and PC dataset with 400 HSI training samples to test the generalizability of the proposed SSTNs following the architecture in Fig. 3, respectively. As shown in Fig. 10, the SSTN achieves comparable OA (91.95%) to SOTA methods (e.g., 92.33% obtained by SSRN) on the UH dataset. As shown in Table I, it is noteworthy that the SSTN contains only 1.2% MACs (5.8 M) compared to those of SSRN (150.9 M). In Fig. 11, the SSTN obtains the best OA (98.96%) among all methods, with only 1.2% MACs of SSRN. The classification maps generated by different methods are in line with quantitative results, which can be clearly observed with the classes of synthetic grass and running track in Fig. 10 and bricks in Fig. 11.

In this study, we conducted all experiments with the PyTorch Framework using an NVIDIA TITAN V100 graphics card. As shown in Tables II–IV, the training and testing times of SSTNs are not the fastest among all models. The main reason is that we adopt naïve implementation using PyTorch modules or basic operations with no specific computational optimization. However, as reported in Table I, the computational complexity and the number of parameters of SSTNs are much more efficient (possess less or equal than 1.2% MACs of SSRNs) than other SOTA methods on five HSI datasets, and this tremendous advantage justifies the efficiency of our SSTNs. Therefore, we firmly believe that the computational costs can be furthered reduced if low-level implementations of spectral association modules are available.

E. Discussion

Inspired by previous pinioning works [16], [21], [23], we design a transformer network mainly consisting of attention modules that account for the characterizes of HSI. The excellent recognition performance of SSTNs on three datasets challenges the prevalence of spatial and spectral convolution layers for learning spectral–spatial features.

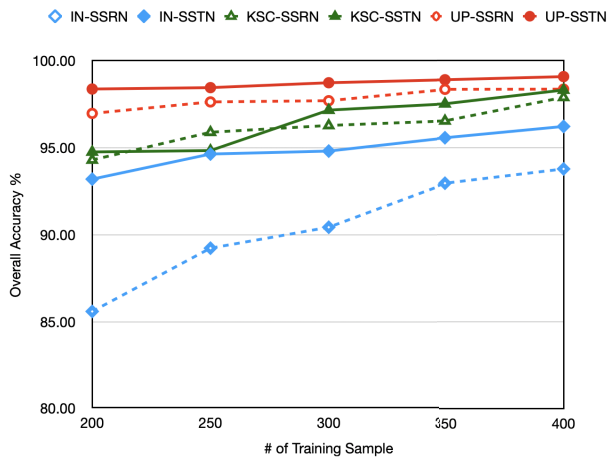


Fig. 13. OA of SSTNs and SSRNs using varying numbers of training samples, ranging from 200 to 400 with an interval of 50, on IN, KSC, and UP datasets.

Spatial convolution connects local regions by aggregating all channels of input features with trainable weights of spatial kernels. Instead, the proposed spectral association provides a solution to overcome the geometric constraints of traditional grid-structure convolution. As shown in Fig. 2, the spectral associate kernels are generated by integrating all spatial locations of masked feature maps.

The FAS strategy is motivated by two reasons. First, typical NAS methods are memory-intensive and computationally complex. Thus, we aim to design an efficient FAS strategy via factorizing its search spaces into independent factors. This improvement essentially solves the problems of the time-consuming NAS strategy. Second, the other purpose of this FAS is to explore a principled mechanism of deciding meta-parameter network settings instead of relying on human expertise or a simple grid-search method. Therefore, this work focuses on designing a lightweight transformer for HSI analysis with an efficient network searching strategy to mitigate the designing problems of existing deep learning models for HSI classification.

