ACTN: Adaptive Coupling Transformer Network for Hyperspectral Image Classification

Xiaofei Yang, *Member, IEEE*, Weijia Cao, *Member, IEEE*, Dong Tang, *Member, IEEE*, Yicong Zhou, *Senior Member, IEEE*, and Yao Lu

***Abstract*—Convolutional neural networks (CNNs) and Transformer networks have shown impressive performance in hyperspectral image (HSI) classification. However, these models usually concentrate on examining either local or global representations of HSI data, frequently falling short of capturing multidimensional representations. Furthermore, these methods fail to fully leverage the strengths of CNNs and Transformers. This article presents the adaptive coupling Transformer network (ACTN), a parallel-hybrid network aiming to improve representation learning for HSI classification. ACTN can capture different types of representation and facilitate mutual learning. Specifically, we introduce a parallel-hybrid module called the adaptive coupling module (ACM), which is designed to capture multifaceted representations from the HSI cube. The ACM consists of two branches: a CNN branch that extracts local contextual representations and a Transformer branch that captures global dependency representations. Our proposal is an adaptive response fusion module (ARFM) that interacts with the hybrid module to merge local and global representations at different resolutions in an adaptive way. In addition, we utilize a cosine similarity function to restrict the loss function in mutual learning, guaranteeing the preservation of both local and global representations to the maximum extent. Extensive experiments conducted on three public HSI datasets demonstrate that ACTN outperforms state-of-the-art methods based on Transformers and CNNs.**

***Index Terms*—Convolution neural network (CNN), hyperspectral image classification, mutual learning, Transformer network.**

I. Introduction

With the development of remote sensing sensors, hyperspectral images (HSIs) containing hundreds of bands can easily be obtained. Unlike RGB natural images, HSIs have unique characteristics and can provide rich spatial and spectral information. The goal of hyperspectral image HSI classification is to identify each pixel in the image into one of several predefined land-cover categories. This task is the cornerstone of various remote sensing applications, including land-cover maps [1], marine monitoring [2], and urban semantic segmentation. HSIs collect hundreds of spectral bands, which makes them different from typical natural images. The abundant spectral signatures and the spatial context provide more information for HSI classification, however, preventing machine learning methods from being directly transferred to HSI classification tasks.

The process of traditional HSI classification methods can be divided into two steps: feature extraction and classifier training. For example, the support vector machine (SVM) is a widely used classifier and constructs an optimal hyperplane for classification and maps spectral information into a high-dimensional space to find the optimal classification boundary [3], [4], [5]. Random forests (RFs) is an ensemble learning method that builds multiple decision trees and combines results to achieve classification [6]. However, these methods may fail to fully consider spatial information in HSI. To improve classification accuracy, some methods adapt to aggregate pixels into objects with spatial continuity and perform classification based on object-level features. For instance, principal component analysis (PCA) [7], [8], local binary pattern (LBP) [9], and linear discriminant analysis (LDA) [10], [11], which combine spectral feature extraction and selection with spatial information fusion to better capture spatial–contextual information and improve classification accuracy and stability. However, these methods would have a failure while handling complex HSIs because of their low generalizability and limited representational capacity.

Recently, deep learning models, especially convolution neural networks (CNNs) have been widely used in various computer vision tasks, including image classification, object detection, and semantic segmentation [12], [13], [14], [15]. This is largely attributed to the convolution operation, which can extract hierarchical local features and learn a powerful feature representation. Inspired by these, many CNN-based methods have been proposed to solve HSI classification issues [16], [17], [18], [19], [20]. For example, Yang et al. [18] integrated the 2-D-CNN and 3-D-CNN to help extract abundant features from the limited HSI samples. The concept of band selection in dimensionality reduction for HSI classification was first introduced by Sun et al. [21]. They proposed the iterative sparse subspace clustering (ISSC)

Received 26 August 2024; revised 27 October 2024; accepted 3 January 2025. Date of publication 16 January 2025; date of current version 30 January 2025. This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA26010000, in part by the National Natural Science Foundation of China (NSFC) Fund under Grant 62301174, and in part by the Guangzhou Basic and Applied Basic Research Topics under Grant 2024A04J2081. *(Corresponding author: Weijia Cao.)*

Xiaofei Yang and Dong Tang are with the School of Electronic and Communication Engineering, Guangzhou University, Guangzhou 510182, China (e-mail: xiaofeiyang@gzhu.edu.cn; tangdong@gzhu.edu.cn).

Weijia Cao is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China, also with the Department of Computer and Information Science, University of Macau, Macau, China, and also with Yangtze Three Gorges Technology and Economy Development Company Ltd., Beijing 101100, China (e-mail: caowj@aircas.ac.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: yicongzhou@um.edu.mo).

Yao Lu is with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China (e-mail: luyao2021@hit.edu.cn).

Digital Object Identifier 10.1109/TGRS.2025.3528411

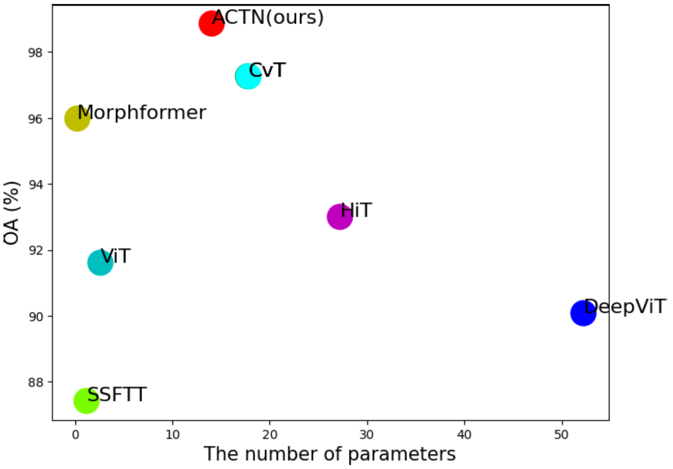


Fig. 1. OA-versus-complexity of state-of-the-art on the Houston2013 dataset.

method to select the best subset of bands from the original HSI. The ISSC method assumes that each band lies in a union of low-dimensional subspaces and can be sparsely represented by other bands within its subspace. Building on this, Sun et al. [22] later introduced the lateral-slice sparse tensor robust principal component analysis (LSSTRPCA) method. This method improves the performance of HSI classification by assuming that a third-order hyperspectral tensor has a low-rank structure, and outliers or gross errors are sparsely scattered in a 2-D space of the tensor. These CNN-based methods integrate the tasks of learning features with training classifiers in an end-to-end way. Such a special structure enables to performance of satisfactory results for HSI classification tasks. Despite the strong capacity of local feature extraction, CNN-based methods may have a failure to capture global representations, for example, long-range dependence. Enlarging the receptive field of the convolution kernel is an intuitive solution, however, it will increase the number of parameters and computation cost.

Most recently, Transformer networks have been introduced into computer vision tasks and achieved satisfactory performance [23], [24], which is attributed to their capability of capturing long-range dependence. For example, Dosovitskiy et al. [25] first employed the Transformer for image classification and proposed a vision image Transformer (ViT) network. In the ViT, the input images would be split into nine patches with positional embeddings and treated as a sequence of tokens. Then, the tokens were fed into a series of Transformer blocks to extract parameterized vectors. The key components of the Transformer are the self-attention mechanism and multilayer perception (MLP), which could collect the spatial transforms and long-range dependencies. Unfortunately, the ViT neglects the 2-D structure of the image, which will decrease the performance. To improve the performance, local features obtained from CNNs are leveraged as input tokens to capture the local spatial information. For example, Graham et al. [26] utilized several convolution layers to extract local features and then fed them into the Transformer blocks. However, these improved Transformers do not precisely embed the local features and global representations into each other.

From the pioneering studies, we can find that a land-cover pixel has a twofold description, such as local contextual information and global representation. However, all the existing methods have the following two challenges.

1) They are challenged in exploring comprehensive representation. For example, CNNs are good at extracting the local feature extraction, while the specialty of Transformers is capturing the global representation. Both CNNs and Transformers are focused on exploiting the single representation of HSIs.
2) They are challenged in mutual learning from each other model. For example, the current methods focus on integrating CNNs into Transformers to improve the performance of Transformers.

To this end, we propose a hybrid network called adaptive coupling Transformer network (ACTN), whose goal is to embed the corresponding local spatial–spectral features and global representations into each other for enhancing feature representation learning. More specifically, ACTN contains two branches: 1) a simple CNN branch for extracting the local features and 2) a Transformer branch for capturing the global representations. In the proposed ACTN, the cross-entropy loss functions are employed to supervise both the CNN and Transformer branches. Additionally, we propose an adaptive response fusion module (ARFM) to bridge between the CNN and the Transformer branches to help with fusing the corresponding local features and global representations. On the one hand, ARFM utilizes down/upsampling strategies to align feature resolutions, the special self-attention mechanisms to select the corresponding representations, LayerNorm and BatchNorm to align feature values, 1 × 1 convolution, and MLP to fuse the two style features. On the other hand, it is inserted into every block to fuse hierarchical local features and global representations. Fully using the proposed ARFM, the ACTN can not only enhance the global perception capability of CNN but also raise the local feature extraction capability of the Transformer. Furthermore, we introduce the cosine similarity function to supervise both the CNN and Transformer branches to improve the performance by using mutual learning. The proposed ACTN is a multibranch network, which consists of three outputs that include CNN, Transformer, and hybrid outputs. Different from other hybrid networks, ACTN could keep the CNN and Transformer separately extracting features and further improve their feature extraction capability. Additionally, the ACTN could make CNN and Transformer help each other to capture the missing information, resulting in breaking the bottleneck of HSI classification. Furthermore, we introduce a cosine similarity function to supervise both the CNN and Transformer branches, enhancing performance through mutual learning. The major contributions of this article are summarized as follows.

1) We propose a hybrid network called ACTN that fully extracts local and global features from HSIs while retaining corresponding local and global representations through mutual learning. In particular, the ACTN could

make the multibranch help each other to capture sufficient features and improve the HSI classification results.

2) We introduce a fusion module called the ARFM for interactively fusing local features and global representations while reducing noise. This function plays a critical role in ensuring that these two branches of the learning process, which might otherwise operate in isolation, are instead encouraged to learn from one another in a mutually beneficial and collaborative manner.
3) We provide a cosine similarity loss function to supervise both local features and global representations in the label subspace, ensuring that the two branches learn from each other. This function plays a critical role in ensuring that these two branches of the learning process, which might otherwise operate in isolation, are instead encouraged to learn from one another in a mutually beneficial and collaborative manner.
4) Extensive experiments on three public HSI datasets demonstrate that the proposed ACTN outperforms state-of-the-art CNN- and Transformer-based methods. For example, as shown in Fig. 1, our ACTN achieves 98.03% accuracy on the Houston2013 dataset, significantly outperforming CvT [27] by 1.6% with 3.79 MB fewer parameters.

The rest of this article is organized as follows. Section II introduces the related work about CNN-based hyperspectral image classification and Transformer-based classification. Section III presents the proposed method and its components. Section IV gives an illustration of the three benchmarks HSI datasets and experimental settings and then presents the results and the analyses of the experiment. Finally, Section V conducts a conclusion.

II. Related Work

A. Convolution Neural Networks for Hyperspectral Image Classification

Since AlexNet [12] has made a breakthrough performance on the ImageNet dataset, CNNs have become a dominant architecture in the computer vision field. There have a series of CNN variants, such as VGG [28], GoogleNet [13], ResNet [14], and DenseNet [15]. The success of CNNs attributed to their strong capability of local feature extraction. Inspired by this, numerous CNN-based methods have been presented for HSI classification [29], [30], [31], [32], [33]. For example, Yang et al. [17] proposed a 2-D-CNN for HSI classification, which was established by stacking three convolution layers. Sharma et al. [34] also designed a simple 2-D-CNN method for recognizing the HSIs. However, 2-D-CNN-based methods only separately process the spatial and spectral information for extracting local features to identify HSIs. To simultaneously analyze the spatial and spectral, Ahmad et al. [35] proposed a 3-D-CNN for HSI classification. Yang et al. [18] proposed a dual-network for hyperspectral image classification by combining the 2-D-CNN and 3-D-CNN. These 3-D-CNN-based methods outperform the 2-D-CNN-based methods, however, they urgently need a lot of training samples. There have been some studies of attention mechanism application in HSI classification. For example, Lorenzo et al. [36] first employed an attention mechanism to select the spectral bands and then fed them into the CNN model to recognize the HSIs. Despite the strong capability of local feature extraction of CNNs, they experience difficulty in capturing the global representations.

