Multiscanning-Based RNN–Transformer for Hyperspectral Image Classification

Weilian Zhou, *Graduate Student Member, IEEE*, Sei-Ichiro Kamata, *Member, IEEE*, Haipeng Wang, *Senior Member, IEEE*, and Xi Xue, *Student Member, IEEE*

***Abstract*— The goal of hyperspectral image (HSI) classification is to assign land-cover labels to each HSI pixel in a patchwise manner. Recently, sequential models, such as recurrent neural networks (RNNs), have been developed as HSI classifiers, which need to scan the HSI patch into a pixel sequence with the scanning order first. However, RNNs have a biased ordering that cannot effectively allocate attention to each pixel in the sequence, and previous methods that use multiple scanning orders to average the features of RNNs are limited by the validity of these orders. To solve this issue, it is naturally inspired by Transformer and its self-attention to discriminatively distribute proper attention for each pixel of the pixel sequence and each scanning order. Hence, in this study, we further develop the sequential HSI classifiers by a specially designed RNN–Transformer (RT) model to feature the multiple sequential characters of the HSI pixels in the HSI patch. Specifically, we introduce a multiscanning-controlled positional embedding strategy for the RT model to complement multiple feature fusion. Furthermore, the RT encoder is proposed for integrating ordering bias and attention reallocation for feature generation at the sequence level. In addition, the spectral–spatial-based soft masked self-attention (SMSA) is proposed for suitable feature enhancement. Finally, an additional fusion Transformer (FT) is deployed for scanning order-level attention allocation. As a result, the whole network can achieve competitive classification performance on four accessible datasets than other state-of-the-art methods. Our study further extends the research on sequential HSI classifiers.**

***Index Terms*— Hyperspectral image (HSI) classification, multiscanning strategy, recurrent neural network (RNN), Transformer.**

I. Introduction

HYPERSPECTRAL imaging technology is critical in the remote sensing field due to its high resolution in the spatial and spectral domains [1]. Furthermore, hyperspectral sensors capture information across the electromagnetic spectrum, from visible to infrared light, allowing the recording of chemical or physical properties of various materials [2]. This results in the hyperspectral image (HSI), a 3-D data cube, which provides valuable information for environmental monitoring [3], mining exploration [4], and other applications. Therefore, the HSI classification task, which involves assigning a label to each pixel, is crucial. A high accuracy in HSI classification results leads to improved quality in these applications and is a widely researched topic in the field of remote sensing [5].

In recent years, deep learning-based models have been extensively developed for HSI classification tasks [6], [7]. These models generally follow the patchwise learning framework, which aims to assign a semantic label to each pixel by processing a patch around it [8]. Recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM), have achieved impressive results in this task from a unique perspective. These models work based on a many-to-one or many-to-many scheme, capturing the contextual features among all pixels in a patch and outputting a single feature representation.

The first RNN-based HSI classifier was proposed by Mou et al. [9], who inputted one pixel's spectral signature into the deep learning model and set the step length to equal the number of bands. Paoletti et al. [10] proposed a scalable and efficient RNN to simplify the internal complexity of the original RNN model for HSI classification. Zhou et al. [11] proposed a spatial RNN that slices the first principle component of the HSI row by row, treating each row as a step for the LSTM algorithm. Hang et al. [12] divided a cropped HSI patch into several equal groups along the spectral domain, using each group as a step input for the RNN. Zhang et al. [13] scanned an HSI patch into a pixel sequence, deploying each pixel with its spectral information as one step for the RNN. Shi and Pun [14] proposed a multiscale CNN-based hierarchical RNN for HSI classification. Finally, in [15], the validity of the multiscanning strategy was investigated for generating features by scanning an HSI patch into multiple pixel sequences with different orderings. The strategy was found to be effective and resulted in significant improvement.

Although previous RNN-based HSI classifiers have been successful, there are still some problems that need to be addressed. One issue is that these models use the last step's output as the final feature for classification without considering the importance of other steps. Some methods implement attention-based weighted summation of each step's output, but the attention values may be biased toward the later pixels [16], leading to a weakening of the information from the central pixel, which should be dominant in determining the final feature. This can result in misclassification when the cropped patch has the same label as its central pixel [7].

Manuscript received 6 December 2022; revised 17 February 2023 and 20 April 2023; accepted 8 May 2023. Date of publication 17 May 2023; date of current version 30 May 2023. *(Corresponding author: Sei-Ichiro Kamata.)*

Weilian Zhou, Sei-Ichiro Kamata, and Xi Xue are with the Image Media Laboratory, Graduate School of Information, Production and Systems, Waseda University, Fukuoka 808-0135, Japan (e-mail: zhouweilian1904@akane.waseda.jp; kam@waseda.jp).

Haipeng Wang is with the Key Laboratory of Electromagnetic Waves (EMW) Information, Fudan University, Shanghai 200433, China (e-mail: hpwang@fudan.edu.cn).

Digital Object Identifier 10.1109/TGRS.2023.3277014

Second, RNN struggles with larger HSI patches as more pixels with different class labels than the central pixel may be present in the pixel sequence, potentially dominating the final decision of the output features. This is due to RNN's inflexibility caused by its reliance on sequence ordering. For example, if the interfering pixels are primarily located toward the end of the pixel sequence, the output features from the final step may be negatively impacted and unable to accurately reflect the land-cover meaning of the HSI patch.

Third, the prior work of the multiscanning strategy processes an HSI patch with various scanning orders to combine complementary contextual features. However, the appropriate consideration of the importance of each scanning order is lacking, which can refer to the previous problem. Some scanning orders might place more interfering pixels later in the pixel sequence, affecting the features negatively. Simply averaging the outputs from different scanning orders may result in decreased discriminability of the features.

Recently, the Transformer model [17], which utilizes a self-attention mechanism to determine the interdependence between elements in a sequence, has been introduced and applied to various domains. Specifically, the vision Transformer (ViT) [18] has been adopted in HSI classification and improved upon by various perspectives (i.e., pixel sequence, patch sequence, or band sequence). For example, Hong et al. [19] proposed a ViT-based HSI classification model with a refined Transformer encoder for band features. He et al. [20] proposed a bidirectional Transformer encoder that allows for flexible and dynamic cropping regions. Qing et al. [21] combined the Transformer with the convolutional block attention module (CBAM) attention block [22] for improved spectral attention. Yang et al. [23] created the hyperspectral image Transformer (HiT) by integrating convolution operations into the Transformer to incorporate both spectral and spatial features [24]. Gao et al. [25] developed the deep Transformer-in-Transformer (TNT) module to extract both patch-level and pixel-level features. Finally, Ibañez et al. [26] proposed the masked autoencoding spectral–spatial Transformer (MAEST) with both reconstruction and classification paths to refine the Transformer features.

In contrast to RNN for HSI pixel sequences, the Transformer processes all pixels simultaneously, with each pixel receiving individual attention for a particular step (i.e., a single pixel considers all the information from the others and generates a weighted representation). This enables it to assign more precise and diverse attention weights to all pixels, thus avoiding the omission of important information [27]. Thus, it is believed that the Transformer is better suited to allowing the central pixel to dominate the final classification feature compared with RNN (i.e., if the central output is used as the feature representation). In addition, for pixel sequences with different scanning orders, the Transformer assigns more rational and unequal weights based on the scanning order, enabling it to judge the impact of each order and produce a more discriminative feature fusion.

However, some studies have shown that the lack of recurrent modeling in the Transformer limits its potential for improvement empirically [28]. Meanwhile, modeling recurrence is crucial for capturing important properties of input sequences, such as structural representations [29], [30] and positional embedding [31]. These are precisely the areas where self-attention in the Transformer falls short. In addition, [32] and [33] suggest that representations learned by both Transformer-based and RNN-based encoders are complementary to each other.

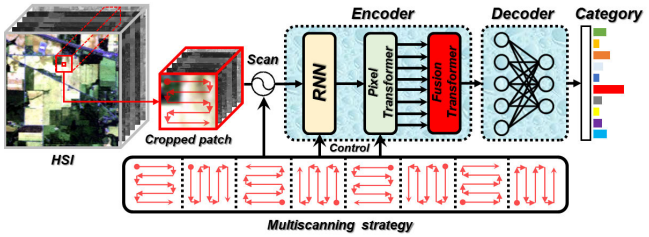


Fig. 1. Concept of the proposed method involves cropping patches from the original HSI, which are then scanned into multiple pixel sequences through a multiscanning strategy. The scanning order controls the positional information in both the RNN and PT components of the encoder. The features from different scanning orders are then fused in the FT. The decoder consists of fully connected layers, normalization layers, and a softmax layer for categorizing the predictions.

Hence, in this study, we aim to improve the multiscanning strategy [15] by incorporating a specially designed RNN–Transformer (RT) model to capture the multiple sequential characteristics of HSI pixels. A novel multiscanning-controlled positional embedding is also proposed to consider the spatial contextual dependencies in different positions. The RT model is a combination of the strengths of RNN and Transformer, incorporating RNN's ordering bias and Transformer's self-attention weights for feature generation. This model allocates scanning order-based attention to determine their positive or negative impact. In addition, a spectral–spatial-based soft mask is proposed for the self-attention layer to eliminate the influence of interfering pixels. The overall concept is illustrated in Fig. 1.

The proposed method was tested on four HSI datasets, and the results showed that the RNN improved the performance of the Transformer, highlighting the importance of modeling recurrence. The multiscanning strategy further boosted the model's results. In addition, the spectral–spatial-based soft mask further ensured the robustness of the model.

This study's distinction from other methods lies in the absence of convolutional neural networks (CNNs) while still achieving competitive classification results. The main contributions of this article can be summarized as follows.

1) This is the first study presenting a Transformer model for HSI classification that is practical and based on multiscanning. The joint utilization of RNN and Transformer has demonstrated its viability as a solution for HSI classification tasks.
2) This study presents a novel concept of using spectral–spatial-based soft self-attention masks for HSI classification, which is a pioneering work in this field. Our aim is to bring a fresh outlook to the feature enhancement process in the Transformer era.

TABLE I
DEFINITION OF NOTATIONS USED IN THE PROPOSED METHOD

Notation	Definition	Type	Size
$\boldsymbol{X}^{(i,j)}$	A cropped HSI patch centered at $\boldsymbol{x}^{(i,j)}$	Tensor	$p \times p \times C$
p	The patch size of cropped HSI patch	scalar	1×1
$\boldsymbol{x}^{(i,j)}$	A pixel with its spatial coordinate (i, j) in the original HSI	vector	$1 \times C$
$\boldsymbol{S}_m^{(i,j)}$	A pixel-sequence with m-th order	matrix	$p^2 \times C$
$\boldsymbol{P}_m^{(i,j)}$	Output from RNN at m-th order	matrix	$p^2 \times C$ord
$\boldsymbol{F}_m^{(i,j)}$	Output from Pixel Transformer at m-th scanning order	matrix	$p^2 \times d$
$\boldsymbol{Y}^{(i,j)}$	Concatenated multiscanning features	matrix	$m \times d$
$\boldsymbol{E}^{(i,j)}$	The attention for each scanning order in multiscanning	vector	$m \times 1$
$\boldsymbol{y}_m^{(i,j)}$	A feature representation from sequential models under m-th scanning order	vector	$1 \times d$
$\boldsymbol{y}_{cls}^{(i,j)}$	Final feature for decoder (classification)	vector	$1 \times d$
d	the dimension of the feature embeddings	scalar	1×1
m	The m-th scanning order in multiscanning, $m = 1, 2, ..., M$	scalar	1×1
$\boldsymbol{A}_m^{(i,j)}$	The attention for each pixel in a pixel sequence	vector	$p^2 \times 1$
l	the l-th layer in the network	scalar	1×1
$\boldsymbol{M}_m^{spa}$	Spatial-based soft attention mask	matrix	$p^2 \times p^2$
$\boldsymbol{M}_m^{spe}$	Spectral-based soft attention mask	matrix	$p^2 \times p^2$

3) We reevaluate the effectiveness of using a pure sequential model for HSI classification. Although the combination of RNN and Transformer has not been extensively explored in the field, we hope this research will contribute to the advancement of related studies in the future.

The rest of this article is organized as follows. The detailed methodology is presented in Section II. In Section III, we describe the experimental details and discuss the results. Section IV concludes this study. Appendixes A and B are prepared at last.

II. MULTISCANNING-BASED RT

This section presents the proposed strategy in the following parts: 1) the proposed strategy; 2) multiscanning-controlled positional embedding; 3) RT encoder; 4) feature selection (FS) layer; 5) spectral–spatial-based soft masked self-attention (SMSA); and 6) multiscanning feature fusion Transformer (FT).

A. Proposed Strategy

The corresponding notations are listed in Table I.

The original HSI ($\boldsymbol{X} \in \mathbb{R}^{H\times W\times C}$) can be defined as follows:

$$\boldsymbol{X} = \left\{ \boldsymbol{x}^{(i,j)} \in \mathbb{R}^{1\times C} \middle|_{j=0,1,...,W-1}^{i=0,1,...,H-1} \right\} \tag{1}$$

where $\boldsymbol{x}^{(i,j)}$ represents spectral signature at spatial position (i, j). Therefore

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}^{(0,0)} & \boldsymbol{x}^{(0,1)} & \dots & \boldsymbol{x}^{(0,W-1)} \\ \boldsymbol{x}^{(1,0)} & \boldsymbol{x}^{(1,1)} & \dots & \boldsymbol{x}^{(1,W-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}^{(H-1,0)} & \boldsymbol{x}^{(H-1,1)} & \dots & \boldsymbol{x}^{(H-1,W-1)}. \end{pmatrix} \tag{2}$$

For one cropped HSI patch $\boldsymbol{X}^{(i,j)}$ central at $\boldsymbol{x}^{(i,j)}$ with patch size p, an odd number, as an example

$$\boldsymbol{X}^{(i,j)} = \left\{ \boldsymbol{x}^{(i+\alpha,j+\beta)} \middle| \alpha, \beta = -\frac{p-1}{2} \sim \frac{p-1}{2} \right\} \tag{3}$$

where $\alpha, \beta \in \mathbb{Z}$. $i+\alpha$ and $j+\beta$ record the position in $\boldsymbol{X}^{(i,j)}$.