We gain three insights from the proposed FAS strategy. First, the success of the FAS strategy lies in decoupling the stages of searching and training. Specifically, we adopt FAS as an independent justification method to search and determine optimal network settings rather than entangling the searching and training stages. Second, the proposed spectral association module is complementary to spatial attention as well as spatial convolution. The spectral association outperforms the spectral convolution in SSRN and is much more computationally efficient. Third, the search space $\mathbf{\Theta}$ of sequential orders in FAS enables a constrained but stable space for architecture search, which is caused by relaxing the differential requirements of super networks in NAS.

V. Conclusion

In this article, we have discovered SSTN using an FAS framework that combines the wisdom of NAS and experts' domain knowledge on three HSI datasets. To reduce the computational burden of NAS, we relax the differentiable requirements of search spaces to allow only a few layer- and block-level choices. Therefore, the FAS strategy overcomes NAS's shortcomings via factorizing the search space into two independent discrete subspaces, each of which involves layer-level operation combinations or block-level sequential orders.

Besides, SSTNs avoid the geometric constraints of convolution operations by adopting spatial attention and spectral association as basic building elements. The spatial attention models the pixel-to-pixel interactions of all positions, while the spectral association measures the correlation between a compact set of spectral vectors to all locations. The experimental results on three widely studied datasets demonstrate that the SSTN outperforms SOTA networks, including an automatic searched model, using much less trainable parameters. We hope that the discovered SSTN and the novel FAS strategy would facilitate applying neural networks and learning frameworks on the Earth observation data.

References

[1] 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.

[2] X. Huang and L. Zhang, "An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4173–4185, Dec. 2008.

[3] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.

[4] L. Wei and D. Qian, "Collaborative representation for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1463–1474, Mar. 2015.

[5] N. M. Nasrabadi, "Hyperspectral target detection: An overview of current and future challenges," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 34–44, Jan. 2014.

[6] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[7] 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.

[8] M. D. Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3747–3762, Oct. 2010.

[9] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 217–231, Jan. 2013.

[10] X. Jia, B.-K. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," *Proc. IEEE*, vol. 101, no. 3, pp. 676–697, Mar. 2013.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[14] 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.

[15] 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.

[16] 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.

[17] L. Zhou *et al.*, "Subspace structure regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4257–4270, Jul. 2020.

[18] J. Liang, J. Zhou, L. Tong, X. Bai, and B. Wang, "Material based salient object detection from hyperspectral images," *Pattern Recognit.*, vol. 76, pp. 476–490, Apr. 2018.

[19] J. Liang, J. Zhou, Y. Qian, L. Wen, X. Bai, and Y. Gao, "On the sampling strategy for evaluation of spectral–spatial methods in hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 862–880, Feb. 2017.

[20] Z. Zhong, J. Li, D. A. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3318–3329, Jul. 2020.

[21] X. Mei *et al.*, "Spectral–spatial attention networks for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 8, p. 963, Apr. 2019.

[22] C. Yan, X. Bai, P. Ren, L. Bai, W. Tang, and J. Zhou, "Band weighting via maximizing interclass distance for hyperspectral image classification," *IEEE Geosci. Remote Sen. Lett.*, vol. 13, no. 7, pp. 922–925, Jul. 2016.

[23] J. He, L. Zhao, H. Yang, M. Zhang, and W. Li, "HSI-BERT: Hyperspectral image classification using the bidirectional encoder representation from transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 165–178, Jan. 2020.

[24] 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, Sep. 2019.

[25] A. Vaswani *et al.*, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: http://arxiv.org/abs/1706.03762

[26] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[27] B. Fang, Y. Li, H. Zhang, and J. C.-W. Chan, "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sens.*, vol. 11, no. 2, p. 159, 2019.

[28] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 11, p. 1307, 2019.

[29] L. Mou and X. X. Zhu, "Learning to pay attention on spectral domain: A spectral attention module-based convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 110–122, Sep. 2020.

[30] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual spectral–spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 449–462, Jan. 2021.