There are two solutions for addressing this problem. One solution is to use a convolution layer with larger receptive fields [37], [38], which could extract much more abundant local features. For example, SENet [39] and GENet [40] introduced the global average pooling operation to aggregate the global context. Another solution is to use the global attention mechanism. For instance, Hu et al. [41] proposed an object attention module and designed a relation network for object detection. However, the existing solutions have obvious disadvantages. For the larger receptive fields, they bring many more parameters that require more training samples. For the integration of attention in CNNs, the local features would deteriorate, if convolution layers are not properly coupled with global attention mechanisms.

B. Transformer Networks for Hyperspectral Image Classification

The ViT is a pioneered work, which first introduces Transformer architectures to computer vision tasks and produces a comparable performance to state-of-the-art CNNs. Recently, Transformers have been applied to various computer vision tasks, including image classification, object detection, and semantic segmentation. The existing Transformer networks can capture the long-range dependence, however, they usually neglect the local features. To address such a problem, Wu et al. [27] proposed a new Transformer network, namely CvT, which integrated the convolution operation and self-attention mechanism in a block. In CvT, the convolution operation is used to extract local features, and self-attention is utilized to capture the global representations. However, these Transformers only leverage convolution operations to extract local features, while not taking full advantage of CNNs and Transformers.

Inspired by these, some researchers have applied Transformer networks to HSI classification. For example. Hong et al. [42] proposed a new backbone Transformer network called SpectralFormer to capture local sequence information from HSIs to improve performance. Roy et al. [43] introduced a new morphological Transformer network called morphFormer for HSI classification, which combined the spectral and spatial morphological convolutions with attention mechanisms to improve the interaction between the structural and shape information of the HSI token and the CLS token. Sun et al. [44] combined a hierarchical CNN module with a Transformer structure to extract shallow spatial–spectral features and transform them into tokenized semantic features and finally presented a spectral–spatial feature tokenization Transformer (SSFTT) model for HSI classification. Tu et al. [45] proposed a hierarchical Transformer architecture called local semantic feature aggregation-based

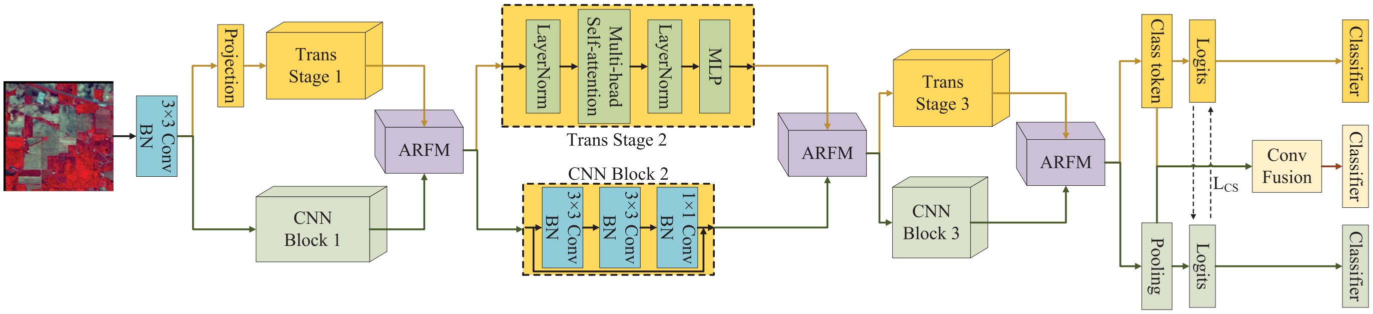


Fig. 2. Overall architecture of the proposed ACTN. It is noted that the ARFM is the proposed ARFM. BN is Batch normalization. L_{CS} denotes the cosine similarity loss. “Trans” is short for the Transformer.

Transformer (LSFAT) for HSI classification. The LSFAT consists of neighborhood aggregation-based attention (NAA) and neighborhood aggregation-based embedding (NAE) modules. Yang et al. [46] integrated the CNN into the Transformer to improve the performance and presented a novel Transformer network called the HSI Transformer (HiT) network for HSI classification.

Different from the existing works, Peng et al. [47] proposed a concurrent neural network, namely Conformer to fuse features interactively. The Conformer utilizes the local features and global representations. Inspired by the Conformer [47], we propose a new concurrent network called ACTN to take advantage of CNNs and Transformers for capturing the local and global features from HSIs. In ACTN, we propose a straightforward couple module called ARFM for fusing the corresponding local features and global representations interactively. The ARFM is a dual-self-attention mechanism, which could generate the corresponding local/global features according to the input features and then fuse the corresponding local and global features. Additionally, we introduce the cosine similarity loss function that considers mutual learning to help the proposed ACTN retain the corresponding local and global representations to the maximum extent. Such ACTN not only precisely fuses the corresponding local features and global representations, but also enhances the representation learning ability of two branches by using the cosine similarity loss. The differences between ACTN and Comformer are reported in Section III-E.

III. Proposed Approach

A. Overview

A land-cover pixel from HSIs has a twofold description: local contextural feature and global representation. Local contextual features are represented by the local spatial contextual of the land-cover pixel, and global representations are the long-range dependence of the land-cover pixel. Both of them are important for recognizing objects. To take full advantage of local features and global representations, we present a hybrid network called ACTN (as shown in Fig. 2) for HSI classification tasks. In ACTN, the proposed ARFM could precisely feed the corresponding global representations obtained from the Transformer branch to local features, and feedback to the corresponding local features from the CNN branch to global representation. As such, the CNN branch has the capability of capturing global representations, while the Transformer branch could exploit the local features.

More specifically, ACTN comprises a head module, hybrid branches (CNN and Transformer branches), three ARFMs, and two classifiers (with class token and pooling operations). The head module is utilized for extracting initial local features, which is a 3 × 3 convolution layer with stride 2 followed by a batch normalization (BN) and a rectified linear unit (ReLU) activation function. The extracted initial local features would be then fed into the hybrid branches. Within hybrid branches, there are three CNN blocks and Transformer stages. Each CNN block has three convolution layers, each of which is followed by a BN and a ReLU activation function. Each Transformer stage is composed of 12 multihead self-attention mechanisms and MLPs. From Fig. 2, each block is followed by an ARFM, which precisely fuses the corresponding local features and global representations interactively.

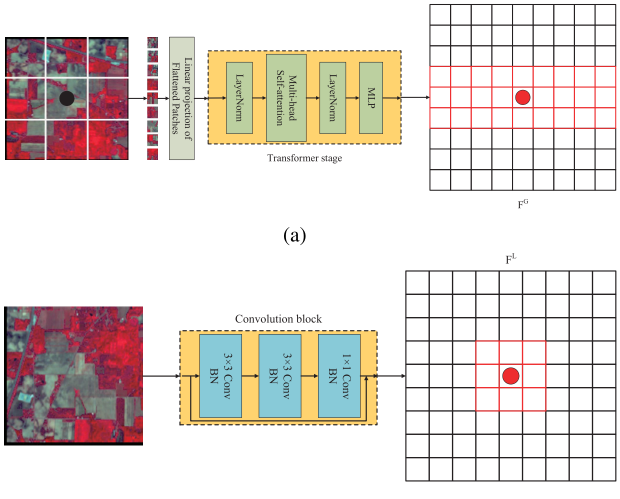
Furthermore, the local features from the CNN branch and the global representations from the Transformer branch are fed into the pooling layer and class token layer, respectively. To ensure the CNN branch and Transformer branch could learn from each other, the cosine similarity function is introduced to make the local and global features similar in the label space. Thus, we use two cross-entropy losses and a cosine similarity loss to supervise the two classifiers.

Finally, a Conv fusion module is used for integrating the local contextual tokens and global representation tokens and generating the final output tokens.

B. Hybrid Branches

Traditional hyperspectral classification methods often depend on single approaches or algorithms, which may not fully utilize diverse spectral information or spatial features effectively. To address this, we propose a hybrid branch method that integrates multiple branches to improve classification accuracy by leveraging the strengths of different branch types within the model architecture. The hybrid branches consist of two different branches: one is the CNN branch for extracting the local contextual features and the other is the Transformer branch for capturing the global representation.

1) CNN Branch: As shown in Fig. 2, the CNN branch adopts three CNN blocks, each of which has three convolution layers followed by BN and ReLU. These three convolution layers are defined as a 1 × 1 channel convolution, a



(a)

(b)

Fig. 3. Twofold descriptions of the HSI, including the global representations F^G obtained by Transformers and the local features F^L obtained by CNNs. (a) Global representation obtained with a Transformer stage. (b) Local features extracted with a Convolution block.

1×1 spatial convolution, a 1×1 channel convolution, and a residual connection between the input feature and output feature. During the training, the stride is set to 1.

2) Transformer Branch: Different from ViT [25], this branch only contains three Transformer stages. According to Fig. 2, each Transformer stage comprises a multihead self-attention (MHSA) module and an MLP block. In the experiments, there are 12 MHSAs to capture the global representations. The settings of MHSA and MLP are followed in ViT.

Many existing approaches may concentrate on single-path models or simple feature combinations. In contrast, the hybrid branches approach differs by providing a more comprehensive integration of diverse data modalities or feature types, such as spectral features, spatial features, and global spatial–spectral dependencies. The model may achieve higher accuracy by effectively utilizing hybrid branches compared to traditional single-branch or single-algorithm methods.

C. Adaptive Response Fusion Module

Local contextual features and global representations captured from HSIs are the two different attributes for depicting the same semantic of the land cover. As shown in (b) of Fig. 3, the local contextual features are extracted using a convolution block consisting of three convolution layers and BN layers. The local contextual features are compact adjacent representations of the image neighborhood, representing the image patch's texture, color, and shape descriptors. At the same time, the land-cover pixel also has another description (i.e., global representation). Different from local contextual features, global representations (as shown in (a) of Fig. 3) are aggregated from compressed patch embedding using a Transformer stage. The global representations are the long-range dependencies and contain contour characterization and shape descriptors of the whole image. Both local contextual features from CNNs and global representations from Transformers are descriptions of the land-cover pixel from HSIs and could help to identify the land-cover pixel. However, CNNs and Transformers only explore a single description and miss the capability of exploring long-range dependencies or the local contextual features, resulting in incomplete characterization of the land-cover pixel. A simple fusion of global representations and local contextual improves the accuracy of the recognition but also brings the side effect of noises (e.g., the introduction of the unrelated background). As a result, how to select the corresponding local contextual features and global representations from HSIs for fusion is a significant problem. Additionally, HSI classification often requires fusing responses from different spectral bands or spatial contexts, which can be complex due to varying noise levels, sensor characteristics, and environmental conditions.

To deal with this problem, we propose an ARFM that contains a dual self-attention block and an up/downsample layer. The architecture of ARFM is shown in Fig. 4. We can observe that the proposed ARFM is a dual structure, which consists of two self-attention modules for selecting the corresponding global/local features and an up/down layer to transform the global/local features. It is noted that the "up" and "down" represent the upsampling and downsampling operations. The processing of ARFM consists of three steps: 1) the transform step: the input features will be transformed using the "up" or "down" operation; 2) the adaptive attention step: generating the corresponding features using the self-attention module; and 3) the residual connection step: outputting the final features by skip connecting the raw input features and the corresponding features.

Finally, ARFM suggests incorporating a module that adaptively adjusts its fusion strategy based on input data characteristics. This module could introduce mechanisms to robustly handle noise and adaptively fuse responses to enhance classification accuracy under varying conditions. It might consider contextual information to optimize fusion decisions, potentially improving the model's robustness and reliability across different scenarios. Unlike existing methods, ARFM could dynamically adjust its fusion parameters or strategies based on real-time input characteristics, potentially offering improved performance by addressing variability and noise explicitly in the fusion module.