Take $p = 5$ as an example

$$\boldsymbol{X}^{(i,j)} = \begin{pmatrix} \boldsymbol{x}^{(i-2,j-2)} & \boldsymbol{x}^{(i-2,j-1)} & \dots & \boldsymbol{x}^{(i-2,j+2)} \\ \boldsymbol{x}^{(i-1,j-2)} & \boldsymbol{x}^{(i-1,j-1)} & \dots & \boldsymbol{x}^{(i-1,j+2)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}^{(i+2,j-2)} & \boldsymbol{x}^{(i+2,j-1)} & \dots & \boldsymbol{x}^{(i+2,j+2)} \end{pmatrix} \tag{4}$$

where $2 \le i \le H-3$, $2 \le j \le W-3$, and $\boldsymbol{X}^{(i,j)} \in \mathbb{R}^{p\times p\times C}$. $\boldsymbol{X}^{(i,j)}[2, 2] = \boldsymbol{x}^{(i,j)}$. Consequently, using the multiscanning strategy, the HSI patch $\boldsymbol{X}^{(i,j)}$ will be transferred into several pixel sequences $\{\boldsymbol{S}_m^{(i,j)} \in \mathbb{R}^{p^2\times C}, m = 1, 2, \dots, M\}$. For example, $\boldsymbol{S}_1^{(i,j)}$ could be arranged as follows:

$$\begin{aligned} \boldsymbol{S}_1^{(i,j)} = \big[& \boldsymbol{x}^{(i-2,j-2)}, \boldsymbol{x}^{(i-2,j-1)}, \boldsymbol{x}^{(i-2,j)}, \boldsymbol{x}^{(i-2,j+1)}, \\ & \boldsymbol{x}^{(i-2,j+2)}, \boldsymbol{x}^{(i-1,j+2)}, \dots\dots, \boldsymbol{x}^{(i+2,j+2)} \big]^\top. \end{aligned} \tag{5}$$

Subsequently, the pixel sequences will undergo the proposed RT encoders to produce the final output of the encoder

$$\boldsymbol{F}_m^{(i,j)} = \mathrm{RT}^L\left(\boldsymbol{S}_m^{(i,j)}\right) \tag{6}$$

where RT denotes the operations in RT encoder, including RNN and pixel Transformer (PT), which is detailed in Section II-C. $l = 1, 2, \dots, L$, represents the layers of RT encoder. For instance, $\boldsymbol{F}_1^{(i,j)} \in \mathbb{R}^{p^2\times d}$ can be defined as follows:

$$\begin{aligned} \boldsymbol{F}_1^{(i,j)} = \big[& \boldsymbol{f}^{(i-2,j-2)}, \boldsymbol{f}^{(i-2,j-1)}, \boldsymbol{f}^{(i-2,j)}, \boldsymbol{f}^{(i-2,j+1)}, \\ & \boldsymbol{f}^{(i-2,j+2)}, \boldsymbol{f}^{(i-1,j+2)}, \dots, \boldsymbol{f}^{(i+2,j+2)} \big]^\top \end{aligned} \tag{7}$$

where each element of $\boldsymbol{F}_1^{(i,j)}$ is corresponding to (5). d is the feature dimension set in RT encoder.

Next, the output features $\boldsymbol{F}_m^{(i,j)}$ are fed into the FS layer to get an intensified feature representation ($\boldsymbol{y}_m^{(i,j)} \in \mathbb{R}^{1\times d}$) separately as follows:

$$\boldsymbol{y}_m^{(i,j)} = \mathrm{FS}\left(\boldsymbol{F}_m^{(i,j)}\right) \tag{8}$$

where FS represents the FS layer operation. It will be detailed in Section II-D.

Then, all feature representations $\{\boldsymbol{y}_m^{(i,j)} \in \mathbb{R}^{1\times d}, m = 1, 2, \dots, M\}$ are prepared for the FT to do the multiscanning-feature fusion by defining and identifying a new class token

$$\boldsymbol{y}_{\mathrm{cls}}^{(i,j)} = \mathrm{FT}(\boldsymbol{Y}^{(i,j)}) \tag{9}$$

where $\boldsymbol{Y}^{(i,j)} = [\boldsymbol{y}_{\mathrm{cls}}^{(i,j)}, \boldsymbol{y}_1^{(i,j)}, \dots, \boldsymbol{y}_M^{(i,j)}]^\top \in \mathbb{R}^{(1+M)\times d}$. FT denotes the FT process. Details are discussed in Section II-F.

Finally, the class token $\boldsymbol{y}_{\mathrm{cls}}^{(i,j)}$ will go through the decoder to interpret the category of central pixel ($\boldsymbol{x}^{(i,j)}$).

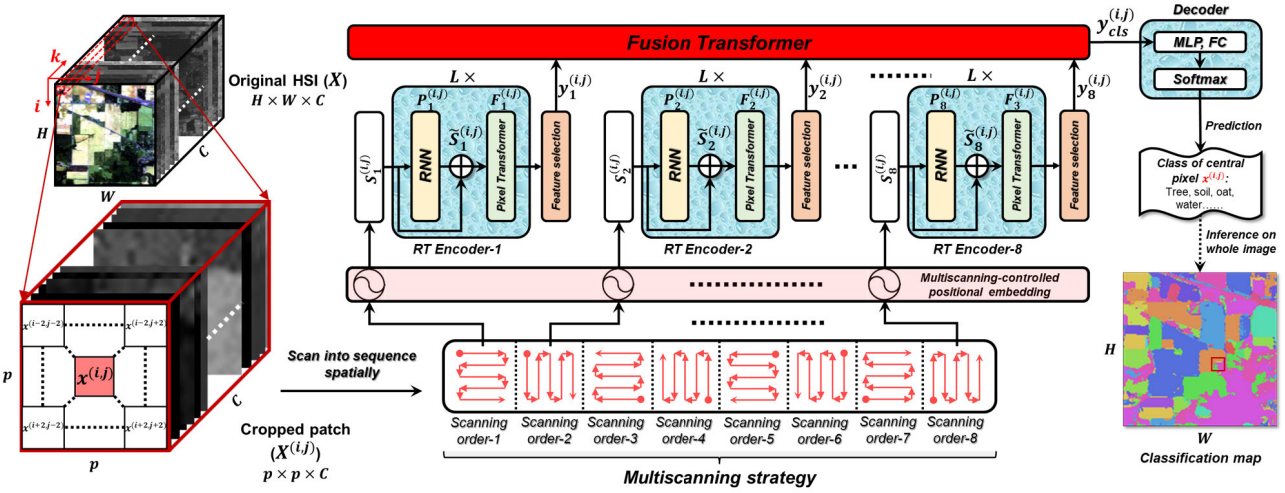


Fig. 2. Proposed method involves the following steps: 1) cropping an HSI patch centered around a central pixel $\boldsymbol{x}^{(i,j)}$ of size $p = 5$ as an example; 2) scanning the cropped patch into multiple pixel sequences (i.e., $\boldsymbol{S}_m^{(i,j)}, m = 1, 2, \ldots, M$) using the multiscanning strategy; 3) applying the RT encoder to each pixel sequence to obtain its feature representation $\boldsymbol{F}_m^{(i,j)}$; 4) using the FS layer to derive a new representation $\boldsymbol{y}_m^{(i,j)}$ from $\boldsymbol{F}_m^{(i,j)}$; 5) fusing all feature representations $\boldsymbol{y}_m^{(i,j)}, m = 1, 2, \ldots, M$ through the FT encoder; 6) using the decoder to interpret the category of the central pixel ($\boldsymbol{x}^{(i,j)}$) based on the fused feature $\boldsymbol{y}_{\text{cls}}^{(i,j)}$; and 7) repeating the process for each central pixel to get the final classification map of the entire image.

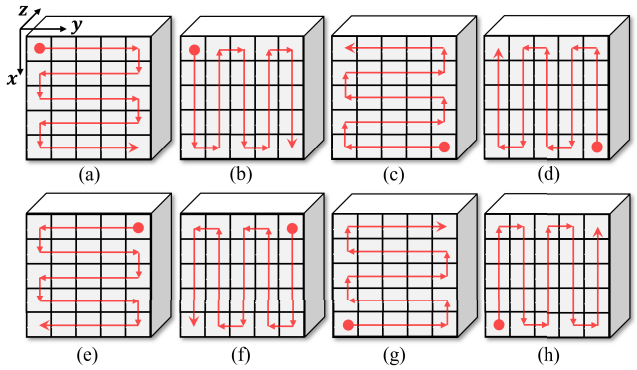


Fig. 3. Illustration of multiscanning strategy on a 5 × 5 HSI patch. Best view in lines. (a) Ordering-1. (b) Ordering-2. (c) Ordering-3. (d) Ordering-4. (e) Ordering-5. (f) Ordering-6. (g) Ordering-7. (h) Ordering-8.

B. Multiscanning-Controlled Positional Embedding

The proper encoding of positions is crucial, as positional embedding has a significant impact on the Transformer's structural comprehension and the provision of ordering information [41]. However, our observation, inspired by the multiscanning strategy, suggests that different arrangements of inputs in a sequence can lead to varied output features for sequential models. As demonstrated in Appendix A, altering the current input $\boldsymbol{S}_m^{(i,j)}[s]$ in training will result in distinctive $\boldsymbol{P}_m^{(i,j)}[s]$ successively. The arrangement of RNNs in this example is comparable to the positional embedding in Transformer. As a result, we believe that adjusting the positional information for Transformer will yield different output features. By appropriately blending these features, it could lead to more discriminative outputs. Thus, we enhance the positional embedding with the multiscanning strategy.

Therefore, the proposed multiscanning-controlled positional embedding ($\boldsymbol{P}^{(i,j)}$) can be generated as follows:

$$\boldsymbol{P}^{(i,j)} = \left\{ \text{RNN}\left(\boldsymbol{S}_m^{(i,j)}\right) \middle| m = 1, 2, \ldots, M \right\} \tag{10}$$

where RNN is the recurrent operation, listed in Appendix A.

Therefore, in this research, the multiscanning-controlled positional embedding ($\boldsymbol{P}^{(i,j)}$) is generated through the U-Turn scanning pattern with $M = 8$, as depicted in Fig. 3. This approach aligns with the previous work in [15].

C. RT Encoder

The ordering mechanism of RNNs leads to bias, resulting in the weakening of information from central pixels in HSI classification.

To address this issue, we incorporate the self-attention mechanism from the Transformer, which can reassess the importance of various attributes within the output of RNNs [47], [48]. Fig. 4 shows the detailed RT encoder.

Therefore, the process of a single layer in the RT encoder can be described as follows. First, RNN is used for both positional and feature embedding, as shown below

$$\boldsymbol{P}_m^{(i,j),l} = \text{RNN}\left(\boldsymbol{S}_m^{(i,j),l}\right) \tag{11}$$

where $l = 1, \ldots, L$ represents the lth layer of the RT encoder.

Subsequently, we apply layer normalization (LN) and a skip connection to normalize and enhance the features

$$\tilde{\boldsymbol{S}}_m^{(i,j),l} = \text{LN}\left(\gamma_m^l \boldsymbol{S}_m^{(i,j),l} + \delta_m^l \boldsymbol{P}_m^{(i,j),l}\right) \tag{12}$$

where $\gamma_m^l, \delta_m^l \in \mathbb{R}^1$ are learnable parameters in linear combination between two matrices, and $\gamma_m + \delta_m = 1$ generally.

Next, the PT is applied to $\tilde{\boldsymbol{S}}_m^{(i,j),l}$ to recalculate the attention weights

$$\tilde{\boldsymbol{T}}_m^{(i,j),l} = \text{PT}\left(\tilde{\boldsymbol{S}}_m^{(i,j),l}\right) \tag{13}$$

where PT represents the operations performed in the PT.

1) PT: The PT contains several operations especially the SMSA module. The embedded feature $\tilde{\boldsymbol{S}}_m^{(i,j),l}$ will first go through the SMSA as follows:

$$\boldsymbol{F}_m^{(i,j),l} = \text{SMSA}\left(\tilde{\boldsymbol{S}}_m^{(i,j),l}\right) \tag{14}$$

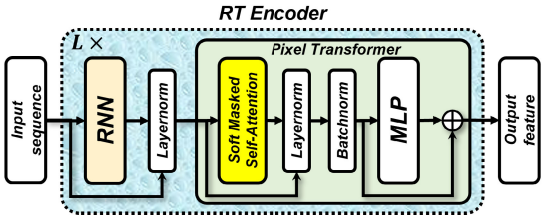


Fig. 4. Detailed structure of the RT encoder.

where $\boldsymbol{F}_m^{(i,j),l} \in \mathbb{R}^{p^2\times d}$ represents the output feature from the SMSA module. Details are prepared in Section II-E.

Then, we perform layer normalization and add a skip connection, similar to before

$$\boldsymbol{T}_m^{(i,j),l} = \mathrm{LN}\big(\boldsymbol{F}_m^{(i,j),l} + \tilde{\boldsymbol{S}}_m^{(i,j),l}\big). \tag{15}$$

Finally, after batch normalization (BN), the features are passed through a feedforward layer to generate the output of the lth layer

$$\tilde{\boldsymbol{T}}_m^{(i,j),l} = \mathrm{MLP}\big(\mathrm{BN}\big(\boldsymbol{T}_m^{(i,j),l}\big)\big) \tag{16}$$

where MLP represents multilayer perceptron (MLP).

Optionally, the lth layer's output is fed as input to the next layer of the RT encoder, which is calculated as follows:

$$\boldsymbol{S}_m^{(i,j),l+1} = \tilde{\boldsymbol{T}}_m^{(i,j),l} + \boldsymbol{T}_m^{(i,j),l}. \tag{17}$$

Consequently, for the final layer (L) of the RT encoder, we transform $\tilde{\boldsymbol{T}}_m^{(i,j),L}$ back into a new feature representation $\boldsymbol{F}_m^{(i,j)}$ through linear transformation

$$\boldsymbol{F}_m^{(i,j)} = \boldsymbol{W}\tilde{\boldsymbol{T}}_m^{(i,j),L} + \boldsymbol{b} \tag{18}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the relevant parameters. $\boldsymbol{F}_m^{(i,j)}$ will be utilized for subsequent procedures.