[31] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*. [Online]. Available: http://arxiv.org/abs/1611.01578

[32] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," 2018, *arXiv:1806.09055*. [Online]. Available: http://arxiv.org/abs/1806.09055

[33] C. Peng, Y. Li, L. Jiao, and R. Shang, "Efficient convolutional neural architecture search for remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 6092–6105, Jul. 2020.

[34] H. Dong, B. Zou, L. Zhang, and S. Zhang, "Automatic design of CNNs via differentiable neural architecture search for PolSAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6362–6375, Sep. 2020.

[35] P. Ghamisi, M. D. Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2015.

[36] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[37] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[38] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017.

[39] X. Ma, H. Wang, and J. Geng, "Spectral–spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.

[40] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[41] 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.

[42] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, "Hyperspectral image classification with Markov random fields and a convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2354–2367, May 2018.

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

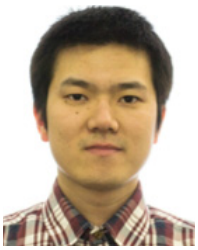
[44] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Nov. 2019, pp. 9167–9176.

[45] Z. Zhong *et al.*, "Squeeze-and-attention networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 13062–13071.

[46] C. Debes *et al.*, "Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2405–2418, Jun. 2014.

[47] 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.

[48] X. He and Y. Chen, "Optimized input for CNN-based hyperspectral image classification using spatial transformer network," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1884–1888, Dec. 2019.



Zilong Zhong (Member, IEEE) received the Ph.D. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 2019.

He is currently a Post-Doctoral Fellow with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include machine learning, computer vision, probabilistic graph models, and their application to analyzing large-scale high-dimension Earth observation data.

Ying Li (Member, IEEE) received the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2021.

She is currently an Assistant Professor with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China. Her research interests include autonomous driving, environmental perception, computer vision, mobile laser scanning, geometric and semantic modeling, and high definition mapping.

Lingfei Ma received the Ph.D. degree in geomatics engineering from the University of Waterloo, Waterloo, ON, Canada, in 2020.

He is currently an Assistant Professor of urban data science with the Central University of Finance and Economics, Beijing, China. His research interests include autonomous driving, mobile laser scanning, intelligent processing of point clouds, 3-D scene modeling, and machine learning.

Jonathan Li (Senior Member, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa, in 2000.

He is currently a Professor with the Department of Geography and Environmental Management and the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada. He has coauthored more than 300 publications, more than 150 of which were published in refereed journals, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), *ISPRS Journal of Photogrammetry and Remote Sensing* (JPRS), and *Remote Sensing of Environment* (RSE). His research interests include information extraction from mobile light detection and ranging (LiDAR) point clouds and earth observation images.

Dr. Li was the Chair of the ICA Commission on Sensor-Driven Mapping from 2015 to 2019 and the International Society for Photogrammetry and Remote Sensing (ISPRS) WG I/2 on LiDAR, Air- and Spaceborne Optical Sensing from 2016 to 2020; and an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.

Wei-Shi Zheng (Member, IEEE) received the Ph.D. degree in applied mathematics from Sun Yat-sen University, Guangzhou, China, in 2008.

He is currently a Full Professor with Sun Yat-sen University. He has published more than 120 papers, including more than 90 publications in main journals [IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), *International Journal of Computer Vision* (IJCV), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNN/TNNLS), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), and *Pattern Recognition* (PR)] and top conferences [IEEE International Conference on Computer Vision (ICCV), IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), International Joint Conference on Artificial Intelligence (IJCAI), and AAAI Conference on Artificial Intelligence (AAAI)]. His research interests include person/object association and activity understanding in visual surveillance, and the related large-scale machine learning algorithm.

Dr. Zheng was a recipient of Excellent Young Scientists Fund of the National Natural Science Foundation of China, and the Royal Society-Newton Advanced Fellowship of United Kingdom. He joined the Microsoft Research Asia Young Faculty Visiting Programme. He is also an Associate Editor of the *Pattern Recognition* journal and an area chair of a number of top conferences.