For local features F^L (with size $C \times H \times W$), we first use a "up" operation to transform the global representation F^G that has $(K + 1) \times E$ to new features (with the same size of F^L). It is noted that the "up" operation is applied for feature fusion and alignment of the model architecture. With the "up" operation, the output features could align with the features from the CNN branch, resulting in feature fusion and alignment with CNN architecture. Then, these generated new features are regarded as the key map (K), and the raw local features F^L are treated as the query map (Q). We concatenate Q and K, then use two convolution layers (θ and δ) to generate the response weight maps. By using the elements' multiplication on K and the response weight maps, we can obtain the corresponding global representations. Finally, we apply a residual connection between the raw input local features and the corresponding global representations and output the new local features $F^{L'}$. The $F^{L'}$ can be calculated as follows:

$$\begin{aligned} Q^L &= F^L, \quad K^G = W_{Up}(F^G), \quad V^G = W_{Up}(F^G) \\ RW^L &= W_\theta W_\delta \text{concate}(Q^L, K^G) \\ F^{L1} &= RW^L \bigotimes V^G \\ F^{L'} &= W_{1D}(F^{L1} + F^L). \end{aligned} \tag{1}$$

Here, W_{Up}, W_θ, and W_δ are the weights of "up", θ, and δ, respectively. "concate" denotes the concatenate operation. W_{1D} denotes the 1-D convolution layer, which is used to re-weight the fusion features. It is noted that the size of all the convolution layers is set as 1 × 1.

For the global representations F^G, its process is similar to the local features F^L. There are two differences: 1) the local features F^L should first use the "down" operation to generate the features that have the same dimensional with F^G; and 2) the θ' and δ' operations use the Layer Norm operations. As such, the output $F^{G'}$ is formulated as

$$\begin{aligned} Q^G &= F^G, \quad K^L = W_{\text{Down}}(F^L), \ V^L = W_{\text{Down}}(F^L) \\ RW^G &= W_{\theta'} W_{\delta'} \text{concate}(Q^G, K^L) \\ F^{G1} &= RW^G \bigotimes V^L \\ F^{G'} &= W_{1D}(F^{G1} + F^G). \end{aligned} \tag{2}$$

Here, W_{Down}, $W_{\theta'}$, and $W_{\delta'}$ are the weights of "down", θ', and δ', respectively. And W_{1D} denotes the 1-D convolution layer, which is used to re-weight the fusion features. The "down" operation is applied for feature fusion and alignment of the model architecture. Through it, the output features could align with the features from the CNN branch, resulting in feature fusion and alignment with CNN architecture.

By using the proposed ARFM, the generated local features $F^{L'}$ contain the local contextual features and the global representation, and the generated global representation $F^{G'}$ also has the local features. Finally, both $F^{L'}$ and $F^{G'}$ will be sent to the next module.

The ARFM, is an innovative module, to interactively fuses corresponding local features and global representations, reducing noise features and representations. It enhances and integrates any potentially missing information within the context of a multibranch network. It also ensures a seamless flow of information by efficiently filling in any potential gaps in data transmission between distinct branches, paving the way for more accurate and robust HSI classification.

D. Cosine Similarity Loss

The objective function of CNNs and Transformers is defined as the cross-entropy error between the predicted and ground-truth labels:

$$\begin{aligned} \text{Loss}_{C1} &= -\sum_{i=1}^{N} y_{o,i} \log(p_{o,i}) \\ \text{Loss}_{C2} &= -\sum_{i=1}^{N} y_{o,i} \log(p_{o,i}) \\ \text{Loss}_{F} &= -\sum_{i=1}^{N} y_{o,i} \log(p_{o,i}) \end{aligned} \tag{3}$$

where N is the number of classes and p denotes the predicted probability observation o is of class i. Loss_{C1}, Loss_{C2}, and Loss_F are the loss function for the CNN network, Transformer network, and the final output, respectively. These three supervised losses are used to train the networks to predict the correct labels.

We believe that the class tokens and the local logits should be similar in the label subspace if both the CNN and Transformer predict the correct labels. To quantify the similarity of the two networks' predictions $p1$ and $p2$, we use the Cosine Similarity Loss.

The distance from $p1$ to $p2$ is calculated as

$$\text{Loss}_{CS} = \frac{p1 \cdot p2}{\|p1\| \|p2\|} = \frac{\sum_{i=1}^{N} p1_i \times p2_i}{\sqrt{\sum_{i=1}^{N}(p1_i)^2 \times \sum_{i=1}^{N}(p2_i)^2}}. \tag{4}$$

Here, $p1_i$ and $p2_i$ are the vectors of $p1$ and $p2$.

The overall loss function for the ACTN is

$$\text{Loss} = \alpha \text{Loss}_{C1} + \beta \text{Loss}_{C2} + \lambda_1 \text{Loss}_F + \lambda_2 \text{Loss}_{CS} \tag{5}$$

where $\alpha = \beta = 1.0$, $\lambda_1 = 0.5$, and $\lambda_2 = 0.005$.

In this way, the performance both of CNN and Transformer networks could be improved and correctly predict the ground-truth label.

The function closely oversees two pivotal aspects of the learning process: local features and global representations. These two branches do not operate independently but interact, learning from each other in a mutually beneficial, collaborative way. This approach significantly enhances the learning process, enabling a richer, more comprehensive understanding of the subject matter as it pertains to the label subspace.

E. Discussions

1) Difference to the Conformer: Inspired by the Conformer [47], we also adopt the residual connection into our CNN branch. More specifically, the Conformer is designed for natural computer vision tasks, while ACTN is built for hyperspectral image classification tasks. Additionally, we remove many residual blocks, in which only three residual convolution blocks for extracting local features. We also use a small convolution layer (the size is 3 × 3) as the head block rather than a larger convolution layer (with the size 7 × 7). Furthermore, we propose an ARFM to fuse the corresponding local features and global representations, instead of a simple convolution layer and a Layer Norm operation. Last but not least, we introduce the Cosine Similarity Loss function to compute the distance between local and global features during the label subspace, which could improve the performance of ACTN.

2) Difference to RelationNet: There are three main differences between RelationNet [41] and the proposed Transformer. The first one is the network architecture. RelationNet is built on the classic self-attention module mechanisms. The proposed ACTN is built on both CNNs and Transformers. Additionally, the proposed ACTN utilizes the ARFM to adaptive aggregate

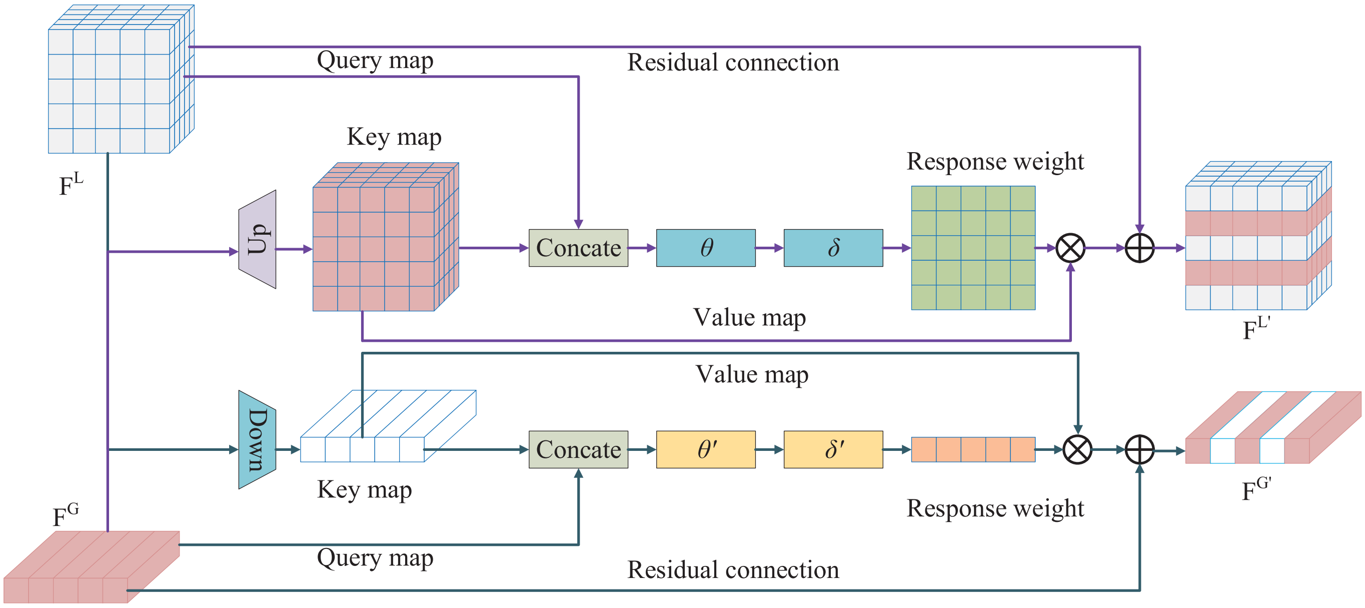


Fig. 4. Architecture of ARFM. F^G and F^L are the global representations and the local features. $F^{G'}$ and $F^{L'}$ denote the fused global representations and local features.

the local features and the global representations. The second difference is the application of the methods. The RelationNet is proposed for object detection, while the proposed ACTN is presented for HSI classification. The third one is the attention mechanism. In ACTN, the ARFM is proposed to adapt and fuse the local and global features. Moreover, the ARFM takes two different mode features (such as local features and global representations) as input features.

3) Difference to Focal Transformer: There are three differences between the Focal Transformer [48] (Focal-T for short) and ACTN. First, Focal-T uses a shift-window module to extract the local features. Different from Focal-T, the proposed ACTN adopts two branches, that is, CNNs and Transformers, in which the CNN branch is used to extract local features, and the Transformer branch is employed to capture the global representations. Second, the style for aggregating information. In Focal-T, the focal attention module is proposed to aggregate the local and global features. While in ACTN, the local features and global representations are fed into the ABFM for adaptive fusing the corresponding features/representations and outputs the new features/representations. Finally, the proposed ACTN achieves better performance than the one of Focal-T.

IV. Experiments

A. Datasets and Setting

1) Datasets: Three HSI datasets are utilized in our experiments, including the Indian Pines scene, Houston2013, and PaviaU datasets.

1) *Indian Pines Scene Dataset:* This HSI dataset was collected by the airborne visible imaging spectrometer (AVIRIS) sensor in 1992 and recorded the information from North-Western Indiana USA. Its spatial resolution is 145 × 145, and 220 bands in the spectral dimension. In the experiments, the number of spectral bands is 200 after moving 20 noise bands. The number of categories is 16, including Alfalfa, Corn, and Woods. It is noted that we use a 10% sample for training and the remains for testing.
2) *Houston2013 Dataset:* This dataset records the University of Houston and its surroundings, Texas, USA, and is collected by the ITRES CASI-1500 sensor. It contains 349 × 1905 in the spatial dimension and 144 spectral bands. In the experiments, it is a cloud-free image provided by the geo-science and remote sensing society (GRSS). There are a total of 15 labeled classes, including Highway, Road, Trees, and so on. We select 10% samples for training and the remaining for testing.
3) *PaviaU Dataset:* This HSI dataset was acquired by the ROSIS sensor at the University of Pavia, Italy. This image contains 610 × 340 pixels and 103 spectral bands. There are nine classes of the PaviaU dataset, including Asphalt, Gravel, trees, and so on. Finally, we select 10% samples for training and the remaining for testing.
4) *WHU-Hi-LongKou (WHL) Dataset:* The data was collected in Longkou Town, Hubei province, China on July 17, 2018, using an 8-mm focal length Headwall Nano-Hyperspec imaging sensor mounted on a DJI Matrice 600 Pro (DJI M600 Pro) UAV platform. The UAV flew at an altitude of 500 m and the resulting imagery had a resolution of 550 × 400 pixels with 270 bands ranging from 400 to 1000 nm. The dataset contains 204 542 labeled samples across nine land-cover classes. In our experiments, we used only 1% of the samples for training, reserving the remaining 99% for testing.

2) Evaluation Metrics: The classification results are evaluated with two widely used metrics, that is, overall accuracy (OA), average accuracy (AA), and Kappa coefficient (κ).