The RT encoder has a unique characteristic where each pixel is given more nuanced and varied weights to avoid losing important information and disregarding irrelevant information. In addition, the positional bias from the RNN can be effectively combined with Transformer.

D. FS Layer

The purpose of the FS layer is to create a center-focused representation of the feature $\boldsymbol{F}_m^{(i,j)}$. To achieve this, we allow the central element of $\boldsymbol{F}_m^{(i,j)}$ to have greater influence on the final feature representation, $\boldsymbol{y}_m^{(i,j)}$, for the mth scanning order. The intensified feature, $\boldsymbol{y}_m^{(i,j)} \in \mathbb{R}^{1\times d}$, is calculated as a weighted sum of all elements in $\boldsymbol{F}_m^{(i,j)}$

$$\boldsymbol{y}_m^{(i,j)} = \sum_{s=0}^{p^2-1} \boldsymbol{A}_m^{(i,j)}[s]\boldsymbol{F}_m^{(i,j)}[s] \tag{19}$$

where $s = 0, 1, \ldots, p^2-1$; $\boldsymbol{F}_m^{(i,j)}[s]$ represents the sth element of $\boldsymbol{F}_m^{(i,j)}$. Meanwhile, $\boldsymbol{A}_m^{(i,j)} \in \mathbb{R}^{p^2\times 1}$ serves as the attention for $\boldsymbol{F}_m^{(i,j)}$, and its element is calculated by

$$\boldsymbol{A}_m^{(i,j)}[s] = \mathrm{Softmax}\left\{\left(\boldsymbol{F}_m^{(i,j)}[s]\right)^\top \boldsymbol{F}_m^{(i,j)}\left[\frac{p^2-1}{2}\right]\right\}. \tag{20}$$

By doing this operation, we believe that a center-oriented feature can be generated to represent the cropped patch $\boldsymbol{X}^{(i,j)}$ in an appropriate manner. This is because the label of the patch is the same as the label of the central pixel, thus contributing to the model training.

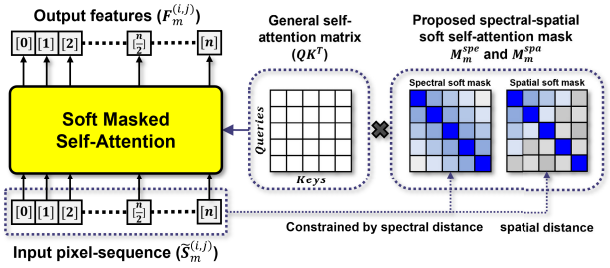


Fig. 5. Brief illustration of the proposed SMSA module. The original query–key attention matrix and the soft spectral and spatial mask are generated from the input tokens. The values of the two soft masks are constrained into [0, 1].

E. Spectral–Spatial-Based SMSA

The goal of HSI classification is to determine the semantic label of a central pixel in a patch of HSI data. These patches are created by randomly selecting a portion of the original HSI data and used as samples. However, the presence of interfering pixels with different labels within the patch can negatively impact the spectral–spatial features used for representation [42]. Hence, it is important to design an attention mask that focuses on relevant information from homogeneous pixels and disregards the effects of interfering pixels.

In the community, most methods attempt to design a hard attention mask that uses binary values of 0 and 1 to determine which features to keep or discard [43], [44], [45]. However, there are several drawbacks to this approach: 1) difficulty in precisely determining the placement of 0's and 1's; 2) the binary values of 0 and 1 can be too strict, leading to discarding of useful information or inclusion of negative information; 3) lack of consideration for the correlation between pixels; and 4) reliance on manual expertise. To address these concerns, we propose a soft attention mask based on the similarity between pixels within a cropped HSI patch, as illustrated in Fig. 5.

Specifically, in one cropped patch ($\boldsymbol{X}^{(i,j)} \in \mathbb{R}^{p\times p\times C}$), as demonstrated in (4), we calculate the pair distance between two pixels using the Euclidean distance

$$d^{\mathrm{spe}}(\boldsymbol{x}^{(i_1,j_1)}, \boldsymbol{x}^{(i_2,j_2)}) = \|\boldsymbol{x}^{(i_1,j_1)} - \boldsymbol{x}^{(i_2,j_2)}\|^2 \tag{21}$$

where $i - ((p-1)/2) \le i_1, i_2 \le i + ((p-1)/2)$; $j - ((p-1)/2) \le j_1, j_2 \le j + ((p-1)/2)$. $\|\cdot\|$ denotes the Euclidean norm. Subsequently, we compute a spectral distance matrix $\boldsymbol{D}^{\mathrm{spe}} \in \mathbb{R}^{p^2\times p^2}$, which is a symmetric matrix consisting of all pair distances $d^{\mathrm{spe}}(\boldsymbol{x}^{(i_1,j_1)}, \boldsymbol{x}^{(i_2,j_2)})$. Then, the values ($w^{\mathrm{spe}}$) of the spectral-based soft mask are obtained from the Gaussian function

$$w^{\mathrm{spe}}(\boldsymbol{x}^{(i_1,j_1)}, \boldsymbol{x}^{(i_2,j_2)}) = \begin{cases} 1, & \text{if } i_1 = i_2,\ j_1 = j_2 \\ \exp\left(-\dfrac{(d^{spe}(\boldsymbol{x}^{(i_1,j_1)}, \boldsymbol{x}^{(i_2,j_2)}))^2}{2\rho^2}\right), & \text{otherwise} \end{cases} \tag{22}$$

where ρ is the average pair distance in the matrix $\boldsymbol{D}^{\text{spe}}$. Longer spectral distances receive a lower weight, with all weights being between 0 and 1. The resulting spectral soft mask, $\boldsymbol{M}^{\text{spe}} \in \mathbb{R}^{p^2 \times p^2}$, is a feature matrix containing all the corresponding pair weights $w^{\text{spe}} \in \mathbb{R}^1$.

Successively, the spatial distance between two pixels is calculated based on their spatial coordinates [49], for instance

$$d^{\text{spa}}(\boldsymbol{x}^{(i_1,j_1)}, \boldsymbol{x}^{(i_2,j_2)}) = |i_1 - i_2| + |j_1 - j_2| \tag{23}$$

where $|\cdot|$ denotes absolute value. The max distance in spatial domain is fixed as follows:

$$d^{\text{spa}}(\max) = (p-1) \times 2. \tag{24}$$

Consequently, we define the spatial distance matrix $\boldsymbol{D}^{\text{spa}} \in \mathbb{R}^{p^2 \times p^2}$, which records the pair distance $d^{\text{spa}}(\boldsymbol{x}^{(i_1,j_1)}, \boldsymbol{x}^{(i_2,j_2)})$. The values ($w^{\text{spa}}$) of the spatial-based soft mask are calculated using the subtract function

$$w^{\text{spa}}(\boldsymbol{x}^{(i_1,j_1)}, \boldsymbol{x}^{(i_2,j_2)}) = \frac{d^{\text{spa}}(\max) - d^{\text{spa}}(\boldsymbol{x}^{(i_1,j_1)}, \boldsymbol{x}^{(i_2,j_2)})}{d^{\text{spa}}(\max)}. \tag{25}$$

The spatial soft mask, denoted as $\boldsymbol{M}^{\text{spa}} \in \mathbb{R}^{p^2 \times p^2}$, comprises all pair weights $w^{\text{spa}} \in \mathbb{R}^1$ corresponding to each pixel pair in the cropped patch.

Finally, the spectral and spatial soft masks ($\boldsymbol{M}^{\text{spe}}$ and $\boldsymbol{M}^{\text{spa}}$) are integrated into the self-attention layer, as detailed in Appendix B, to perform soft feature generation. The overall process of SMSA is summarized as follows:

$$\boldsymbol{F}_m^{(i,j)} = \text{Softmax}\left(\frac{\boldsymbol{Q}_m(\boldsymbol{K}_m)^T}{\sqrt{d_k}} \odot \boldsymbol{M}_m^{\text{spe}} \odot \boldsymbol{M}_m^{\text{spa}}\right)\boldsymbol{V}_m \tag{26}$$

where $\odot$ represents Hadamard product. $\boldsymbol{M}_m^{\text{spe}}$ and $\boldsymbol{M}_m^{\text{spa}}$ represent the spectral mask and spatial mask at the mth scanning order, respectively. This is because the soft masks are created from the original input, allowing for the accurate identification and reduction of interference from pixels.

In addition, SMSA can also be expanded into a multihead version, which maps the input into several heads' features with various $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$. The results from each head are then combined by concatenation

$$\boldsymbol{F}_m^{(i,j)} = \text{Concat}\left(\boldsymbol{F}_{m,1}^{(i,j)}, \boldsymbol{F}_{m,2}^{(i,j)}, \ldots, \boldsymbol{F}_{m,h}^{(i,j)}\right)\boldsymbol{W} \tag{27}$$

where h is the head number and $\boldsymbol{W}$ is the relevant parameters.

F. Multiscanning Feature FT

The integrated network can be implemented in two ways. The first approach involves processing each scanned sequence individually. The second approach proposes working with pairs of sequences. In the multiscanning strategy, there are four pairs of forward and backward scanning orders, such as order-1 and order-3. To benefit from the expansive reception fields and make the model more effective, it is suggested to feed these pairs into a bidirectional RNN (Bi-RNN) [46].

1) Scheme-1: The process of generating feature representation for an HSI patch begins with cropping the patch and scanning it into multiple pixel sequences using the multiscanning strategy. Each pixel sequence is treated as an individual and passed through the RT encoder to generate a feature representation $\boldsymbol{y}_m^{(i,j)}$ at the mth scanning order. The resulting multiple features are then concatenated to capture scanning order-based attention using the FT module.

The feature representation $\boldsymbol{Y}^{(i,j)} = [\boldsymbol{y}_{\text{cls}}^{(i,j)}, \ldots, \boldsymbol{y}_8^{(i,j)}]^\top \in \mathbb{R}^{(1+8)\times d}$ is created by concatenating the features generated by RT encoder for each of the eight scanning orders, along with an extra learnable class token $\boldsymbol{y}_{\text{cls}}^{(i,j)}$. The resulting feature vector is of dimension $(1+8) \times d$ and is used as input for the FT encoder, which is designed based on the general ViT encoder as described in [18]

$$\tilde{\boldsymbol{Y}}^{(i,j)} = \text{ViT}(\boldsymbol{Y}^{(i,j)}) \tag{28}$$

$$\boldsymbol{y}_{\text{cls}}^{(i,j)} = \sum_{m=1}^{8} \boldsymbol{E}^{(i,j)}[m]\tilde{\boldsymbol{Y}}^{(i,j)}[m] \tag{29}$$

where $\boldsymbol{E}^{(i,j)} \in \mathbb{R}^{8\times 1}$; m represents the mth element of $\tilde{\boldsymbol{Y}}^{(i,j)}$ or $\boldsymbol{E}^{(i,j)}$, respectively, relevant to the multiscanning orders. Each element of $\boldsymbol{E}^{(i,j)}$ is obtained by the dot product

$$\boldsymbol{E}^{(i,j)}[m] = \text{Softmax}\left\{(\tilde{\boldsymbol{Y}}^{(i,j)}[m])^\top \tilde{\boldsymbol{Y}}^{(i,j)}[0]\right\}. \tag{30}$$

The final fused feature $\boldsymbol{y}_{\text{cls}}^{(i,j)}$ is then passed through the subsequent decoder for classification. It is important to note that each scanning order is treated independently, and positional information is not incorporated in (28).

2) Scheme-2: In scheme-2, the multiscanning sequences are processed in pairs. The calculation process is as follows:

$$\overrightarrow{\boldsymbol{P}_m^{(i,j)}}[s] = \phi(\overrightarrow{\boldsymbol{W}_m}\boldsymbol{S}_m^{(i,j)}[s] + \overrightarrow{\boldsymbol{U}_m}\boldsymbol{P}_m^{(i,j)}[s-1] + \overrightarrow{\boldsymbol{b}_m}) \tag{31}$$

$$\overleftarrow{\boldsymbol{P}_m^{(i,j)}}[s] = \phi(\overleftarrow{\boldsymbol{W}_m}\boldsymbol{S}_m^{(i,j)}[s] + \overleftarrow{\boldsymbol{U}_m}\boldsymbol{P}_m^{(i,j)}[s+1] + \overleftarrow{\boldsymbol{b}_m}) \tag{32}$$

where $\overrightarrow{\boldsymbol{P}_m^{(i,j)}}[s]$ and $\overleftarrow{\boldsymbol{P}_m^{(i,j)}}[s]$ represent the forward and backward outputs at the sth-order pixel in pixel sequence; $\overrightarrow{\boldsymbol{W}_m}$, $\overleftarrow{\boldsymbol{W}_m}$, $\overrightarrow{\boldsymbol{U}_m}$, and $\overleftarrow{\boldsymbol{U}_m}$ represent the weight coefficient matrices in the forward and backward manners, respectively. Also, $\overrightarrow{\boldsymbol{b}_m}$ and $\overleftarrow{\boldsymbol{b}_m}$ are biases in the inverse manner.

As an example, $\overrightarrow{\boldsymbol{P}_m^{(i,j)}}[s]$ and $\overleftarrow{\boldsymbol{P}_m^{(i,j)}}[s]$ can be regarded as the outputs from $\boldsymbol{P}_1^{(i,j)}[s]$ and $\boldsymbol{P}_3^{(i,j)}[s]$, respectively. Subsequently, we concatenate the outputs from Bi-RNN of two inverse sequences by

$$\boldsymbol{P}_m^{(i,j)}[s] = \text{Concat}\left[\overrightarrow{\boldsymbol{P}_m^{(i,j)}}[s], \overleftarrow{\boldsymbol{P}_m^{(i,j)}}[s]\right]. \tag{33}$$

Hence, the new feature $\boldsymbol{Y}^{(i,j)} = [\boldsymbol{y}_{\text{cls}}^{(i,j)}, \ldots, \boldsymbol{y}_4^{(i,j)}]^\top \in \mathbb{R}^{(1+4)\times 2d}$. Similar to (28)–(30), it will be fed into FT for multiscanning feature fusion and then classification layer.

III. Experiments

In this section, the four well-known HSI datasets are described first. Then, the implementation details and comparison methods are introduced. Finally, extensive experiments are performed with an ablation analysis to evaluate the performance of the proposed method both quantitatively and qualitatively.