TABLE I

Comparison With the State-of-the-Art CNNs and Transformers on the Indian Pines Scene Dataset (10% Training Samples)

Class	2D-CNN	3D-CNN	Hybridsn	SyCNN	ViT	Deep ViT	CvT	HiT	SSFTT	MorphFormer	SS_TMNet	DCTN	ACTN
Alfalfa	92.82 ± 4.80	69.95 ± 17.92	18.85 ± 29.79	90.24 ± 3.39	80.76 ± 5.74	81.65 ± 10.84	48.74 ± 15.97	91.14 ± 3.48	89.65 ± 6.91	82.13 ± 22.40	87.48 ±8.15	72.29 ±11.04	95.45 ± 0.93
Corn-notill	93.81 ± 1.97	88.61 ± 1.17	84.84 ± 11.22	92.94 ± 0.46	94.29 ± 0.77	94.14 ± 1.06	91.40 ± 2.24	94.49 ± 0.39	94.11 ± 1.08	93.38 ± 2.14	88.56 ±2.34	95.70 ±1.57	94.92 ± 0.56
Corn-mintill	92.19 ± 1.77	86.74 ± 1.90	75.93 ± 18.40	91.90 ± 0.53	93.63 ± 0.58	92.86 ± 1.30	85.23 ± 4.89	94.43 ± 0.59	90.13 ± 2.66	91.28 ± 3.66	76.50 ±3.23	88.93 ±1.40	93.09 ± 0.72
Corn	97.94 ± 1.50	93.92 ± 2.44	80.93 ± 17.01	97.14 ± 1.20	99.62 ± 0.28	99.22 ± 1.22	92.32 ± 4.40	99.73 ± 0.13	94.90 ± 3.46	95.26 ± 4.04	82.19 ±3.92	92.77 ±2.77	99.66 ± 0.24
Grass-pasture	93.09 ± 3.32	93.44 ± 0.70	73.56 ± 16.43	93.75 ± 0.50	92.00 ± 1.11	89.45 ± 2.91	88.40 ± 2.40	92.89 ± 0.24	93.08 ± 2.52	94.72 ± 1.55	81.71 ±3.49	92.04 ±1.28	94.74 ± 0.62
Grass-trees	95.65 ± 2.97	94.82 ± 0.67	75.90 ± 15.75	92.21 ± 0.73	90.79 ± 1.41	90.74 ± 2.85	93.92 ± 1.02	93.50 ± 0.63	95.98 ± 1.29	95.47 ± 1.38	97.76 ±0.92	98.64 ±0.68	96.24 ± 0.22
Grass-pasture-mowed	7.94 ± 18.29	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	8.08 ± 22.04	0.00 ± 0.00	54.69 ± 35.84	71.56 ± 18.01	72.09 ±16.30	73.76 ±9.76	29.73 ± 30.38
Hay-windrowed	99.69 ± 0.55	98.87 ± 1.10	87.72 ± 13.09	98.05 ± 0.40	99.77 ± 0.22	99.79 ± 0.17	99.31 ± 0.81	99.61 ± 0.15	98.77 ± 1.35	99.81 ± 0.39	94.39 ±0.47	97.75 ±0.95	99.98 ± 0.04
Oats	73.30 ± 29.09	0.00 ± 0.00	2.45 ± 5.09	1.82 ± 5.45	4.29 ± 10.09	25.28 ± 27.21	8.49 ± 15.63	0.00 ± 0.00	54.34 ± 32.04	21.28 ± 30.14	68.66 ±17.26	76.70 ±15.73	0.00 ± 0.00
Soybean-notill	87.78 ± 1.60	83.25 ± 1.22	78.05 ± 9.87	87.01 ± 0.44	89.90 ± 0.36	88.77 ± 1.22	84.38 ± 1.41	89.48 ± 0.27	87.11 ± 1.77	88.80 ± 3.58	87.19 ±1.98	93.57 ±1.21	88.77 ± 0.58
Soybean-mintill	96.26 ± 1.24	94.38 ± 0.51	91.41 ± 4.16	94.41 ± 0.24	96.55 ± 0.12	96.65 ± 0.55	94.64 ± 0.63	96.65 ± 0.06	96.78 ± 0.82	96.27 ± 0.59	90.70 ±1.63	95.46 ±0.53	97.01 ± 0.28
Soybean-clean	91.80 ± 2.21	89.11 ± 1.71	78.53 ± 12.76	92.55 ± 0.64	92.96 ± 1.34	93.57 ± 1.30	86.16 ± 5.24	93.85 ± 0.42	89.52 ± 3.35	87.66 ± 5.08	81.85 ±3.97	94.68 ±1.39	93.65 ± 0.40
Wheat	98.12 ± 1.32	86.71 ± 7.81	54.68 ± 33.43	93.85 ± 1.03	96.73 ± 1.43	97.08 ± 1.64	89.64 ± 4.40	97.11 ± 1.27	95.00 ± 3.62	94.35 ± 4.88	97.18 ±3.02	99.89 ±0.18	98.03 ± 0.86
Woods	98.28 ± 2.42	97.81 ± 0.58	94.10 ± 5.22	98.02 ± 0.32	98.27 ± 0.38	97.75 ± 0.75	98.39 ± 0.52	98.47 ± 0.17	98.67 ± 0.66	98.87 ± 0.31	96.21 ±0.89	98.49 ±0.40	98.93 ± 0.25
Buildings-Grass-Trees-Drives	97.82 ± 1.46	93.48 ± 2.25	73.19 ± 17.09	97.99 ± 0.41	98.00 ± 0.60	98.25 ± 1.12	91.41 ± 3.60	98.70 ± 0.47	96.08 ± 2.76	96.06 ± 1.48	63.92 ±3.78	77.76 ±1.60	98.50 ± 0.46
Stone-Steel-Towers	52.74 ± 21.39	46.87 ± 17.62	41.04 ± 32.91	68.64 ± 2.88	53.30 ± 17.49	63.78 ± 21.77	67.35 ± 23.50	67.64 ± 2.82	39.20 ± 32.30	25.11 ± 33.47	87.73 ±3.15	96.80 ±1.98	67.32 ± 4.50
OA (%)	94.48 ± 1.41	91.48 ± 0.52	83.10 ± 10.19	93.42 ± 0.09	94.50 ± 0.30	94.28 ± 0.43	91.41 ± 0.93	95.03 ± 0.15	94.09 ± 0.97	94.03 ± 0.90	84.67 ±1.25	92.85 ±0.41	**95.40 ± 0.11**
AA(%)	83.81 ± 3.38	73.92 ± 2.12	62.87 ± 12.02	80.85 ± 0.16	78.40 ± 1.18	79.90 ± 2.11	77.31 ± 2.41	80.47 ± 0.52	84.54 ± 3.91	82.75 ± 2.85	85.56 ± 4.91	86.55 ± 3.55	**88.47 ± 1.39**
κ (%)	93.69 ± 1.61	90.25 ± 0.59	80.70 ± 11.52	92.50 ± 0.10	93.71 ± 0.35	93.46 ± 0.49	90.20 ± 1.06	94.32 ± 0.17	93.25 ± 1.10	93.18 ± 1.03	82.66 ±1.41	91.87 ±0.47	**94.75 ± 0.12**

TABLE II

Comparison With the State-of-the-Art CNNs and Transformers on the Houston2013 Dataset (10% Training Samples)

Class	2D-CNN	3D-CNN	HybridSN	SYCNN	ViT	Deep ViT	CvT	HiT	SSFTT	Morphformer	SS_TMNet	DCTN	ACTN
Healthy Grass	95.21 ± 1.66	91.06 ± 3.44	90.76 ± 1.88	93.88 ± 0.90	87.43 ± 3.25	91.33 ± 2.17	87.30 ± 10.73	94.49 ± 0.52	96.42 ± 1.16	85.97 ± 8.75	97.60 ±0.64	98.86 ±0.50	97.20 ± 0.44
Stressed grass	95.51 ± 1.72	88.85 ± 6.46	86.26 ± 8.27	94.03 ± 1.64	80.16 ± 5.76	83.59 ± 2.71	90.21 ± 4.40	91.22 ± 2.10	97.16 ± 1.42	88.97 ± 5.58	98.44 ±0.56	99.33 ±0.22	98.50 ± 0.57
Synthetic GrassTrees	99.24 ± 0.45	97.36 ± 3.06	95.85 ± 3.67	96.81 ± 1.89	97.95 ± 0.85	98.64 ± 0.56	97.70 ± 2.80	98.97 ± 0.29	99.30 ± 0.58	91.37 ± 13.84	99.50 ±0.23	99.74 ±0.29	99.13 ± 0.20
Trees	94.72 ± 2.18	88.28 ± 4.52	79.71 ± 7.46	92.54 ± 1.43	81.31 ± 6.65	83.87 ± 4.62	89.45 ± 3.27	89.26 ± 1.20	97.68 ± 0.90	90.96 ± 4.95	97.26 ±0.96	99.21 ±0.41	96.19 ± 0.45
Soil	99.81 ± 0.17	95.72 ± 3.71	96.41 ± 3.83	99.06 ± 0.49	97.80 ± 1.25	98.62 ± 1.51	99.17 ± 0.67	99.53 ± 0.24	99.43 ± 0.63	97.96 ± 2.17	98.19 ±0.33	98.65 ±0.12	99.89 ± 0.15
Water	94.05 ± 0.97	86.16 ± 1.92	91.69 ± 3.12	90.80 ± 2.16	86.50 ± 1.79	87.48 ± 2.36	93.58 ± 2.23	86.87 ± 0.82	93.61 ± 3.07	90.85 ± 4.25	93.67 ±2.33	98.75 ±1.07	93.23 ± 1.39
Residential	96.82 ± 1.02	83.00 ± 6.98	78.74 ± 17.12	92.90 ± 1.13	87.38 ± 1.99	87.28 ± 2.68	92.51 ± 3.82	91.87 ± 0.83	97.52 ± 0.80	85.78 ± 24.73	94.54 ±1.03	98.20 ±0.45	97.97 ± 0.62
Commercial	97.62 ± 1.51	89.55 ± 1.61	93.01 ± 2.65	92.73 ± 1.31	90.55 ± 3.34	94.05 ± 1.95	94.16 ± 3.55	96.41 ± 0.64	97.88 ± 1.24	90.76 ± 9.56	95.74 ±1.35	98.22 ±0.69	99.57 ± 0.21
Road	95.42 ± 1.51	85.03 ± 3.38	73.95 ± 14.36	91.56 ± 1.34	86.62 ± 1.57	87.98 ± 1.78	87.78 ± 2.93	89.78 ± 0.99	96.70 ± 1.56	87.54 ± 7.99	94.29 ±1.32	97.66 ±0.56	97.00 ± 0.71
Highway	99.09 ± 1.57	91.67 ± 6.54	91.92 ± 7.92	98.51 ± 0.56	96.43 ± 3.22	95.01 ± 3.44	97.50 ± 1.56	97.93 ± 0.51	99.78 ± 0.38	93.89 ± 9.76	96.91 ±0.81	98.89 ±0.49	100.00 ± 0.01
Railway	99.58 ± 0.46	81.57 ± 5.87	85.29 ± 7.92	94.61 ± 1.93	93.86 ± 2.51	93.32 ± 4.12	94.73 ± 3.22	99.37 ± 0.45	99.76 ± 0.44	91.69 ± 16.26	94.94 ±0.72	98.51 ±0.33	99.97 ± 0.07
Parking Lot 1	97.88 ± 1.83	94.30 ± 1.81	95.79 ± 2.71	96.66 ± 0.55	91.42 ± 4.91	93.49 ± 3.96	95.83 ± 3.47	98.68 ± 0.21	98.70 ± 1.24	88.63 ± 13.89	96.50 ±1.00	98.96 ±0.22	99.59 ± 0.33
Parking Lot 2	97.55 ± 2.44	81.89 ± 6.24	86.51 ± 8.08	90.13 ± 1.46	91.84 ± 1.55	89.01 ± 4.03	95.80 ± 2.87	95.93 ± 1.53	98.54 ± 1.23	92.25 ± 6.66	93.42 ±1.61	97.68 ±1.01	99.40 ± 0.58
Tennise Court	99.98 ± 0.07	98.60 ± 0.97	92.98 ± 6.16	99.70 ± 0.33	99.20 ± 0.85	98.43 ± 2.66	97.96 ± 3.28	99.99 ± 0.04	99.67 ± 0.47	99.04 ± 1.46	99.88 ±0.19	100.00 ±0.00	100.00 ± 0.00
Running Track	97.87 ± 1.71	94.77 ± 5.41	91.48 ± 5.60	95.62 ± 2.21	96.93 ± 1.89	98.14 ± 0.72	96.65 ± 3.12	98.39 ± 0.33	98.95 ± 0.71	93.32 ± 7.49	98.98 ±0.58	99.24 ±0.88	98.97 ± 0.36
OA(%)	97.32 ± 0.48	89.54 ± 3.21	88.19 ± 4.75	94.68 ± 0.55	90.27 ± 1.69	91.58 ± 1.50	93.52 ± 1.88	95.20 ± 0.33	98.15 ± 0.53	90.98 ± 7.71	96.22 ±0.35	98.31 ±0.16	**98.58 ± 0.16**
AA(%)	97.03 ± 0.37	89.74 ± 2.87	88.97 ± 4.09	94.56 ± 0.65	90.45 ± 1.42	91.38 ± 1.43	93.74 ± 1.95	94.80 ± 0.32	97.81 ± 0.59	90.99 ± 7.46	95.15± 0.45	97.32± 0.36	**98.16 ± 0.22**
κ(%)	97.10 ± 0.51	88.70 ± 3.47	87.24 ± 5.13	94.24 ± 0.60	89.48 ± 1.83	90.89 ± 1.62	92.99 ± 2.04	94.81 ± 0.35	98.00 ± 0.58	90.24 ± 8.36	95.92 ±0.38	98.17 ±0.17	**98.46 ± 0.17**