Algorithm 1 Pseudo-Procedure of Proposed Scheme-1

Input: Input the HSI data $\boldsymbol{X} \in \mathbb{R}^{H\times W\times C}$ and ground-truth $\boldsymbol{L} \in \mathbb{R}^{H\times W}$; patch size $= p$; training sample rate $= u\%$;
Initialization: Optimizer: Adam; learning rate = 0.01: criterion = CrossEntropy; epoch = 200; batch size = 100;
Other settings: All feature size = 64; encoder layers = 3;
Output: Predicted labels of the test dataset.
Procedure:
1. Create all sample patches from $\boldsymbol{X}$, and divide them into training dataset and test dataset .
2. Generate training loader and test loader.
Forward: for one cropped patch ($\boldsymbol{X}^{(i,j)} \in \mathbb{R}^{p\times p\times C}$),
3. Perform Multiscanning strategy on $\boldsymbol{X}^{(i,j)}$ to generate eight pixel-sequence $\boldsymbol{S}_m^{(i,j)}, m = 1, 2, 3, \ldots, 8$.
4. For each $\boldsymbol{S}_m^{(i,j)}$, perform RNN to generate $\boldsymbol{P}_m^{(i,j)}$.
5. For each $\boldsymbol{S}_m^{(i,j)}$, perform spectral-spatial-based soft masks to get $\boldsymbol{M}_m^{spe}$, and $\boldsymbol{M}_m^{spa}$.
6. Input $\boldsymbol{S}_m^{(i,j)}$ and $\boldsymbol{P}_m^{(i,j)}$ into RT encoder where the attention matrix is constrained by $\boldsymbol{M}_m^{spe}$, and $\boldsymbol{M}_m^{spa}$.
7. Get output features $\tilde{\boldsymbol{T}}_m^{(i,j)}$ at last layer of RT encoder
8. Do the linear transformation on $\tilde{\boldsymbol{T}}_m^{(i,j)}$ to get $\boldsymbol{F}_m^{(i,j)}$
9. Calculate attention $\boldsymbol{A}_m^{(i,j)}$ for $\boldsymbol{F}_m^{(i,j)}$, and obtain the final feature representation $\boldsymbol{y}_m^{(i,j)}$ for m-th scanning order.
10. Embed a extra learnable class token ($\boldsymbol{y}_{\text{cls}}^{(i,j)}$) for fusing all weighted scanning features as the total features for subsequent MLP and fully connected layer.
10. Use $Softmax$ function to obtain a vector $\boldsymbol{g}^{(i,j)}$ distributing the probability value on each label.
return $\boldsymbol{g}^{(i,j)}$
11. Use test dataset with the trained model to get predicted labels.

A. Description of Datasets

The four publicly available HSI datasets are illustrated in Figs. 6–9, along with their corresponding false-color images, ground truths, and legends. The datasets are obtained from the provided website[1] without any preprocessing.

The Indian Pines (IP) dataset is a mixed vegetation site with 145 × 145 pixels and 220 spectral bands. To remove water absorption bands, the number of spectral bands was reduced to 200, resulting in 16 different land-cover classes.

The Pavia University (PU) dataset consists of 610 × 340 pixels and initially has 115 bands. In the experiment, 12 of the noisy bands were removed, leaving 103 bands to be used. The dataset has nine different ground cover types.

The Salinas (SA) was captured by an airborne visible/infrared imaging spectrometer (AVIRIS) sensor over the SA valley and contains 204 bands with a resolution of 512 × 217 pixels. It includes 16 classes and has a relatively uniform distribution of ground objects.

The Houston (HU) 2013 dataset was utilized in the 2013 IEEE Geoscience and Remote Sensing Society (GRSS) data fusion contest.[2] It features 144 spectral bands with a size of 349 × 1905 pixels and includes 15 different land-cover classes.

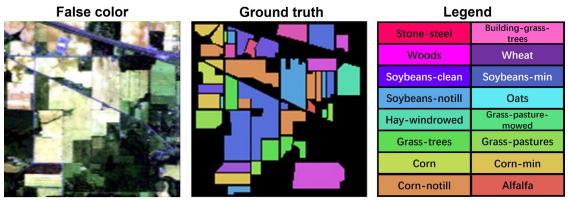


Fig. 6. IP: false-color image, ground truth, and legend.

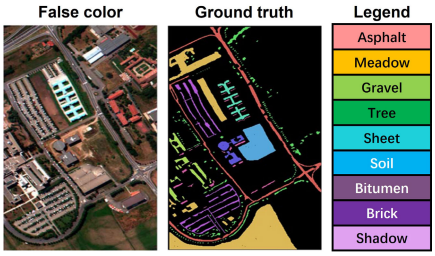


Fig. 7. PU: false-color image, ground truth, and legend.

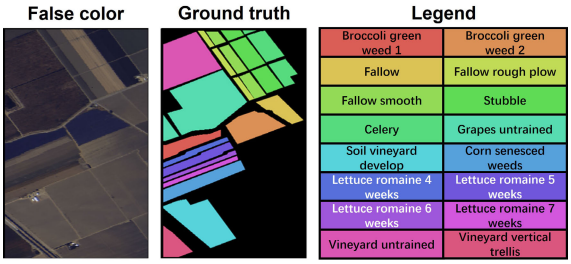


Fig. 8. SA: false-color image, ground truth, and legend.

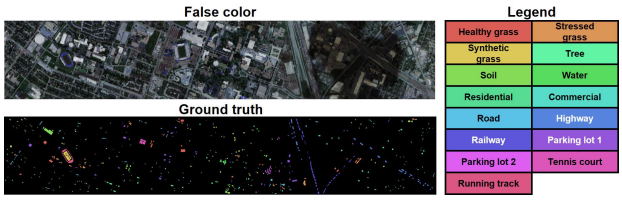


Fig. 9. HU 2013: false-color image, ground truth, and legend.

The labeled samples are separated into training and testing sets to assess the effectiveness and practicality of the proposed methods. Typically, 10% of the labeled pixels are designated as training samples, while the remaining 90% are used for testing. In addition, 5% of the training samples are selected as validation samples. The number of sample selections can be seen in Tables II–V.

B. Experimental Setting

1) General Setting: For our proposed method, the number of training epochs is set to 200, and the batch size is 100. The Adam optimizer and cross-entropy loss function are used for training. Moreover, we replace all RNN units with LSTM units in the experiments. The number of LSTM layers is set to 3, and the hidden size in LSTM is 64. The feature dimensions in the model, such as LSTM, Transformer, and fully connected layer, are also set to 64. The number of RT layers is set to 3. The hyperparameters in the Transformer will be discussed later. A dropout layer with a probability of 0.5 is added before the fully connected layer to prevent overfitting. The initial patch size is set as 7 × 7, 9 × 9, 11 × 11, and 9 × 9 for

[1]http://lesun.weebly.com/hyperspectral-data-set.html
[2]http://www.grss-ieee.org/community/technical-committees/data-fusion

TABLE II
LAND-COVER TYPES AND NUMBER OF PIXELS ON THE IP DATASET

Class		Number of Samples	
Type No.	Name	Training	Testing
1	Alfalfa	5	41
2	Corn-notill	143	1285
3	Corn-mintill	83	747
4	Corn	24	213
5	Grass-pasture	48	435
6	Grass-trees	73	657
7	Grass-pasture-mowed	3	25
8	Hay-windowed	48	430
9	Oats	2	18
10	Soybean-notill	97	875
11	Soybean-mintill	246	2209
12	Soybean-clean	59	534
13	Wheat	20	185
14	Woods	126	1139
15	Building-grass-trees	39	349
16	Stone-steal-towers	9	84
Total		1025	9224

TABLE III
LAND-COVER TYPES AND NUMBER OF PIXELS ON THE SA DATASET

Class		Number of Samples	
Type No.	Name	Training	Testing
1	Broccoli green weed 1	200	1808
2	Broccoli green weed 2	372	3353
3	Fallow	197	1778
4	Fallow rough plow	139	1254
5	Fallow smooth	268	2410
6	Stubble	395	3563
7	Celery	357	3221
8	Grapes untrained	1127	10143
9	Soil vineyard develop	620	5582
10	Corn senesced weeds	327	2950
11	Lettuce romaine 4 weeks	106	961
12	Lettuce romaine 5 weeks	192	1734
13	Lettuce romaine 6 weeks	91	824
14	Lettuce romaine 7 weeks	107	963
15	Vineyard untrained	726	6541
16	Vineyard vertical trellis	180	1626
Total		5404	48711

TABLE IV
LAND-COVER TYPES AND NUMBER OF PIXELS ON THE PU DATASET

Class		Number of Samples	
Type No.	Name	Training	Testing
1	Asphalt	663	5967
2	Meadows	1864	16784
3	Gravel	209	1889
4	Trees	306	2757
5	Metal sheets	134	1210
6	Bare soil	502	4526
7	Bitumen	133	1197
8	Bricks	368	3313
9	Shadow	94	852
Total		3860	38924

IP, PU, SA, and HU datasets, respectively. The experiments are conducted on the PyTorch 1.8 platform using a Geforce RTX 3060 GPU. The code for this work will be available at the GitHub repository[3] for reproducibility purposes.

2) Evaluation Indicators: The effectiveness of the proposed method, as well as that of other methods for comparison, is quantitatively analyzed through a evaluation of the classification performance. This evaluation is based on four key indicators: overall accuracy (OA), average accuracy (AA), the kappa coefficient (Kappa), and accuracy within each class.

[3]https://github.com/zhouweilian1904/

TABLE V
LAND-COVER TYPES AND NUMBER OF PIXELS ON THE HU 2013 DATASET

Class		Number of Samples	
Type No.	Name	Training	Testing
1	Healthy grass	126	1126
2	Stressed grass	126	1129
3	Synthetic grass	69	627
4	Tree	122	1120
5	Soil	125	1118
6	Water	32	292
7	Residential	125	1141
8	Commercial	127	1120
9	Road	124	1127
10	Highway	125	1104
11	Railway	124	1112
12	Parking lot 1	124	1110
13	Parking lot 2	46	422
14	Tennis court	45	385
15	Running track	40	594
Total		1502	13527

A higher value for each of these indicators indicates a more optimal classification outcome.

C. Brief Description of Compared Methods

The performance of the proposed idea was evaluated by comparing it with other methods across four categories: 1) Transformer only; 2) Transformer plus CNNs; 3) RNNs; and 4) Transformer plus RNNs. For this comparison, several state-of-the-art models were considered, including general ViT [18], SpeFormer [19], 1-D-CNN with Transformer (1DCT) [34], spatial-spectral transformer (SST) [35], spectral-spatial feature tokenization transformer (SSFTT) [38], self-attention-based transformer (SAT) [21], 3-D asymmetric neural architecture search (3-D-ANAS) [37], CasRNN [12], and multi-LSTM [15]. These models are all selected as patch-wise classifiers, and the parameter settings are consistent with their references. The comparison of results is based on the same ratio of training samples. The details of initial patch size (input size) for each method are described below.

1) General ViT refers to the implementation of the original ViT model. In this implementation, the feature dimension is set to 64, the number of heads is 2, and the number of depth layers is 2. Each pixel is considered as a single token. The initial patch size is set as 7×7 for all datasets.
2) SpeFormer is a modified Transformer-based HSI classifier that incorporates groupwise spectral embedding with cross-layer adaptive fusion (CAF) modules into the Transformer encoder. In this comparison, the patch-based CAF module was selected. The initial patch size is set as 7×7 for all datasets.
3) The 1DCT is a model that combines a 1-D-CNN and a Transformer. In this model, each pixel in a patch undergoes 1-D-CNN processing for feature embedding, and then, the Transformer is used to capture the self-attention weights among all pixels for further enhancement. The initial patch size is set as 25×25 for all datasets.
4) The SST model employs a well-designed CNN (VGG-16 [36]) to extract spatial features, a modified Transformer (featuring a dense connection) to capture sequential spectral relationships, and a MLP to perform

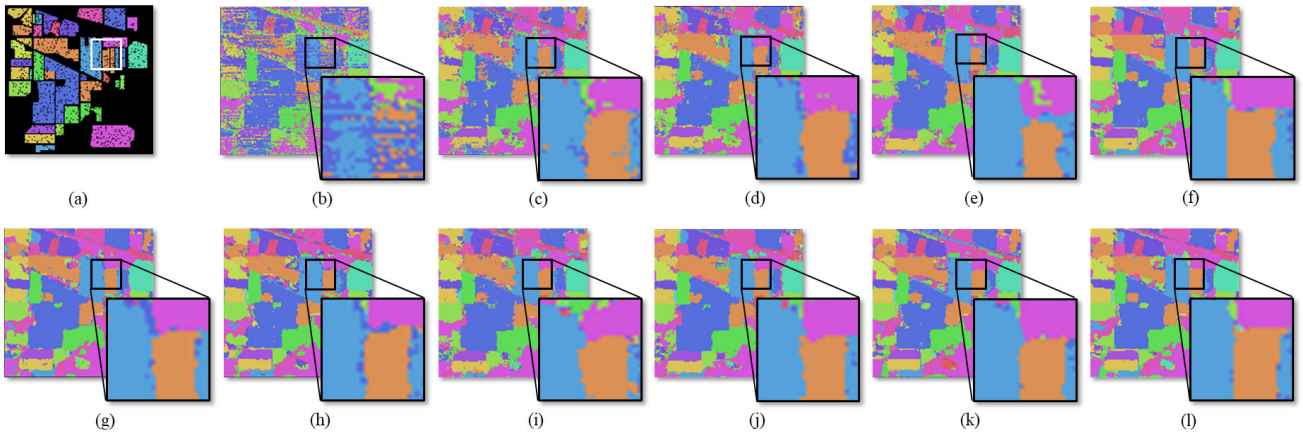


Fig. 10. Classification maps with different methods for the IP dataset. (a) Testing area. (b) General ViT (OA = 65.932%). (c) SpeFormer (OA = 86.190%). (d) 1DCT (OA = 91.361%). (e) SST (OA = 90.157%). (f) SSFTT (OA = 97.214%). (g) SAT (OA = 96.012%). (h) 3-D-ANAS (OA = 96.524%). (i) CasRNN (OA = 93.290%). (j) Multi-LSTM (OA = 95.902%). (k) Our scheme-1 (OA = 97.109%). (l) Our scheme-2 (OA = 97.751%).

TABLE VI

QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TERMS OF OA, AA, AND KAPPA AS WELL AS THE ACCURACIES FOR EACH CLASS ON THE IP DATASET. THE BEST ONE IS SHOWN IN BOLD

	Transformer only		Transformer + CNNs					RNNs		Transformer + RNNs	
	general ViT	SpeFormer	1DCT	SST	SSFTT	SAT	3D-ANAS	CasRNN	Multi-LSTM	scheme 1	scheme 2
1	65.19	65.62	75.81	88.17	95.81	94.33	94.58	90.70	95.81	96.72	**96.81**
2	67.61	85.41	93.39	92.38	98.13	96.32	98.37	94.01	98.28	98.28	**98.48**
3	64.02	79.28	87.21	89.69	94.25	94.10	92.78	89.89	94.37	94.19	**94.38**
4	61.66	77.50	87.79	87.94	93.58	91.22	91.59	**96.52**	95.21	94.52	95.22
5	65.01	90.03	90.92	90.11	94.69	97.33	96.63	91.19	96.03	97.64	**97.66**
6	71.33	96.15	97.74	96.29	**100**	99.71	99.58	97.22	**100**	**100**	**100**
7	61.28	37.61	69.81	86.91	94.12	95.21	92.83	92.60	94.01	95.03	**95.29**
8	69.10	95.44	96.52	94.22	**100**	99.70	99.44	99.41	99.38	**100**	**100**
9	62.91	45.32	53.27	87.08	93.37	93.23	92.96	99.08	**100**	98.80	**100**
10	67.31	84.66	89.13	89.92	95.67	95.81	95.36	90.06	95.18	94.98	**95.88**
11	66.28	85.81	93.66	91.20	**97.66**	97.00	97.35	94.12	96.62	96.05	96.89
12	64.06	74.79	88.54	91.38	97.37	96.09	96.56	94.30	97.21	97.33	**97.95**
13	63.33	98.01	98.81	92.01	97.56	100	98.29	98.77	99.14	**100**	**100**
14	67.14	95.77	95.42	92.09	97.78	99.04	98.49	97.63	98.66	**100**	**100**
15	66.28	70.07	74.44	91.21	**96.21**	87.42	94.75	81.30	89.29	91.83	92.55
16	65.03	86.51	95.20	90.88	96.73	97.77	95.74	95.41	96.70	97.60	**98.51**
OA	65.932	86.190	91.361	90.157	97.214	96.012	96.524	93.290	95.902	97.109	**97.751**
AA	63.218	79.249	87.732	90.718	96.434	95.893	95.878	93.889	96.618	97.060	**97.476**
Kappa	0.601	0.842	0.902	0.903	0.969	0.955	0.961	0.924	0.953	0.969	**0.973**

the final classification task. The initial patch size is set as 33 × 33 for all datasets.

5) SSFTT first applies 3-D-CNN and 2-D-CNN on the HSI. Then, it introduces a Gaussian-weighted feature tokenizer for feature transformation, and the transformed features are fed into the Transformer encoder module for feature learning. The initial patch size is set as 13 × 13 for IP, PU, and SA datasets, and 9 × 9 for HU dataset.
6) The SAT model incorporates a spectral attention block using CBAM [22] prior to the spatial Transformer. The initial patch size is set as 16 × 16 for all datasets.
7) The 3-D-ANAS reevaluates the search space of previous HSI classification neural architecture search (NAS) methods and introduces a novel hybrid search space that incorporates 3-D-CNN, 2-D spatial CNN, and 2-D spectral CNN. The Transformer module is added to enhance local region-focused features learned by the CNNs with global information. The initial patch size is set as 14 × 14 for IP and HU datasets, and 24 × 24 for PU and SA datasets.
8) In the CasRNN model, a cropped HSI patch is processed CNNs and RNNs both alongside spectral domain. Two kinds of RNNs are deployed for eliminating spectral redundancy and enhancing nonadjacent information. The initial patch size is set as 27 × 27 for all datasets.
9) The multi-LSTM, our previous work [15], employs a multiscanning strategy to scan the HSI patch into several pixel sequences with different scanning orders. By using eight different scanning orders, the local patch is complemented with correlative dependence. The concatenated features from all scanning orders are then fed into an RNN for additional complementarity. The initial patch size is set as 5 × 5 for IP and PU datasets, and 7 × 7 for SA and HU datasets.

D. Quantitative Results and Classification Maps

The quantitative classification results, including OA, AA, Kappa, and accuracy for each class, are presented in Tables VI–IX for IP, PU, SA, and HU datasets, respectively. In addition, the corresponding classification maps

TABLE VII

QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TERMS OF OA, AA, AND KAPPA AS WELL AS THE ACCURACIES FOR EACH CLASS ON THE PU DATASET. THE BEST ONE IS SHOWN IN BOLD

	Transformer only		Transformer + CNNs					RNNs		Transformer + RNNs	
	general ViT	SpeFormer	1DCT	SST	SSFTT	SAT	3D-ANAS	CasRNN	Multi-LSTM	scheme 1	scheme 2
1	76.08	91.02	96.45	91.26	97.90	92.17	98.77	92.71	98.42	98.32	**98.91**
2	75.18	88.91	95.08	90.81	97.33	93.07	97.01	91.40	97.88	98.19	**98.81**
3	76.43	90.87	92.09	93.01	**99.27**	94.59	96.39	87.39	95.69	98.46	98.99
4	73.14	99.01	97.77	96.12	**100**	98.17	**100**	96.01	**100**	**100**	**100**
5	80.03	96.14	**100**	96.03	**100**	98.06	**100**	95.61	**100**	**100**	**100**
6	78.10	81.71	97.71	95.21	**100**	96.68	**100**	95.05	99.70	99.91	**100**
7	79.19	84.59	97.68	97.28	**100**	97.13	99.59	90.78	98.64	**100**	**100**
8	80.14	86.44	95.10	91.99	97.15	93.87	**98.21**	92.77	97.49	97.03	97.77
9	76.05	96.11	97.12	94.98	**100**	96.07	**100**	97.01	**100**	**100**	**100**
OA	76.117	89.405	96.351	94.111	99.171	95.261	98.561	91.748	98.877	99.022	**99.253**
AA	74.008	90.533	96.555	94.076	99.072	95.003	98.886	93.192	98.646	99.101	**99.387**
Kappa	0.731	0.839	0.940	0.936	0.990	0.941	0.984	0.834	0.976	0.990	**0.991**

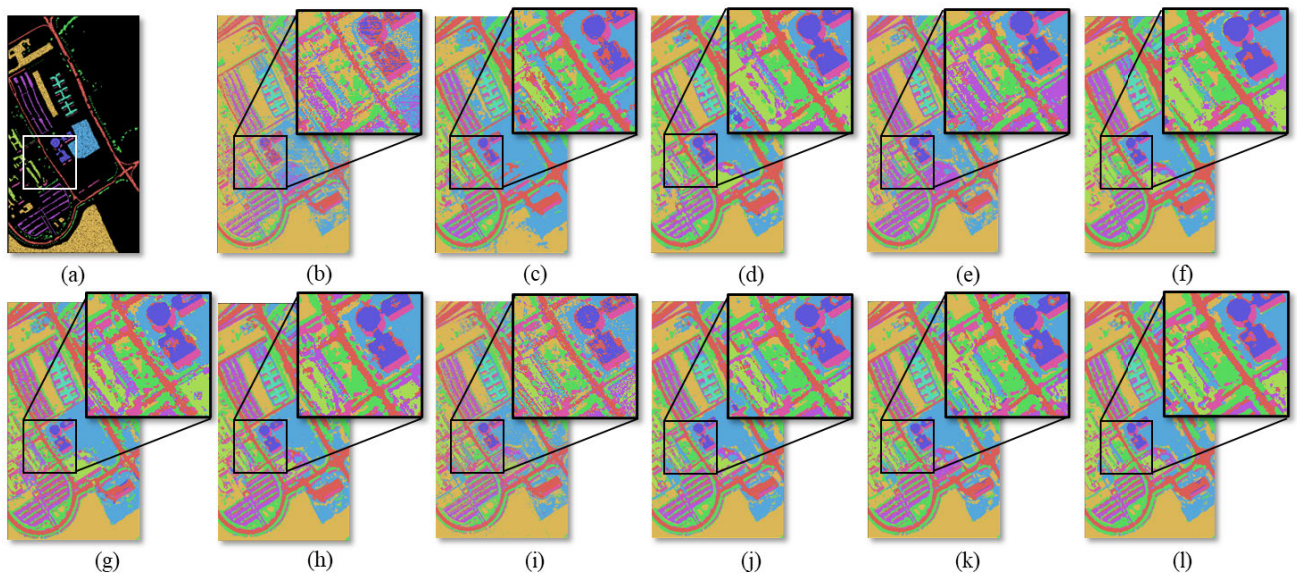


Fig. 11. Classification maps with different methods for the PU dataset. (a) Testing area. (b) General ViT (OA = 76.117%). (c) SpeFormer (OA = 89.405%). (d) 1DCT (OA = 96.351%). (e) SST (OA = 94.111%). (f) SSFTT (OA = 99.171%). (g) SAT (OA = 95.261%). (h) 3-D-ANAS (OA = 98.561%). (i) CasRNN (OA = 91.748%). (j) Multi-LSTM (OA = 98.877%). (k) Our scheme-1 (OA = 99.022%). (l) Our scheme-2 (OA = 99.253%).

demonstrating the training and testing samples are displayed in Figs. 10–13.

It is evident that the methods that solely employ the Transformer for HSI classification do not produce satisfactory results. The results obtained from the implementation of the general ViT model show poor performance in the overall classification maps. On the other hand, the SpeFormer method results in a significant improvement when compared with the general ViT model. However, the SpeFormer method still results in misclassifications that appear as noise-like patterns. Upon examination of the classification maps, it can be observed that certain land-cover boundaries are irregular, and some homogeneous regions appear to be not smooth, as indicated by the boxes in the classification maps.

On the other hand, some methods adopt a combination of CNNs and Transformer for HSI classification. In the case of 1DCT, the 1-D-CNN primarily influences the spectral features, leading to subpar classification results. SST employs the VGG-16 architecture on the HSI data prior to the Transformer encoder, integrating spectral information in the process. The SAT approach includes the CBAM attention block for spectral attention before the spatial Transformer. Although these methods demonstrate improved performance, they still exhibit limitations. Recently, SSFTT and 3-D-ANAS have combined 3-D-CNN, 2-D-CNN, and Transformer to achieve comparable results, particularly in terms of smooth regions and clear boundaries. However, it should be noted that the utilization of deep 3-D-CNN and 2-D-CNN leads to a higher computational burden and prolonged processing time.

The RNN-based methodologies of CasRNN and multi-LSTM present an alternate viewpoint in tackling the classification task in HSI. The CasRNN methodology integrates CNNs and RNNs, with the latter responsible for enhancing spectral features. It accomplishes this by first dividing the HSI along the spectral domain into smaller subimages, thus focusing solely on spectral features. The result of this process is a refined feature representation that eliminates redundancies and integrates nonadjacent information for the purpose of classification. Despite the absence of shuffling in the spatial domain through the use of CNNs, the classification results tend to exhibit an irregular spatial distribution. On the other hand, multi-LSTM employs a multiscanning approach to scan HSI patches into multiple complementary pixel sequences, thereby demonstrating the superiority of diverse ordering for

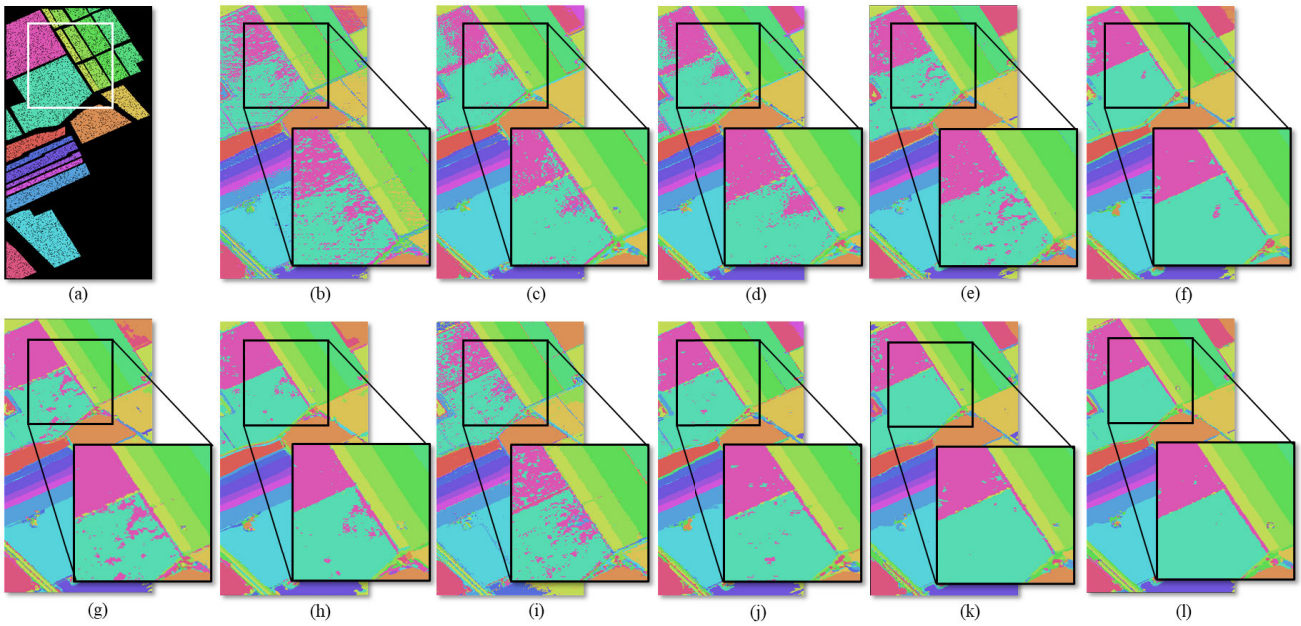


Fig. 12. Classification maps with different methods for the SA dataset. (a) Testing area. (b) General ViT (OA = 79.773%). (c) SpeFormer (OA = 92.440%). (d) 1DCT (OA = 93.215%). (e) SST (OA = 95.884%). (f) SSFTT (OA = 98.225%). (g) SAT (OA = 96.159%). (h) 3-D-ANAS (OA = 97.450%). (i) CasRNN (OA = 91.701%). (j) Multi-LSTM (OA = 97.351%). (k) Our scheme-1 (OA = 98.323%). (l) Our scheme-2 (OA = 98.723%).