TABLE III

Comparison With the State-of-the-Art CNNs and Transformers on the PaviaU Dataset (10% Training Samples)

Class	2D-CNN	3D-CNN	HybridSN	SYCNN	ViT	Deep ViT	CvT	HiT	SSFTT	Morphformer	SS_TMNet	DCTN	ACTN
Asphalt	98.68 ± 0.33	94.09 ± 2.75	95.67 ± 1.91	97.42 ± 0.35	98.31 ± 0.33	96.96 ± 1.53	90.49 ± 5.36	98.32 ± 0.25	98.07 ± 2.25	94.76 ± 3.42	96.11 ±0.24	98.57 ±0.09	99.37 ± 0.14
Meadows	99.85 ± 0.02	99.56 ± 0.53	99.45 ± 0.21	99.78 ± 0.09	99.63 ± 0.11	99.65 ± 0.12	97.86 ± 2.34	99.73 ± 0.06	99.85 ± 0.03	99.67 ± 0.19	92.67 ±0.08	97.66 ±0.07	99.85 ± 0.01
Gravel	98.80 ± 0.33	97.54 ± 0.86	96.64 ± 2.57	97.81 ± 0.37	97.34 ± 0.78	98.05 ± 0.38	87.19 ± 10.57	98.44 ± 0.50	99.09 ± 0.46	86.82 ± 23.92	92.35 ±0.66	90.37 ±1.03	99.36 ± 0.20
Trees	94.79 ± 0.58	88.70 ± 3.58	86.19 ± 5.96	90.63 ± 1.93	93.09 ± 0.52	90.41 ± 3.92	83.68 ± 5.51	93.82 ± 0.50	94.53 ± 3.53	87.85 ± 4.12	96.46 ±0.49	98.69 ±0.33	96.63 ± 0.32
Painted metal sheets	93.80 ± 0.67	93.33 ± 0.52	92.85 ± 0.31	93.03 ± 0.99	94.67 ± 1.15	96.20 ± 2.05	94.31 ± 1.43	94.56 ± 0.53	93.94 ± 1.16	94.58 ± 2.46	99.66 ±0.16	99.99 ±0.02	95.66 ± 1.25
Bare Soil	99.95 ± 0.05	98.44 ± 1.83	99.66 ± 0.22	99.78 ± 0.08	99.83 ± 0.20	99.61 ± 0.31	92.92 ± 9.40	99.66 ± 0.12	99.92 ± 0.11	99.84 ± 0.18	99.91 ±0.09	99.04 ±0.36	99.94 ± 0.04
Bitumen	98.88 ± 0.56	97.51 ± 3.89	95.25 ± 3.12	98.91 ± 0.67	98.85 ± 0.23	99.10 ± 0.37	84.11 ± 15.21	99.23 ± 0.10	98.91 ± 0.49	90.65 ± 20.10	99.05 ±0.57	98.86 ±0.30	99.14 ± 0.18
Self-Blocking Bricks	99.02 ± 0.24	96.43 ± 1.37	92.73 ± 5.99	97.04 ± 1.40	98.07 ± 0.42	97.58 ± 0.74	91.45 ± 5.16	98.79 ± 0.24	99.12 ± 0.32	92.82 ± 5.67	98.31 ±0.38	95.04 ±0.40	99.13 ± 0.23
Shadows	84.21 ± 1.41	78.94 ± 5.91	75.97 ± 4.87	79.35 ± 2.80	83.99 ± 2.13	88.35 ± 4.33	80.90 ± 6.91	84.66 ± 1.50	85.15 ± 4.00	86.53 ± 2.07	98.02 ±0.76	99.95 ±0.06	88.94 ± 2.52
OA(%)	98.66 ± 0.10	96.76 ± 0.94	96.28 ± 1.62	97.75 ± 0.33	98.24 ± 0.14	97.93 ± 0.66	93.30 ± 4.05	98.46 ± 0.10	98.58 ± 0.65	96.33 ± 2.17	91.74 ±0.12	96.57 ±0.14	**99.09 ± 0.08**
AA (%)	94.38 ± 0.47	84.89 ± 2.68	83.71 ± 3.80	89.78 ± 1.04	84.53 ± 2.50	91.99 ± 0.96	96.39 ± 0.36	91.51 ± 0.80	96.51 ± 0.35	96.22 ± 0.50	95.31 ± 0.57	96.32± 0.46	**96.81 ± 0.36**
κ(%)	98.22 ± 0.14	95.70 ± 1.25	95.08 ± 2.14	97.03 ± 0.44	97.67 ± 0.18	97.25 ± 0.87	91.04 ± 5.49	97.96 ± 0.13	98.12 ± 0.86	95.13 ± 2.90	89.44 ±0.16	95.49 ±0.19	**98.80 ± 0.10**

TABLE IV

Comparison With the State-of-the-Art CNNs and Transformers on the WHL Dataset (1% Training Samples)

Class	2D-CNN	3D-CNN	HybridSN	SYCNN	ViT	Deep ViT	CvT	HiT	SSFTT	Morphformer	SS_TMNet	DCTN	ACTN
Corn	99.87 ±0.03	99.34 ±0.40	99.34 ±0.58	99.55 ±0.10	99.33 ±0.13	99.66 ±0.08	99.83 ±0.07	99.77 ±0.04	99.88 ±0.04	99.89 ± 0.06	99.86 ± 0.11	99.98 ± 0.01	99.98 ± 0.00
Cotton	99.72 ±0.09	96.30 ±1.24	97.55 ±3.18	99.54 ±0.19	83.45 ±0.89	96.41 ±1.60	99.39 ±0.23	97.74 ±0.69	99.69 ±0.11	99.78 ± 0.13	98.55 ± 1.50	99.94 ± 0.05	99.96 ± 0.03
Sesame	94.97 ±1.27	55.88 ±29.79	81.52 ±14.54	95.19 ±1.71	51.31 ±21.70	88.79 ±4.29	97.91 ±0.71	91.44 ±1.50	98.93 ±0.54	99.44 ± 0.42	95.42 ± 4.00	99.95 ± 0.12	99.98 ± 0.03
Broad-leaf soybean	99.12 ±0.08	96.24 ±0.89	98.22 ±0.64	98.38 ±0.18	96.51 ±0.67	98.52 ±0.22	99.49 ±0.09	98.71 ±0.21	99.63 ±0.05	99.65 ± 0.16	99.79 ± 0.10	99.87 ± 0.04	99.87 ± 0.01
Narrow-leaf soybean”	95.42 ±0.71	87.80 ±2.44	81.06 ±10.19	89.92 ±1.29	42.38 ±7.08	89.80 ±2.31	97.10 ±1.00	95.35 ±1.41	98.30 ±0.48	97.60 ± 1.50	91.07 ± 4.08	99.38 ± 0.53	99.42 ± 0.16
Rice	98.57 ±0.19	98.11 ±0.66	97.28 ±0.91	97.49 ±0.59	98.54 ±0.40	98.87 ±0.16	98.80 ±0.11	99.26 ±0.05	98.96 ±0.44	99.05 ± 0.23	99.23 ± 0.25	99.03 ± 0.48	99.39 ± 0.03
Water	99.74 ±0.04	99.51 ±0.22	97.11 ±0.53	99.27 ±0.19	99.28 ±0.11	99.39 ±0.18	99.70 ±0.12	99.45 ±0.04	99.56 ±0.12	99.49 ± 0.15	99.98 ± 0.01	99.74 ± 0.08	99.67 ± 0.05
Roads and houses	85.56 ±0.78	82.84 ±2.22	78.60 ±7.09	83.26 ±0.99	80.96 ±2.04	83.05 ±1.42	84.75 ±0.96	87.96 ±0.74	89.32 ±2.82	91.03 ± 1.82	88.66 ± 3.04	87.30 ± 1.69	91.15 ± 0.54
Mixed weed	78.58 ±2.04	79.05 ±2.99	37.10 ±12.86	74.30 ±2.91	77.37 ±2.01	74.30 ±2.89	79.17 ±2.84	80.65 ± 1.01	84.48 ±3.09	83.76 ± 1.79	78.67 ± 8.08	82.71 ± 2.24	85.32 ± 0.05
OA(%)	98.35 ± 0.09	96.60 ± 0.63	95.77 ± 0.80	97.59 ± 0.13	95.28 ± 0.48	97.55 ± 0.17	98.50 ± 0.12	98.18 ± 0.11	98.86 ± 0.20	98.74 ± 0.16	98.61 ± 0.26	98.93 ± 0.13	**99.13 ± 0.05**
AA (%)	93.24 ± 0.46	84.58 ± 3.95	82.29 ± 3.59	90.84 ± 0.71	77.94 ± 3.08	90.01 ± 1.04	94.34 ± 0.58	92.43 ± 0.53	95.72 ± 0.72	95.03 ± 0.67	94.58 ± 1.04	95.18 ± 0.55	**96.13 ± 0.53**
κ(%)	97.82 ± 0.13	95.49 ± 0.85	94.38 ± 1.07	96.82 ± 0.17	93.74 ± 0.65	96.76 ± 0.23	98.03 ± 0.16	97.60 ± 0.15	98.50 ± 0.26	98.34 ± 0.21	98.17 ± 0.35	98.59 ± 0.18	**98.86 ± 0.07**

3) Comparison Methods: We choose seven state-of-the-art methods to compare with the ACTN, including CNNs (e.g., 2-D-CNN [17] and 3-D-CNN [18]) and Transformers (such as ViT [25], DeepViT [49], CvT [27], HiT [46], MorphFormer [43], SS_TMNet [50], DCTN [51], and SSFTT [44]).

4) Setting: During the training, we randomly crop 100 patches with the size of 15 × 15 as the inputs. We set the iterations as 100 and implemented all the comparison methods and ACTN with the PyTorch framework. We update the ACTN using an Adam optimizer. Furthermore, we set the learning rate as $1e-3$ and the batch size is 100.

B. Results and Analysis

We conduct experiments on three widely used HSI datasets in terms of two metrics. The results are reported on Tables I–IV. In tables, the best results are marked in bold.