TABLE VIII

QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TERMS OF OA, AA, AND KAPPA AS WELL AS THE ACCURACIES FOR EACH CLASS ON THE SA DATASET. THE BEST ONE IS SHOWN IN BOLD

	Transformer only		Transformer + CNNs					RNNs		Transformer + RNNs	
	general ViT	SpeFormer	1DCT	SST	SSFTT	SAT	3D-ANAS	CasRNN	Multi-LSTM	scheme 1	scheme 2
1	82.17	94.31	95.31	98.29	**100**	96.79	98.07	92.02	**100**	**100**	**100**
2	84.13	96.06	97.48	**100**	**100**	**100**	**100**	94.12	**100**	**100**	**100**
3	80.49	94.87	95.34	99.61	**100**	99.79	**100**	97.05	98.69	**100**	**100**
4	81.64	93.16	94.93	96.32	**100**	99.28	97.31	91.06	99.23	**100**	**100**
5	83.16	95.48	96.48	97.89	**100**	**100**	98.57	92.33	98.03	**100**	**100**
6	83.05	95.17	96.13	98.01	**100**	98.80	**100**	94.12	**100**	**100**	**100**
7	80.17	92.06	93.07	96.81	98.21	99.12	94.08	93.16	98.33	**100**	**100**
8	75.18	87.19	88.43	92.41	94.62	91.10	91.29	83.13	93.02	94.88	**97.02**
9	81.08	93.16	94.03	95.99	99.78	95.89	**100**	93.12	98.96	**100**	**100**
10	75.19	89.17	91.99	95.09	96.94	94.02	95.19	91.59	96.81	97.53	**98.76**
11	87.06	90.94	91.59	96.23	**98.72**	96.00	97.01	91.30	96.33	97.99	98.59
12	80.61	92.47	93.13	95.01	97.44	94.12	96.19	91.18	96.03	98.02	**99.01**
13	81.49	93.36	94.19	95.99	99.05	95.99	97.99	93.75	99.13	**100**	**100**
14	80.17	92.07	93.49	96.25	98.35	95.34	98.03	91.52	97.01	97.44	**98.59**
15	71.12	88.19	89.01	92.23	**94.35**	89.54	90.81	84.99	91.24	91.23	94.01
16	72.03	86.99	88.17	92.67	93.28	90.08	92.19	86.94	92.22	93.12	**94.18**
OA	79.773	92.440	93.215	95.884	98.225	96.159	97.450	91.701	97.351	98.323	**98.723**
AA	76.995	91.112	93.487	96.175	98.171	95.991	97.201	91.337	97.189	98.138	**98.760**
Kappa	0.751	0.902	0.923	0.958	0.980	0.957	0.973	0.910	0.970	0.981	**0.985**

these sequences in making RNNs more discriminative. This approach leads to significantly improved results compared with CasRNN, including clearer land-cover boundaries and a reduction in noise-like misclassifications.

Our study presents a multiscanning-based RT model that leverages the strengths of both RNNs and Transformers for HSI classification. By using a multiscanning strategy, the HSI patch is transformed into several pixel sequences, which are then processed by an RT encoder equipped with a spectral–spatial-based soft mask for feature generation and selection. The objective of this approach is to mitigate the negative impact of interfering pixels and improve the accuracy of land-cover classifications, particularly at the boundaries between classes. The results of our method are compared with other methods and demonstrate improved performance, especially in scheme-2 (Bi-RNN with Transformer). The classification maps exhibit smoother regions and clearer boundaries between land-cover classes.

Compared with SSFTT and 3-D-ANAS, our proposed method incorporates multiple scanning orders and soft masks to address the diverse perspectives of an HSI patch, which can be viewed as a data augmentation technique to enhance the discriminative ability of the model. Consequently, our approach demonstrates superior results compared with those of SSFTT and 3-D-ANAS. Moreover, our method enhances the previous work of multi-LSTM by seamlessly integrating RNN and Transformer and addressing the limitations that were observed in multi-LSTM, particularly in regards to its performance in larger initial patch sizes and the feasibility of incorporating attention mechanisms.

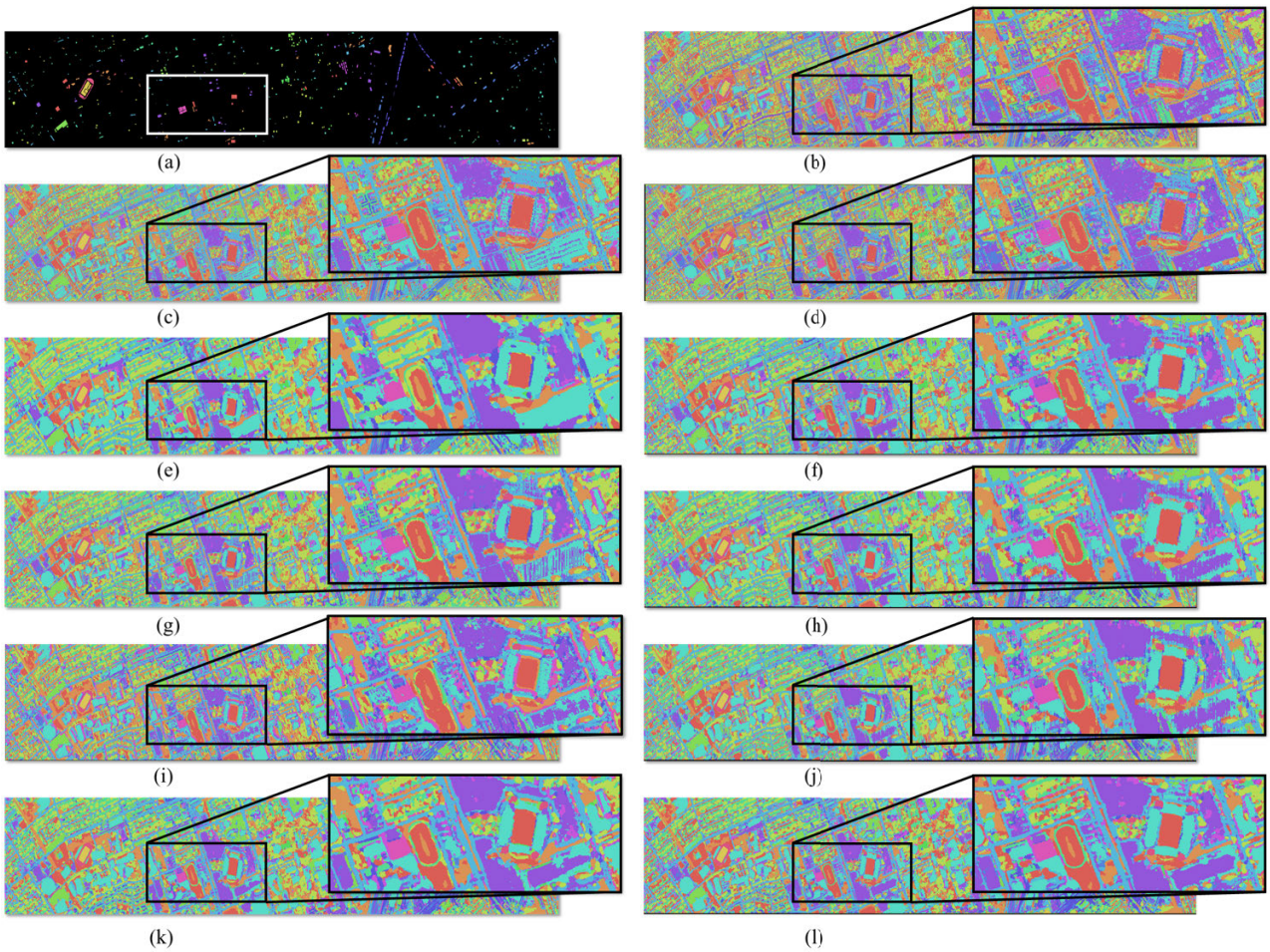


Fig. 13. Classification maps with different methods for the HU 2013 dataset. (a) Testing area. (b) General ViT (OA = 80.816%). (c) SpeFormer (OA = 90.992%). (d) 1DCT (OA = 94.795%). (e) SST (OA = 96.599%). (f) SSFTT (OA = 99.208%). (g) SAT (OA = 98.483%). (h) 3-D-ANAS (OA = 98.816%). (i) CasRNN (OA = 93.001%). (j) Multi-LSTM (OA = 98.343%). (k) Our scheme-1 (OA = 99.031%). (l) Our scheme-2 (OA = 99.441%).

TABLE IX

QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TERMS OF OA, AA, AND KAPPA AS WELL AS THE ACCURACIES FOR EACH CLASS ON THE HU 2013 DATASET. THE BEST ONE IS SHOWN IN BOLD

	Transformer only		Transformer + CNNs					RNNs		Transformer + RNNs	
	general ViT	SpeFormer	1DCT	SST	SSFTT	SAT	3D-ANAS	CasRNN	Multi-LSTM	scheme 1	scheme 2
1	92.41	93.40	98.59	98.54	99.59	99.01	99.33	93.57	99.37	99.69	**99.78**
2	96.11	94.91	98.91	97.45	**100**	99.58	99.38	91.43	99.43	99.64	99.65
3	75.18	95.92	**100**	**100**	**100**	**100**	99.51	95.01	99.77	**100**	**100**
4	94.89	94.17	99.42	99.33	99.62	99.43	99.41	94.38	99.91	99.75	**100**
5	94.60	94.89	98.18	95.27	**99.64**	99.37	98.17	90.12	98.31	99.18	99.61
6	91.52	94.92	99.79	98.61	**100**	**100**	97.54	91.54	97.04	99.42	**100**
7	68.28	91.48	95.61	98.19	99.47	99.06	98.36	91.76	97.16	98.47	**99.72**
8	71.79	89.66	92.94	97.74	**99.23**	97.51	97.59	96.52	96.71	98.71	98.66
9	73.74	80.84	87.76	95.86	98.60	95.89	98.21	91.01	97.39	98.16	**98.83**
10	65.91	86.83	91.06	99.11	99.11	98.73	98.85	93.07	97.45	99.56	**99.57**
11	73.48	86.57	93.12	96.40	98.48	98.17	97.85	90.81	98.56	99.04	**99.42**
12	59.22	83.94	89.08	98.58	99.06	95.01	98.37	94.38	95.64	98.53	**99.08**
13	51.59	75.96	79.24	96.13	98.74	99.38	98.73	90.22	98.04	98.77	**99.53**
14	89.01	95.23	99.51	**100**	**100**	**100**	99.31	95.03	99.30	**100**	**100**
15	96.63	94.47	99.79	**100**	**100**	99.98	99.50	94.99	99.81	**100**	**100**
OA	80.816	90.992	94.795	96.599	99.208	98.483	98.816	93.001	98.343	99.031	**99.441**
AA	79.579	90.212	94.866	97.080	99.436	98.759	98.181	92.922	98.259	99.261	**99.590**
Kappa	0.777	0.903	0.944	0.963	0.992	0.984	0.987	0.920	0.978	0.989	**0.994**

In addition, the proposed scheme-2, which integrates Bi-RNN and Transformer, represents a more compact and pragmatic solution when compared with scheme-1. Our analysis demonstrates that the use of Bi-RNN enables the simultaneous processing of two inverse sequences and expands the receptive field, leading to the generation of more discriminative and informative features.

The accuracy of several classes in the classification results is found to be suboptimal for all the methods under consideration, such as class “15” in the IP dataset and class

TABLE X
EFFECTS OF PARAMETER, γ AND δ, IN POSITIONAL EMBEDDING WITH MULTISCANNING STRATEGY

	Datasets							
	Indian Pines		Pavia University		Salinas		Houston 2013	
Scanning	δ	γ	δ	γ	δ	γ	δ	γ
1	0.6665	0.3335	0.7978	0.2022	0.5766	0.4234	0.5549	0.4451
2	0.8197	0.1803	0.6908	0.3092	0.8484	0.1516	0.5278	0.4722
3	0.8096	0.1904	0.7107	0.2893	0.7957	0.2043	0.6158	0.3842
4	0.7172	0.2828	0.6102	0.3898	0.8695	0.1305	0.6167	0.3833
5	0.5516	0.4484	0.7470	0.2530	0.8417	0.1583	0.5197	0.4803
6	0.6158	0.3842	0.6183	0.3817	0.8299	0.1701	0.5158	0.4842
7	0.5941	0.4059	0.7710	0.2290	0.8063	0.1937	0.6599	0.3401
8	0.7625	0.2375	0.9766	0.0234	0.8244	0.1756	0.5441	0.4559

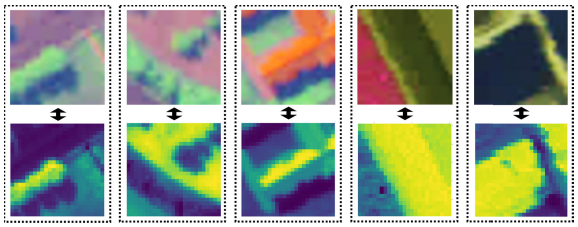


Fig. 14. Examples of attention maps in the PT. The upper one shows the false color of cropped HSI patch. The below one is the corresponding attention maps $\boldsymbol{A}'^{(i,j)}$.

"15" in the SA dataset. Analysis of the confusion matrix revealed that most of the misclassifications were predicted as unlabeled data or other similar classes, which may be due to intrinsic problems in HSI, such as the presence of the same material with different spectra or different materials with similar spectra, as previously reported in the literature [50]. Furthermore, the imbalance in the training samples selected randomly from each class could lead to a larger intraclass variance and a smaller between-class distance, making it challenging for the models to generalize. Among the methods, SSFTT showed the best results on these two classes, which could be attributed to the use of a Gaussian-weighted feature tokenizer and 3-D-CNNs that transformed the spatial–spectral features into more separable semantic features on the samples. The optimization of spatial–spectral features through the use of RNNs and Transformer models to enhance specific class accuracy remains an area of ongoing research. This presents an opportunity for further investigation and exploration in the field.

E. Other Analyses

1) Balance Weight in Positional Embedding, γ and δ: In the positional encoding, many studies have utilized a direct addition of positional information to the original input with the assumption that $\gamma = \delta$, as demonstrated in Appendix B. However, a casual summation of these values may not be appropriate. Hence, our investigation into the optimal values of γ and δ with the use of a multiscanning strategy was conducted to demonstrate their impact on model performance, as shown in Table X.

Obviously, there is a tendency for $\delta_m > \gamma_m$ primarily, where δ_m serves as the weight for the positional features in the mth scanning order. It should be noted that the outputs of the RNN are used as the positional features. The results indicate that the positional features tend to play a dominant role in the input features for the PT, suggesting that the RNN is capable of encoding more discriminative features than the original inputs.

Therefore, our approach differs from others, as it utilizes dynamic weights γ and δ to enhance the informativeness of the inputs for the PT. Our findings in this regard could potentially contribute to the advancement of research on positional embedding.

2) Attention Map Visualization: In the PT, the attention sequence $\boldsymbol{A}^{(i,j)} \in \mathbb{R}^{p^2 \times 1}$ can be obtained through the calculation described in (20). This sequence records the dot-product value between each pixel and the central pixel, and it can be transformed into an attention map $\boldsymbol{A}'^{(i,j)}$ for visualization purposes. This attention map has a size of $\mathbb{R}^{p \times p}$. Some representative attention maps are illustrated in Fig. 14.

The attention maps visually demonstrate the effectiveness and superiority of the designed PT module. The attention maps display a finer appearance, including details, such as edges, object outlines, and textural structures. This observation supports the efficacy of the module from a visual perspective.

This observation highlights the capability of the PT in capturing important spectral–spatial information from the attention areas, which can distinguish between homogeneous and interfering pixels in HSI cubes. This enhances the discriminative power of the extracted features for accurate HSI classification.

3) Effects of Initial Patch Size for Performance: In this section, the impact of the initial patch size on the classification performance is explored. A comprehensive range of patch sizes, ranging from 3 to 23, was evaluated empirically. The results, in terms of OA, for the various patch sizes on four datasets are presented in Table XI.

Within an optimal range, increasing patch size improves model performance by incorporating more spatial information. However, larger patch sizes can negatively impact results by introducing interfering pixels. The IP dataset, characterized by small spatial size and tight land-cover distributions, cannot support larger patch sizes, whereas the SA dataset, with a larger size and uniform distribution of ground objects, can accommodate larger patch sizes. The PU and HU datasets, both with large spatial size and complex land-cover distributions, exhibit optimal performance at patch sizes of 9×9, while the SA and IP dataset performs best with a patch size of 11×11 and 7×7, respectively, as determined from our results.

As previously noted, the pure RNN-based approach, exemplified by multi-LSTM, has been observed to encounter difficulties in processing larger patches. To investigate this effect, we present an overview of the accuracies obtained using different patch sizes. A comparison between the performance of multi-LSTM and our proposed model is depicted in Fig. 15. Notably, multi-LSTM demonstrates a more rapid decline in accuracy beyond its optimal patch size, while our model exhibits consistent accuracy despite slight degradation.

In addition, it is noteworthy that the relatively stable results even with larger patch sizes can be attributed to the implementation of the soft attention masks in our proposed method. These attention masks effectively eliminate the negative influence of interfering pixels by assigning them weights, thus ensuring stability in performance.

TABLE XI

IMPACT OF DIFFERENT PATCH SIZES FOR THE OA (%) ON FOUR DATASETS. THE BEST ONE IS HIGHLIGHTED

	Datasets			
Patch size	Indian Pines	Pavia University	Salinas	Houston 2013
3×3	96.598	98.216	96.991	98.123
5×5	97.225	98.881	97.332	98.335
7×7	**97.751**	98.001	98.010	99.013
9×9	97.558	**99.253**	98.219	**99.441**
11×11	97.501	99.159	**98.723**	99.213
13×13	97.418	99.021	98.661	99.111
15×15	97.222	98.885	98.438	98.863
17×17	97.011	98.369	98.221	98.669
19×19	96.881	98.042	98.002	98.129
21×21	96.582	97.819	97.699	98.002
23×23	96.318	97.665	97.331	97.863

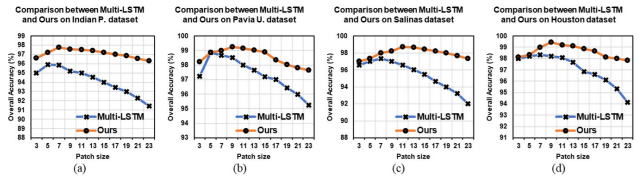


Fig. 15. Performance comparison between multi-LSTM and our proposed method for various patch sizes. (a) IP dataset. (b) PU dataset. (c) SA dataset. (d) HU 2013 dataset.

4) Effects of the Number of Training Samples: In this experiment, we aim to investigate the effect of the number of training samples on the performance of each method. The only parameter that is varied in this experiment is the number of selected training samples, while all other parameters are kept constant. The training sets for the analysis are comprised of randomly selected 1%, 3%, 5%, 10%, 15%, 20%, and 25% of the labeled samples from each class.

The results of the OA of various methods for different sizes of training sets are depicted in Fig. 16 for four datasets. The horizontal axis represents the percentage of selected training samples per class, and the vertical axis indicates the OA.

From Fig. 16, it is observed that the OA of each method increases with the increase in the number of training samples, approaching a saturation point of nearly 100% for several methods. Our proposed method demonstrates good performance even with a small number of samples. It is noted that the performance of other methods is found to be less accurate when the proportion of samples was 10%. The SSFTT method, however, demonstrated comparable or even superior performance compared to our proposed method. In addition, it is evident that the increase in the proportion of training samples beyond 5% does not result in significant changes.

5) Analysis on Multiheads in PT and FT: In order to examine the impact of the number of heads on the performance of our multiheads self-attention, additional experiments were conducted. The number of heads was varied as a crucial hyperparameter in both the PT and FT settings. The results of these experiments, in terms of OA, are displayed in Fig. 17. It is worth noting that there are several hyperparameters in the Transformer setting that can be altered, such as the feature dimension and layer depth. We fixed the other settings to a commonly used configuration, with a depth of 1, all dimensions equal to 64, and a dropout rate of 0.5.

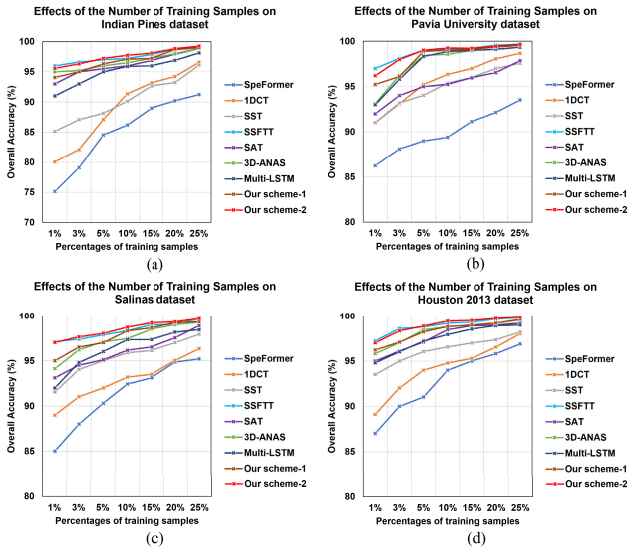


Fig. 16. OA of different models with different training samples percentages. (a) IP dataset. (b) PU dataset. (c) SA dataset. (d) HU 2013 dataset.

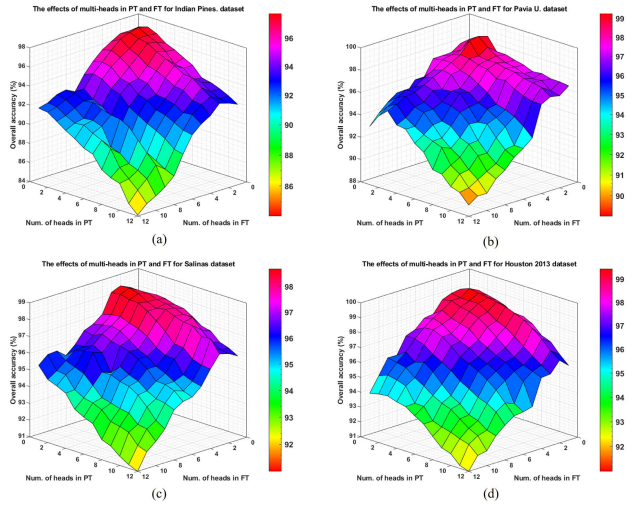


Fig. 17. Overall accuracies effected by the multiheads self-attention in Transformer settings (i.e., PT and FT). (a) IP dataset. (b) PU dataset. (c) SA dataset. (d) HU 2013 dataset. The x-axis represents number of heads in PT, and the y-axis denotes the number of heads in FT. The z-axis records the OA.

Based on the results presented in Fig. 17, it can be observed that there is a limited improvement in the OA, as the number of heads increases. The optimal number of heads for each dataset varies, with the best results being obtained for IP with PT heads $= 2$ and FT heads $= 2$, PU with PT heads $= 2$ and FT heads $= 3$, SA with PT heads $= 3$ and FT heads $= 2$, and HU with PT heads $= 2$ and FT heads $= 2$. It is worth noting that the model is more sensitive to the number of heads in the PT. The appropriate selection of the number of heads in the multihead self-attention mechanism is, therefore, of paramount importance in achieving optimal results.

6) Analysis on Layer Depth in FT: In order to investigate the impact of layer depth on the performance of the Transformer-based model, additional experiments were conducted with the other settings fixed (i.e., all dimensions set to 64, dropout set to 0.5, and the number of multiheads determined based on the results from the previous analysis).

Due to the incorporation of the PT into the proposed RT encoder layer, the present discussion focuses on the effects

TABLE XII
EFFECTS OF LAYER DEPTH IN THE PROPOSED FT

	Datasets			
Layer Depth	Indian Pines	Pavia University	Salinas	Houston 2013
1	96.861	98.723	98.216	98.776
2	97.113	**99.253**	**98.723**	99.123
3	**97.751**	98.884	98.334	**99.441**
4	97.126	97.221	98.231	98.593
5	96.773	97.029	98.005	98.228
6	96.229	96.793	97.591	97.861

of layer depth in the FT. The results of the experiment are documented in Table XII. Our findings indicate that the optimal FT depths for the four datasets are 3, 2, 2, and 3, respectively. It is evident that when the FT depth is 1, the model experiences underfitting for the IP, SA, and HU datasets. On the other hand, a deeper layer may result in overfitting of the performance. Although deep layers are commonly employed in training models for various tasks, it is essential to carefully consider and select the appropriate hyperparameters in the Transformer model, including the number of layers and the number of multiheads.

F. Ablation Study

1) Exploring the Capacity With Multiscannings: In light of the improved performance observed through the utilization of a multiscanning strategy, it is natural to pose two important questions: 1) what is the capacity of the multiscanning process? and 2) which scanning is the most significant? To address these questions, we conduct experiments focusing on three key points:

1) the impact of a single scanning on OA;
2) the saturation effect of utilizing multiple scannings;
3) the relative importance of each scanning in the analysis of an HSI patch.

For the first point, we conducted experiments by deploying one single scanning for training. The results, depicted in Fig. 18, indicate that each single scanning yields a similar OA. This can likely be attributed to the limited generative capacity of utilizing a single scanning. In addition, the unsatisfactory validation accuracy observed in these experiments highlights the underfitting of the model when utilizing a single scanning.

For the second point, we conducted experiments by incorporating different numbers of scannings in the range of 1–8. It is important to note that when the number of scannings is even (i.e., 2, 4, 6, and 8), the Bi-RNN architecture described in Section II-F is utilized. The experimental results are presented in Fig. 19. It is observed that an increase in the number of scanning orders leads to an improvement in the classification accuracy, thereby validating the effectiveness of the multiscanning strategy. However, there is an apparent saturation in the performance beyond a certain number of scannings, indicating a redundancy in incorporating all the scannings. The optimal number of scannings for IP and PU datasets was observed to be 4, while for the SA dataset, it was found to be 3. Finally, for the HU dataset, which is complex in nature and characterized by an unbalanced spatial distribution and limited samples, the performance was observed to be saturated at six scannings.

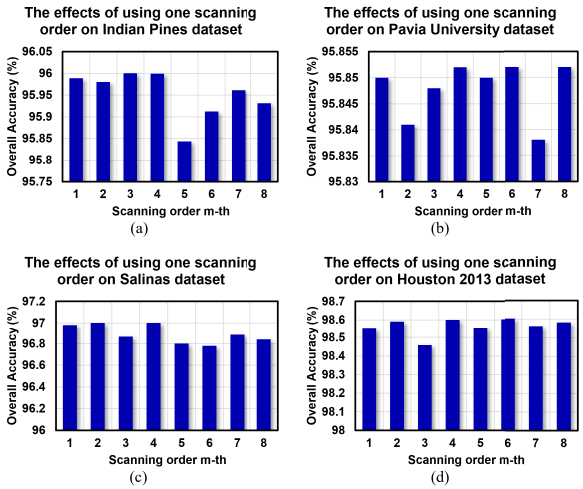


Fig. 18. Effects of deploying one single scanning order for classification performance. (a) IP dataset. (b) PU dataset. (c) SA dataset. (d) HU 2013 dataset.

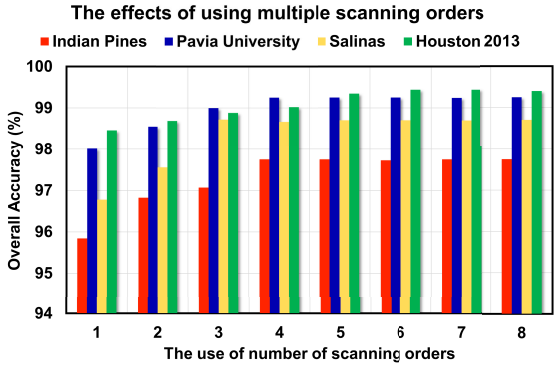


Fig. 19. Effects of using multiple scanning orders for performance.