From Tables I–IV, we can observe that the proposed ACTN achieves the best classification results on the three HSI datasets. It demonstrates the superiority of the proposed

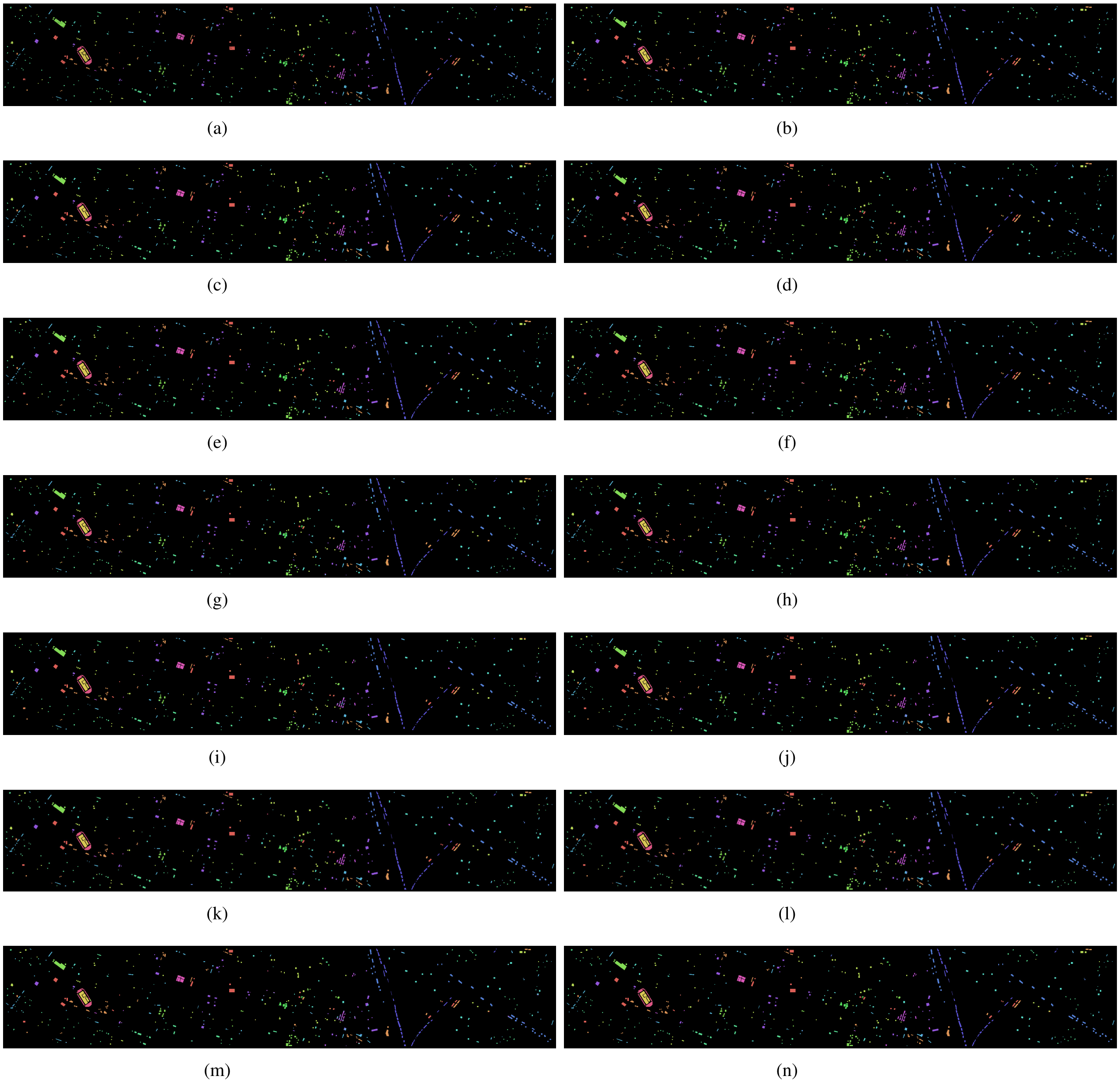

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n)

Fig. 5. Classification maps obtained by different methods on the Houstong2013 scene dataset (with 10% training samples). (a) Ground truth. (b) 2-D-CNN. (c) 3-D-CNN. (d) SyCNN. (e) HybridSN. (f) ViT. (g) Deep ViT. (h) CvT. (i) HiT. (j) SSFTT. (k) DCTN. (l) SS_TMNet. (m) MorphFormer. (n) ACTN.

ACTN. This is mainly because ACTN utilizes ARFM to fuse the precise corresponding local and global features and KL-divergence (KL) to supervise the two classifiers. In particular, ACTN achieves satisfactory classification results on the imbalanced Indian Pines dataset. More importantly, ACTN performs best in almost every category.

One interesting finding is that the Transformer-based methods (such as CvT and HiT) containing the local feature extraction operation achieve better performance than the classical Transformer-based methods (e.g., ViT and DeepViT). This result may be explained by the fact that the classical Transformers could capture the global representations from the patch embeddings but easily neglect the local features. Attributing to extracting the local features from the HSIs, the Transformer-based methods (such as CvT, SSFTT, DCTN, SS_TMNet MorphFormer, and HiT) outperform the classical Transformer-based methods. For example, CvT surpasses ViT by 3.25% on the Houston2013 dataset (93.52% versus 90.27%). These findings may help us to understand the necessity of combining local features and global representations. A simple fusion strategy will help to improve the classification accuracy, however, it also brings some noise and redundant information. Compared to the existing Transformer-based methods, the proposed ACTN could fuse the corresponding local/global features using the ARFM. As a result, it performs better classification performance than the existing Transformer-based methods.

On the other hand, CNNs, which are a type of deep learning model, are renowned for their adeptness at extracting localized contextual features from data. They are especially

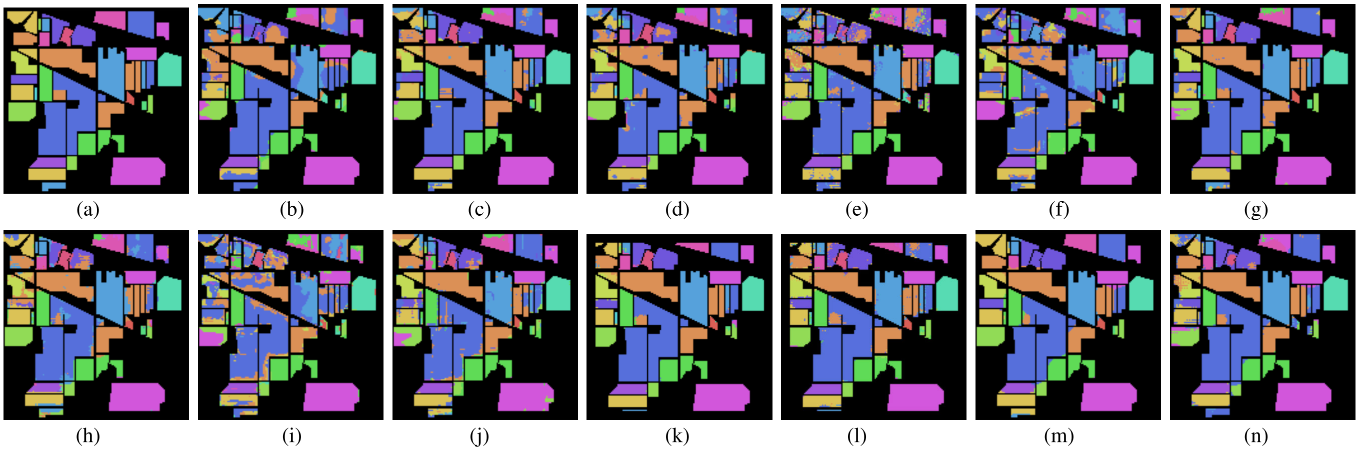


Fig. 6. Classification maps obtained by different methods on the Indian Pines Scene dataset (with 10% training samples). (a) Ground truth. (b) 2-D-CNN. (c) 3-D-CNN. (d) SyCNN. (e) HybridSN. (f) ViT. (g) Deep ViT.(h) CvT. (i) HiT. (j) SSFTT. (k) DCTN. (l) SS TMNet. (m) MorphFormer. (n) ACTN.

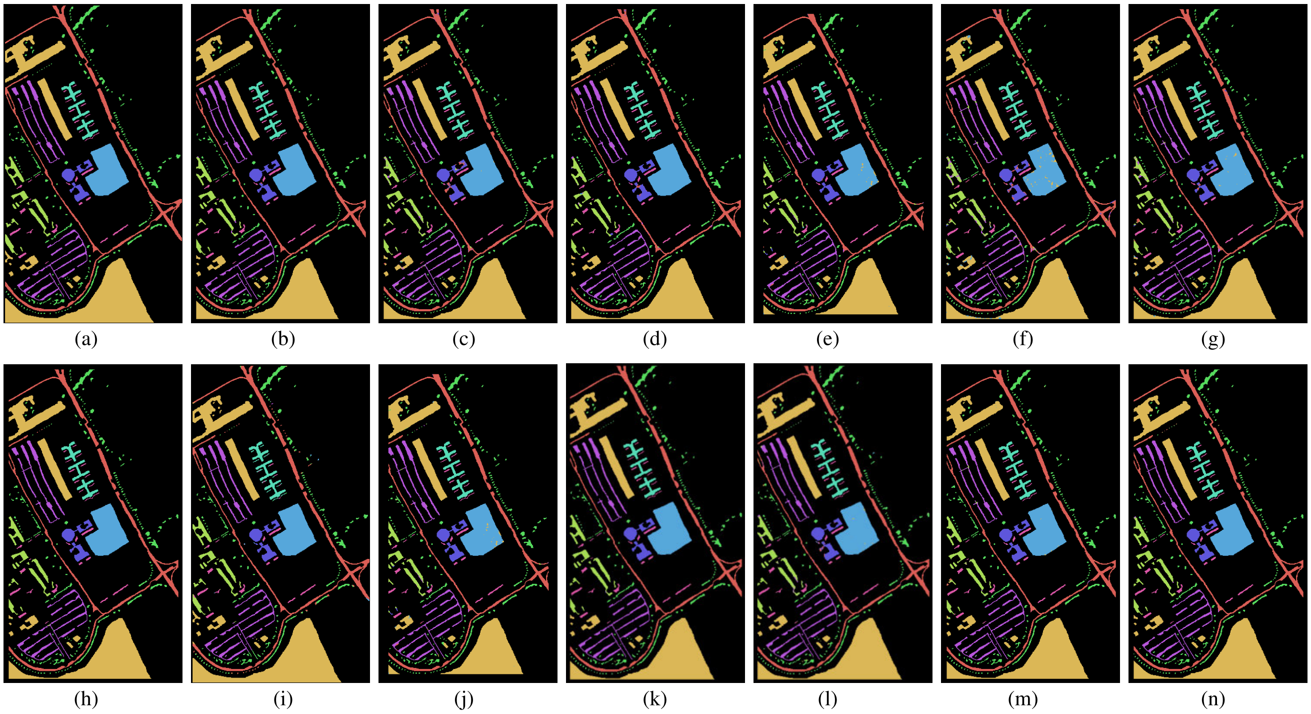
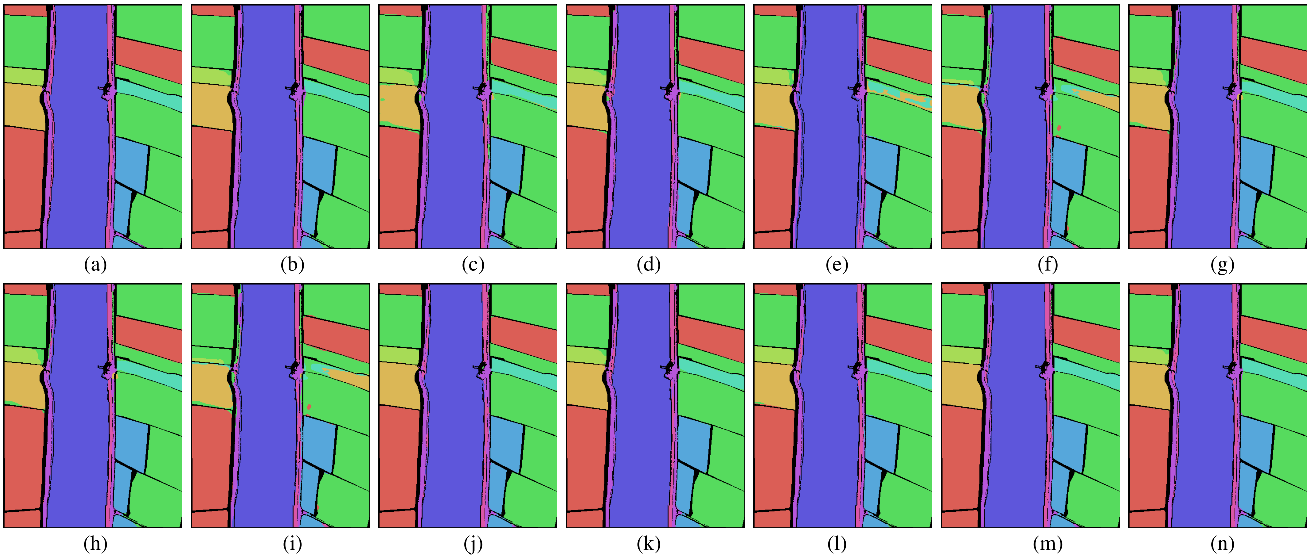


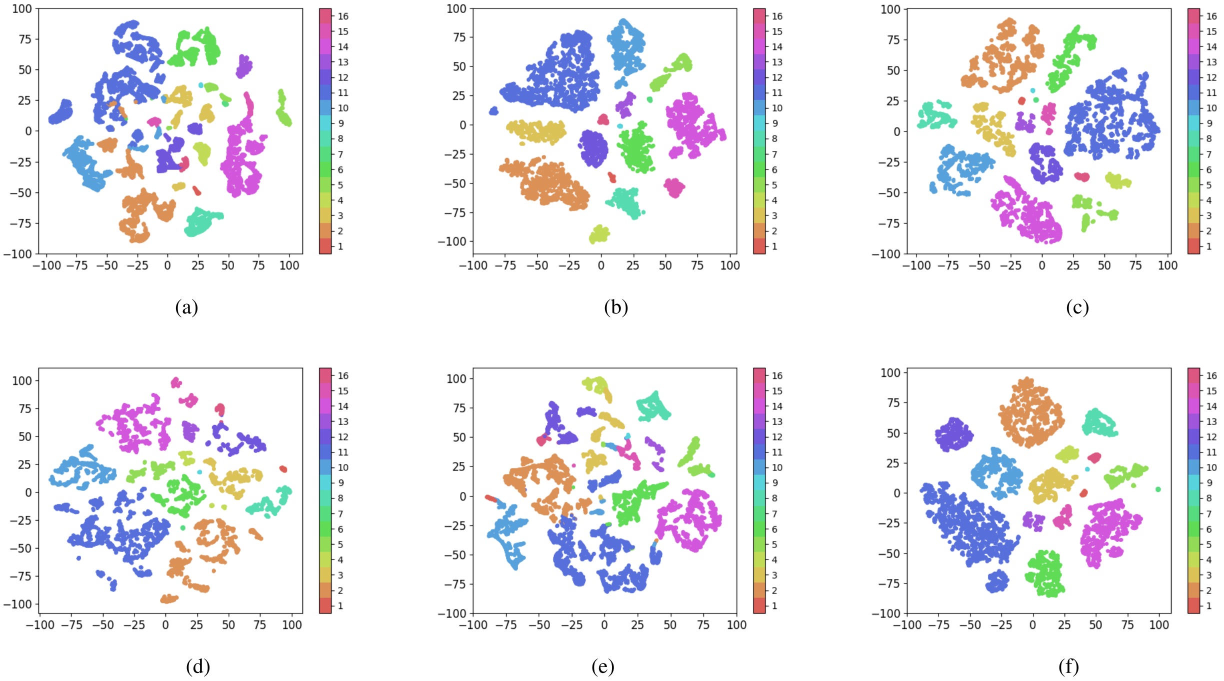
Fig. 7. Classification maps obtained by different methods on the PaviaU dataset (with 10% training samples). (a) Ground truth. (b) 2-D-CNN. (c) 3-D-CNN. (d) SyCNN. (e) HybridSN. (f) ViT. (g) Deep ViT. (h) CvT. (i) HiT. (j) SSFTT. (k) DCTN. (l) SS TMNet. (m) MorphFormer. (n) ACTN.

effective when the relationships between adjacent data points are significant, such as in images where nearby pixels are highly correlated. However, CNNs often find it challenging to effectively capture and interpret global representations or contextual relationships that span large distances or involve nonadjacent data points. This limitation is likely rooted in their architecture, which emphasizes local connections and is not as adept at modeling long-range relationships. In contrast, Transformer-based models, a relatively newer architecture in the field of machine learning, excel at capturing global relationships and contextual representations, but often falter when it comes to extracting and making use of localized features. As shown in Tables I–IV, the aforementioned characteristics of CNNs and Transformer-based models manifest differently when applied to the task of HSI classification. The tables illustrate that despite these differences, existing CNN models still represent the dominant approach in this field, and they consistently achieve superior classification results when compared to their Transformer-based counterparts. For instance, a commonly used 2-D-CNN model not only achieves the best performance among all models based on CNN architectures but also performs better classification tasks than any existing Transformer-based methods. This study provides empirical evidence that local features, which CNNs excel at capturing, play a crucial role in image classification tasks, and it also indicates that there is still room for improvement in Transformer-based methods. Finally, a closer examination of the results indicates that the proposed ACTN model, which is an advanced architecture that fuses both local and global features, outperforms traditional CNN models such as 2-D-CNN


(a) (b) (c) (d) (e) (f) (g)

(h) (i) (j) (k) (l) (m) (n)

Fig. 8. Classification maps obtained by different methods on the WHL dataset (with 1% training samples). (a) Ground truth. (b) 2-D-CNN. (c) 3-D-CNN. (d) SyCNN. (e) HybridSN. (f) ViT. (g) Deep ViT. (h) CvT. (i) HiT. (j) SSFTT. (k) MorphFormer. (l) SS TMNet. (m) DCTN. (n) ACTN.



(a) (b) (c)

(d) (e) (f)

Fig. 9. T-SNE results obtained by different methods on the IndianPines dataset (with 10% training samples). (a) 2-D-CNN. (b) HybridSN. (c) ViT. (d) SSFTT. (e) MorphFormer. (f) ACTN.

and 3-D-CNN. Specifically, ACTN surpasses the performance of CNNs by a notable margin of 1.26% (98.58% versus 97.32%). This compelling result once again underscores the importance of fusing local and global features and demonstrates the continued value of refining and improving CNN architectures.

Figs. 5–8 provide a visual comparative analysis of the performance of the proposed approach on three widely used benchmark datasets in the field of remote sensing image analysis: Indian Pines scene, Houston2013, and PaviaU. From these figures, it is clear that the proposed approach, specifically the implementation of the proposed ACTN, demonstrates an impressive capability to predict all classes with a high degree of accuracy. More importantly, it generates a classification map that remains faithful to the ground truth, suggesting that the proposed ACTN effectively integrates both local features and global representations. This could be attributed to the design of ACTN, which is specifically engineered to fuse these different types of features coherently, thereby enhancing its classification performance. On the other hand, a careful examination of the figures also reveals that most of the compared methods would produce classification maps with notable levels of noise. This finding is in alignment with the quantitative results summarized in the tables provided

TABLE V

COMPARISON OF COMPUTATIONAL COMPLEXITY

Methods	FLOPs(G)	Param (MB)	Training Time(s)	Testing Time(s)	OA (%)	AA (%)	κ (%)
ViT	2.71	52.22	727.89	3.43	90.27 ± 1.69	90.45 ±1.42	89.48 ± 1.83
DeepViT	2.71	52.22	1497.97	6.82	91.58 ± 1.50	91.38 ±1.43	90.89 ± 1.62
CvT	9.04	17.77	2758.06	11.65	93.52 ± 1.88	93.75 ±1.95	92.99 ± 2.04
HiT	1.81	16.94	112.04	6.7	95.20 ± 0.33	94.80 ±0.32	94.81 ± 0.35
SSFTT	0.97	87.04	389.62	1.59	98.15 ± 0.53	97.82 ±0.59	98.00 ± 0.58
MorphFormer	0.74	87.04	874.99	4.16	90.98 ± 7.71	90.99 ±7.46	90.24 ± 8.36
SS_TMNet	2.67	83.33	333.93	31.12	96.22± 0.35	95.15± 0.45	95.92± 0.38
DCTN	1.48	45.32	253.16	20.69	98.31± 0.16	97.32± 0.36	98.17 ±0.17
ACTN	1.79	14.04	870.97	3.65	**98.58 ± 0.16**	**98.16 ± 0.22**	**98.46 ± 0.17**

TABLE VI

STUDY OF THE INPUT SIZE

Sizes	IndianPines		PaviaU		Houston2013	
	OA	Kapp	OA	Kapp	OA	Kapp
9 × 9	**98.91 ± 0.10**	**98.76 ± 0.11**	**99.77 ± 0.03**	**99.70 ± 0.04**	**99.54 ± 0.18**	**99.50 ± 0.19**
11 × 11	98.04 ± 0.25	97.76 ± 0.28	99.59 ± 0.02	99.46 ± 0.03	99.22 ± 0.17	99.15 ± 0.19
13 × 13	96.82 ± 0.16	96.37 ± 0.18	99.47 ± 0.05	99.30 ± 0.07	99.00 ± 0.17	98.92 ± 0.18
15 × 15	95.40 ± 0.11	94.75 ± 0.12	99.09 ± 0.08	98.80 ± 0.10	98.58 ± 0.16	98.46 ± 0.17
17 × 17	93.90 ± 0.22	93.03 ± 0.25	98.61 ± 0.20	98.16 ± 0.26	98.17 ± 0.20	98.02 ± 0.22
19 × 19	92.32 ± 0.20	91.21 ± 0.23	97.50 ± 0.42	96.69 ± 0.55	97.27 ± 0.20	97.05 ± 0.22

TABLE VII

STUDY OF THE SIMILARITY FUNCTION ON THE HOUSTON2013 DATASET

Methods	OA (%)	κ (%)
CNN	95.50	95.14
ACTransformer_ED	97.59	97.51
ACTransformer_KL	**98.58**	**98.46**
ACTransformer_CS	98.03	97.88

above. One plausible explanation for this observation is that CNNs, which are popular in computer vision tasks, tend to overlook capturing the global representation. At the same time, Transformer-based models, which have recently gained significant attention in the field, might struggle to effectively fuse the local and global features. These findings underscore the importance of designing methods that can effectively balance the integration of both local and global features to improve the overall performance of remote sensing image classification tasks.

We also conducted the t-SNE visualization comparative analysis of the proposed ACTN and five comparison methods, including CNNs (such as 2-D-CNN and HybridSN) and Transformers (e.g., ViT, SSFTT, and MorphFormer). The visualization results of different methods on the IndianPines dataset are reported in Fig. 9. According to Fig. 9, we can find that the proposed ACTN is capable of reducing interclass misclassification and enhancing the in-class clustering. This is mainly because of the special integration of the ARFM and KL. This also demonstrates that the ACTN could effectively capture the global dependencies and local contextual information of the land cover from HSIs. We can also find that the CNNs and Transformers would bring much more interclass misclassification while limiting in achieving a better clustering result.

C. Comparison of Computational Complexity

We also analyzed the computational complexity of all comparison methods and ACTN based on the Houston2013 dataset. Table V reports the OA, FLOPs, and parameters. We can observe that ACTN performs better than the state-of-the-art Transformers-based methods. On this HSI classification task, the ACTN surpasses the LeViT by 0.05% while saving 17% parameters.

Our analysis revealed that CvT had the longest training time of 2758.06 s (about 46 min), which may be due to its higher FLOPs (9.04G) and smaller Param (MB) (17.77 MB). The high computational complexity and small model size could have prolonged the training process. DeepViT also had longer training times (1497.97 s), similar to CvT, likely due to its relatively high FLOPs and Param (MB). SSFTT had the shortest test time of 1.59 s, likely because of its lower FLOPs (0.97G) and larger Param (MB) (87.04 MB). The low computational complexity and relatively large model size could have contributed to its fast test speed. The ViT also showed shorter test times (3.43 and 3.65 s), likely due to its slightly higher FLOPs and Param (MB), but the relatively small model size and higher OA and κ performance could have contributed to rapid testing. Finally, ACTN exhibits superior performance, achieving an overall accuracy of 98.58% and a Kappa score of 98.46%. This indicates robust classification capabilities. Despite moderate computational requirements (1.79G FLOPs and 14.04 MB parameters), it achieves efficient training in 870.97 s and swift testing in 3.65 s. These attributes make ACTN a strong contender for applications requiring both accuracy and operational speed.

D. Ablation Studies

1) Ablation Study of the Input Size: In this subsection, we investigate the impact of the input size on the three different datasets. The input size is varied from 9 × 9 to 19 × 19. The classification results are listed in Table VI.

From Table VI, we can observe that the performances on all datasets are different by varying the different input sizes. In particular, the smaller input size could perform better classification results in terms of OA and κ. This is probably because a small input size could offer the high precise features to identify the land cover, while a large input size could bring some noise. It also demonstrates the robustness of the proposed ACTN. To evaluate the proposed ACTN, we set the input size to 15 × 15.

2) Ablation Study of the Similarity Function: Table VII reports the investigation on the effects of the similarity function based on the Houston2013 dataset. We choose three similarity functions, including cosine similarity (CS), Euclidean distance (ED), and KL, resulting in ACTN_CS, ACTN_ED, and ACTN_KL, respectively. We can observe that the similarity function is needed to improve the performance. The baseline CNN performs very poorly without any similarity functions. Moreover, the ACTN_KL produces the best performance. As such, we adopt the KL as the similarity function to compute the distance between local and global features.

3) Ablation Study of Different Modules: In Table VIII, we present the ablation investigation on the effects of ARFM, KL based on the Houston2013 dataset. It is noted that "T" denotes the Transformer. All the methods have the same CNN blocks and Transformer stages. We choose two baseline methods, including the 2-D-CNN and ViT. According to Table VIII, we can find that the 2-D-CNN achieves the classification results in terms of OA (OA = 97.32% ± 0.48) and κ (κ = 97.10% ± 0.51), while the ViT obtains poor results in terms of

TABLE VIII

STUDY OF DIFFERENT MODULES

Methods	ARFM	KL	OA (%)	κ (%)
2D-CNN	×	×	97.32 ± 0.48	97.10 ± 0.51
ViT	×	×	90.27 ± 1.69	89.48 ± 1.83
ACTN_ARFM_CNN	√	×	97.98 ± 0.36 (↑0.66%)	97.28 ± 0.32 (↑0.18%)
ACTN_ARFM_T	√	×	97.35 ± 0.38 (↑7.08%)	97.13 ± 0.41 (↑7.65%)
ACTN_ARFM_KL_CNN	√	√	98.31 ± 0.19(↑0.99%)	98.17 ± 0.21(↑1.07%)
ACTN_ARFM_KL_T	√	√	98.03 ± 0.48(↑7.76%)	97.87 ± 0.52(↑8.39%)
ACTN_ARFM_KL_Final	√	√	**98.58 ± 0.16**	**98.46 ± 1.36**

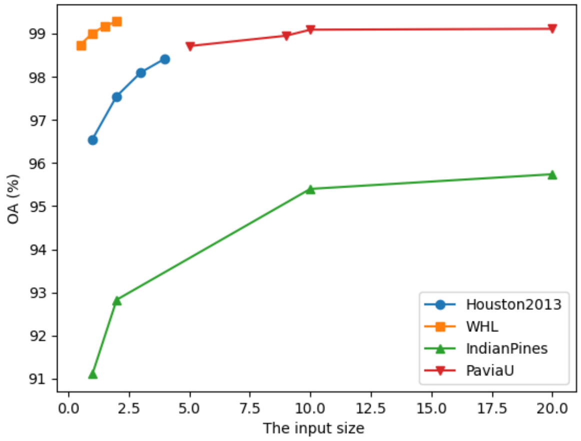


Fig. 10. Classification results (OA) achieved by the proposed ACTN with a varying number of training samples on four benchmark datasets.

OA (OA = 90.27% ± 1.69) and κ (κ = 89.48% ± 1.83). This is mainly because the CNN could capture the local contextual features to achieve a better result. The ViT is capable of exploring the global dependencies, but neglecting the local features.

From Table VIII, we can find that the new methods based on ARFM and KL achieve better performance in terms of OA and Kappa than the 2-D-CNN and ViT. We can observe that the ACTN_ARFM_CNN outperforms the 2-D-CNN in terms of OA (0.69%) and κ (1.07%) and the ACNT_ARFM_T outperforms the ViT in terms of OA (6.9%) and κ (8.39%). The reason is that the ARFM can interactively fuse both corresponding local features and global representations, thus reducing noise in these areas. Additionally, it can enhance and integrate any potentially missing information within the broader context of the CNN and Transformer branches. Especially, the ACTN_ARFM_KL_CNN outperforms the 2-D-CNN in terms of OA (0.99%) and κ (1.07%) and the ACNT_ARFM_KL_T outperforms the ViT in terms of OA (7.76%) and κ (8.39%). This is mainly because of the integration of ARFM and KL, which help the ACTN to capture much more precise discriminate features to identify the land cover.

Finally, we fuse the features from the 2-D-CNN and Transformer branches to classify the land cover, resulting in ACTN_ARFM_KL_Final that achieves the best performances in terms of OA (OA = 98.58% ± 0.16) and κ (κ = 98.46% ± 1.36). The reason is because of the fusion of the best local contextual features and global dependencies.

4) Ablation Study of Training Samples: We conduct extensive experiments on four benchmark HSI datasets, varying the training sample size. From Fig. 10, we observe that as the number of training samples increases, the overall accuracy (OA) metrics show incremental improvements. Specifically, we see an upward trend from 97.71% to 98.61%, indicating that with more training samples, the method's ability to correctly classify and account for agreement beyond chance improves consistently. Similarly, overall accuracy increases from 98.27% to 98.95% as the number of training samples increases. This improvement underscores the method's enhanced capability to correctly classify instances across all classes as more data becomes available for training. Therefore, increasing the number of training samples positively impacts the method by improving its classification accuracy and robustness, suggesting that larger training datasets lead to more accurate and reliable classification outcomes.

V. CONCLUSION

In this article, we propose a hybrid network called ACTN for HSI classification by taking advantage of the CNN and vision Transformer. Different from previous methods, we propose an ARFM to adaptive fuse the corresponding local features and global representations without extra noise feature/representation. To improve the representation learning capability of ACTN, we supervise the classifiers by introducing the cosine similarity function. Moreover, the cosine similarity function could help ACTN enhance the local features and global representations' interaction capabilities. Finally, extensive experiments demonstrate that the proposed ACTN outperforms both the state-of-the-art CNNs and visual Transformers. In the future, we will continue studying other fusion modules to further enhance the learning capability of the Transformer.

REFERENCES

[1] F. Chen, K. Wang, T. Van De Voorde, and T. F. Tang, "Mapping urban land cover from high spatial resolution hyperspectral data: An approach based on simultaneously unmixing similar pixels with jointly sparse spectral mixture analysis," *Remote Sens. Environ.*, vol. 196, pp. 324–342, Jul. 2017.

[2] X. Kang, Z. Wang, P. Duan, and X. Wei, "The potential of hyperspectral image classification for oil spill mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5538415.

[3] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Cambridge, MA, USA: MIT Press, 2001.

[4] Q. Ye, P. Huang, Z. Zhang, Y. Zheng, L. Fu, and W. Yang, "Multiview learning with robust double-sided twin SVM," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12745–12758, Dec. 2022.

[5] Y.-N. Chen, T. Thaipisutikul, C.-C. Han, T.-J. Liu, and K.-C. Fan, "Feature line embedding based on support vector machine for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 1, p. 130, Jan. 2021.

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

[7] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 6308–6325, 2020.

[8] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

[9] S. Jia, B. Deng, J. Zhu, X. Jia, and Q. Li, "Local binary pattern-based hyperspectral image classification with superpixel guidance," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 749–759, Feb. 2018.

[10] W. Liao, A. Pizurica, P. Scheunders, W. Philips, and Y. Pi, "Semisupervised local discriminant analysis for feature extraction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 184–198, Jan. 2013.

[11] Q. Wang, Z. Meng, and X. Li, "Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2077–2081, Nov. 2017.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 84–90.

[13] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[14] 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.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 4700–4708.

[16] 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.

[17] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.

[18] X. Yang et al., "Synergistic 2D/3D convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 12, p. 2033, Jun. 2020.

[19] 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, Jan. 2019.

[20] W. Sun and Q. Du, "Hyperspectral band selection: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 118–139, Jun. 2019.

[21] W. Sun, L. Zhang, B. Du, W. Li, and Y. M. Lai, "Band selection using improved sparse subspace clustering for hyperspectral imagery classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2784–2797, Jun. 2015.

[22] W. Sun, G. Yang, J. Peng, and Q. Du, "Lateral-slice sparse tensor robust principal component analysis for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 1, pp. 107–111, Jan. 2020.

[23] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jeǵou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2021, pp. 10347–10357.

[24] Y. Jiang, S. Chang, and Z. Wang, "TransGAN: Two pure transformers can make one strong GAN, and that can scale up," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 14745–14758.

[25] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–21.

[26] B. Graham et al., "LeViT: A vision transformer in ConvNet's clothing for faster inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12259–12269.

[27] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[29] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[30] 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.

[31] Y. Xu, B. Du, and L. Zhang, "Self-attention context network: Addressing the threat of adversarial attacks for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 30, pp. 8671–8685, 2021.

[32] Z. Wang, B. Chen, R. Lu, H. Zhang, H. Liu, and P. K. Varshney, "FusionNet: An unsupervised convolutional variational network for hyperspectral and multispectral image fusion," *IEEE Trans. Image Process.*, vol. 29, pp. 7565–7577, 2020.

[33] C. Zhao, W. Zhu, and S. Feng, "Superpixel guided deformable convolution network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 3838–3851, 2022.

[34] V. Sharma, A. Diba, T. Tuytelaars, and L. Van Gool, "Hyperspectral CNN for image classification & band selection, with application to face recognition," KU Leuven, ESAT, Leuven, Belgium, Tech. Rep. KUL/ESAT/PSI/1604, 2016.

[35] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A fast and compact 3-D CNN for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[36] P. Ribalta Lorenzo, L. Tulczyjew, M. Marcinkiewicz, and J. Nalepa, "Band selection from hyperspectral images using attention-based convolutional neural networks," 2018, *arXiv:1811.02667*.

[37] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[38] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[39] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 7132–7141.

[40] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2018, pp. 1–11.

[41] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3588–3597.

[42] D. Hong et al., "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5518615.

[43] S. K. Roy, A. Deria, C. Shah, J. M. Haut, Q. Du, and A. Plaza, "Spectral–spatial morphological attention transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5503615.

[44] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522214.

[45] B. Tu, X. Liao, Q. Li, Y. Peng, and A. Plaza, "Local semantic feature aggregation-based transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5536115.

[46] X. Yang, W. Cao, Y. Lu, and Y. Zhou, "Hyperspectral image transformer classification networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5528715.

[47] Z. Peng et al., "Conformer: Local features coupling global representations for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 367–376.

[48] J. Yang et al., "Focal attention for long-range interactions in vision transformers," in *Proc. Neural Inf. Process. Syst.*, vol. 34, May 2021, pp. 30008–30022. [Online]. Available: https://api.semanticscholar.org/CorpusID:245011146

[49] D. Zhou et al., "DeepViT: Towards deeper vision transformer," 2021, *arXiv:2103.11886*.

[50] X. Huang, Y. Zhou, X. Yang, X. Zhu, and K. Wang, "SS-TMNet: Spatial–spectral transformer network with multi-scale convolution for hyperspectral image classification," *Remote Sens.*, vol. 15, no. 5, p. 1206, Feb. 2023.

[51] Y. Zhou, X. Huang, X. Yang, J. Peng, and Y. Ban, "DCTN: Dual-branch convolutional transformer network with efficient interactive self-attention for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5508616.

Xiaofei Yang (Member, IEEE) received the B.S. degree from Suihua University, Suihua, China, in 2011, and the M.S. and Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 2014 and 2019, respectively.

He was a Post-Doctoral with the Department of Computer and Information Science, University of Macau, Macau, China, from 2020 to 2023. He is currently a Lecturer with the School of Electronics and Communications Engineering, Guangzhou University, Guangzhou, China. His research interests include semi-supervised learning, deep learning, remote sensing, transfer learning, and graph mining.

Weijia Cao (Member, IEEE) received the master's and Ph.D. degrees in computer science from the University of Macau, Macau, China, in 2013 and 2017, respectively.

She is currently an Associate Researcher with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China. Her research interests revolve around machine learning and remote sensing image processing.

Dong Tang (Member, IEEE) received the B.S. degree from Nanhua University, Hengyang, China, in 1989, the M.S. degree from Hunan University, Changsha, China, in 1999, and the Ph.D. degree in communications and information systems from Sun Yat-sen University, Guangzhou, China, in 2006.

From 2014 to 2015, he was a Research Fellow with the University of California, Irvine, CA, USA. Currently, he is a Professor with the School of Electronics and Communications Engineering, Guangzhou University, Guangzhou. His main research interests include signal processing, deep learning, intelligent network systems, and wireless communications.

Yicong Zhou (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Tufts University, Medford, MA, USA, in 2010.

He is a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE) and was recognized as one of the "Highly Cited Researchers" in 2020, 2021, 2023, and 2024. He serves as a Senior Area Editor for IEEE Transactions on Circuits and Systems for Video Technology and an Associate Editor for IEEE Transactions on Cybernetics, IEEE Transactions on Neural Networks and Learning Systems, and IEEE Transactions on Geoscience and Remote Sensing.

Yao Lu received the B.S. degree in computer science and technology from Huaqiao University, Xiamen, China, in 2015, and the Ph.D. degree in computer applied technology from Harbin Institute of Technology, Harbin, China, in 2020.

She was a Post-Doctoral Fellow at the University of Macau, Macau, China, from 2020 to 2021. She is currently an Assistant Professor with the Biocomputing Research Center, Harbin Institute of Technology, Shenzhen, China. Her research interests include pattern recognition, deep learning, computer vision, and relevant applications.