For the third point of inquiry, we examine the attention given to each scanning order for a particular HSI patch. Given that our approach adopts a spatial perspective, we posit that the spatial distribution resulting from different scanning orders will impact the overall performance, as outlined in Section I. To this end, we present several examples of scanning-based weights in Fig. 20(a). In addition, we perform numerous experiments to calculate the importance of each scanning order for each dataset by counting the number of times each scanning-based weight exceeds the average, as shown in Fig. 20(b). Our findings suggest that the model adaptively allocates different scanning-based weights to each scanning order for a given HSI patch. Some scanning orders receive higher weights (i.e., greater than the average), which could indicate a positive influence on the model training, while others have a negative impact. Furthermore, the proportion of importance for each scanning order varies across datasets when multiple scannings are combined, and this variation is strongly dependent on the dataset itself and initialized patch size (e.g., patch size decides spatial pattern). Nevertheless, for all datasets, scanning-1 and scanning-7 tend to be more important than the others.

2) Comparison Between RNN and 1-D-CNN for Integration With Transformer: With regards to sequence processing, 1-D-CNNs offer the ability to handle sequential data through parallel convolution along the pixel sequence. Thus, we conduct additional experiments to assess the impact of integrating 1-D-CNNs with Transformer models in comparison

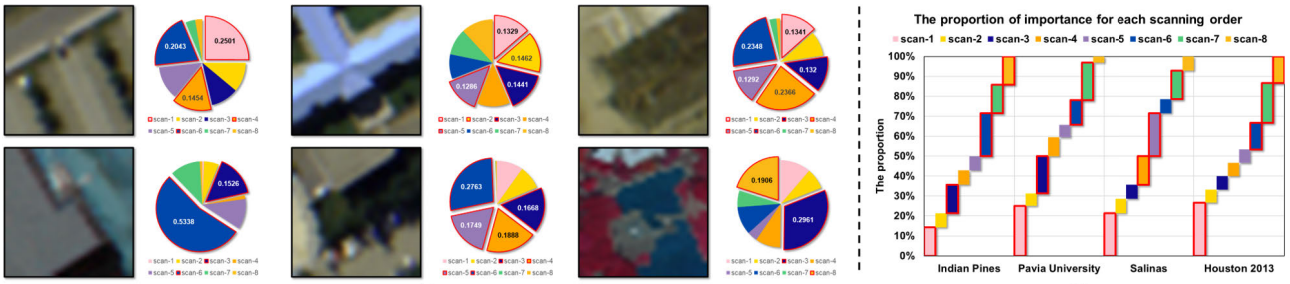


Fig. 20. (a) Illustration of the proportion of scanning-based weights for a single HSI patch, with higher weights (≥0.125) highlighted. (b) Importance proportion of each scanning, as calculated by counting the frequency of each scanning-based weight being greater than 0.125 for different datasets.

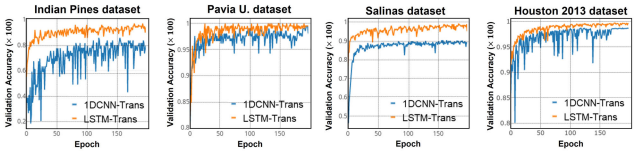


Fig. 21. Validation accuracy of both RT and 1-D-CNN–Transformer is compared across four datasets.

TABLE XIII
EFFECTS OF USING PROPOSED SOFT MASK FOR OA

	Datasets			
	Indian Pines	Pavia University	Salinas	Houston 2013
w/o mask	94.999	97.419	97.196	98.226
w/ spe mask	97.035	98.977	98.405	99.012
w/ spa mask	95.615	98.125	98.179	98.619
w/ both	**97.751**	**99.253**	**98.723**	**99.441**

with RNNs-based Transformer models. The parameters in these experiments were carefully selected to be as similar as possible to the proposed method for 1-D-CNNs, with a single feature extraction layer utilized in both the RT and the 1-D-CNN–Transformer. The specifics of the 1-D-CNN configuration are as follows: a kernel size of 3, 64 kernels, and a stride of 1.

Eventually, the comparison of validation accuracy on four datasets is presented in Fig. 21. The results indicate that the RT outperforms the 1-D-CNN–Transformer in processing pixel sequences, thus highlighting the superiority of RNN (LSTM) in handling sequential data and modeling ordering bias. Furthermore, the RT demonstrates higher stability.

3) Effects of Spectral–Spatial-Based Soft Self-Attention Mask: The integration of the proposed spectral–spatial-based soft mask into the RT encoder enables the network to selectively allocate importance to critical elements while downgrading the significance of less critical ones, as opposed to equal allocation. This results in more discriminative outputs and improved classification accuracy, as demonstrated in Table XIII. In particular, the improvements are significant for the IP and HU datasets, which is attributed to the fact that the two datasets lead to a higher proportion of interfering pixels in the cropped patch with a larger patch size.

Furthermore, it is observed that the spectral-based soft mask outperforms the spatial-based soft mask. This phenomenon is attributed to the higher precision in determining the similarity among pixels by the spectral-based soft mask, which functions as the primary component, while the spatial-based mask can be seen as an auxiliary aid.

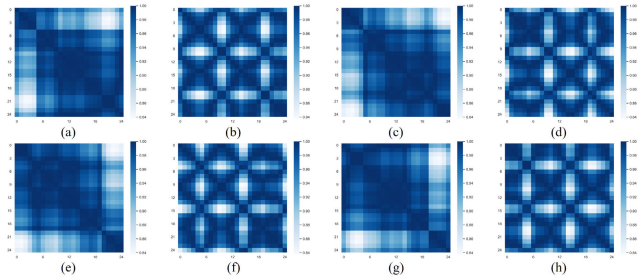


Fig. 22. Example of the spectral-based soft masks ($\boldsymbol{M}_m^{\text{spe}}$) with a size of 25 × 25 (i.e., patch size = 5). The lighter parts represent the lower soft weight. (a) Ordering-1. (b) Ordering-2. (c) Ordering-3. (d) Ordering-4. (e) Ordering-5. (f) Ordering-6. (g) Ordering-7. (h) Ordering-8.

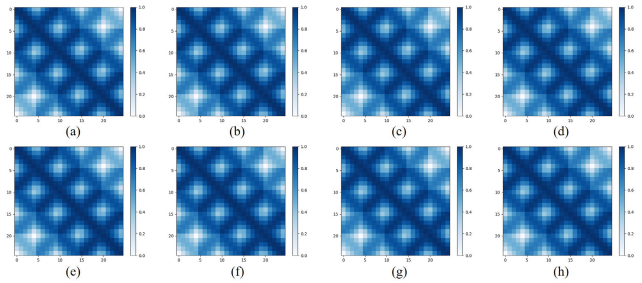


Fig. 23. Example of the spatial-based soft masks ($\boldsymbol{M}_m^{\text{spa}}$) with a size of 25 × 25. We find that the patterns of spatial soft mask with different scannings are same. The lighter parts represent the lower soft weight. (a) Ordering-1. (b) Ordering-2. (c) Ordering-3. (d) Ordering-4. (e) Ordering-5. (f) Ordering-6. (g) Ordering-7. (h) Ordering-8.

TABLE XIV
INVESTIGATION OF EFFECTS OF DIFFERENT POSITIONAL EMBEDDING (PE) STRATEGIES

	Datasets			
	Indian Pines	Pavia University	Salinas	Houston 2013
Absolute PE	85.169	85.951	87.993	84.651
Learned PE	92.559	97.225	96.125	98.512
Sin/cos PE	93.618	95.119	96.002	97.133
Ours	**97.751**	**99.253**	**98.723**	**99.441**

Furthermore, it was discovered that the pattern of the mask is determined by the positional embedding. This results in the generation of different spectral-based soft masks and identical spatial-based soft masks for the multiscanning strategy, as demonstrated in Figs. 22 and 23, respectively. This allows for the appropriate allocation of attention weights to each pixel. This hypothesis is supported by the observed improvements in classifications on land-cover boundaries, and as previously mentioned, the interfering pixels are often cropped into an HSI patch across different land-cover regions.

4) Comparison on Positional Embedding Methods: In this study, we propose utilizing RNNs for the purpose of positional

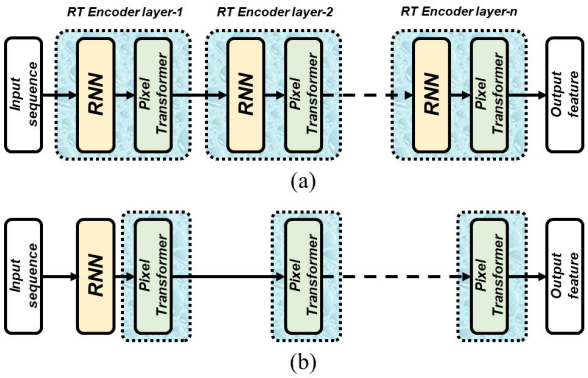


Fig. 24. Two designs of integrating RNN with Transformer. (a) Design-1 includes RNN and Transformer into one RT encoder. (b) Design-2 deploys RNN once as the positional embedding before several Transformer layers.

TABLE XV
EFFECTS OF DIFFERENT DESIGNS FOR RT ENCODER

		The number of RT layers						
		1	2	3	4	5	6	7
Design-1	IP	96.881	97.552	**97.751**	97.482	97.028	97.124	96.881
	PU	97.901	**99.253**	98.303	98.214	97.999	97.738	97.201
	SA	97.581	**98.723**	98.349	98.226	98.001	97.508	97.134
	HU	99.058	99.128	**99.441**	98.917	98.256	98.002	97.449
Design-2	IP	94.228	95.337	95.412	95.047	94.248	93.831	93.004
	PU	95.127	95.999	96.237	96.338	96.041	95.229	95.071
	SA	96.194	96.218	96.821	96.713	96.217	95.462	94.512
	HU	97.021	97.665	97.568	97.120	96.771	96.025	95.881

embedding. The utilization of RNNs allows for the generation of continuous dynamical parameters through its internal ordering mechanism. To validate the proposed strategy, a series of experiments were conducted by implementing various positional embedding methods, including the following: 1) the proposed strategy utilizing RNNs; 2) absolute positional embedding in [40]; 3) learned positional embedding as described in [18]; and 4) sinusoidal/cosine positional embedding. The results of deploying different positional embedding methods on the proposed model are presented in Table XIV.

In this study, the utilization of absolute positional embedding involves concatenating an index number with the sequence length. However, this approach can result in substantial differences among sequence values, making it difficult to manage long sequence lengths and leading to unstable results. On the other hand, the commonly used learned positional embedding is characterized by its noise, lack of structure, independence, and discrete nature. This makes it challenging to capture the positional information among values, as it assigns a position to each value instead of considering the steps in the sequence. The sinusoidal/cosine positional embedding approach, which is well known for its relative design that captures the relationship among steps, is designed manually with static numbers. This can be deemed as inadequate and inflexible for different situations. In contrast, the proposed positional embedding strategy is composed of a set of positional embedding and feature embedding with dynamic, learnable parameters, allowing for flexibility in various situations.

5) Design of RT Encoder: In prior works, positional embedding is typically implemented prior to multiple Transformer layers. Our proposed method, however, integrates RNNs with the Transformer architecture, where the RNN serves as the positional embedding component. To arrive at an optimal combination design, two strategies were implemented, as depicted in Fig. 24. In addition, further experiments were conducted to examine the impact of the two designs on performance. The results are presented in Table XV.

The results of the experiments conducted in this study demonstrate that Design-1, which integrates the RNN and Transformer into a single RT encoder, outperforms Design-2 in terms of classification performance. It is observed that the OA increases in a corresponding manner, as the number of layers increases initially. The IP and HU datasets achieve the optimal results when three layers of RT encoder are utilized, whereas the optimal number of layers for the PU and SA datasets is 2. It is noted that an excessive increase in the number of layers may result in a degradation of classification performance. Thus, the selection of an appropriate number of RT layers is critical for the generation of discriminative features.

IV. CONCLUSION

This study presents a multiscanning-based RNN-enhanced Transformer for HSI classification. The proposed method leverages the strengths of both RNN and Transformer while mitigating their respective limitations. The RNN-induced positional embedding adds an ordering bias to the Transformer, enabling the continuous capture of input dependencies. In addition, the proposed multiscanning-controlled positional embedding comprehensively captures the relationships among pixels with different orderings, effectively serving as an augmentation operation for sequential models to train a more robust model. The spectral–spatial-based soft self-attention masks enable automatic and appropriate feature enhancement based on the data. The final classification is performed using the multiscanning-enhanced features integrated within the overall structure. Compared with other methods, the proposed approach yields competitive results with fewer parameters and without any CNNs units on four public datasets, with particular improvement in the classification of land-cover boundaries. Future research may aim to further advance the utilization of RNN and Transformer in the HSI classification task.

APPENDIX

In this section, a comprehensive overview of RNNs and the self-attention mechanism in Transformer models is presented. The purpose of this review is to provide a better understanding of the fundamental concepts and principles behind these advanced neural network models.

A. RNN

RNNs are used to perform sequential modeling. It is because RNNs' internal ordering bias constructs the global dependencies among inputs step by step. Give the sequential data $\boldsymbol{S}$ with length of n, where $\boldsymbol{S}[s]$, $s = 0, 1, \ldots, n$, denotes either a scalar or a vector. The output $\boldsymbol{P}$ values, a sequential feature, with the same length n from RNN are calculated by

$$\boldsymbol{P}[s] = \begin{cases} 0, & \text{if } s = 0 \\ \phi(\boldsymbol{P}[s-1], \boldsymbol{S}[s]), & \text{otherwise} \end{cases} \tag{34}$$

where ϕ is a nonlinear function, such as rectified linear unit (ReLU) or tanh.

The update rule of (34) is usually implemented as follows:

$$\boldsymbol{P}[s] = \phi(\boldsymbol{W}\boldsymbol{S}[s] + \boldsymbol{U}\boldsymbol{P}[s-1] + \boldsymbol{b}) \tag{35}$$

where $\boldsymbol{W}$, $\boldsymbol{U}$, and $\boldsymbol{b}$ represent the relevant transformation parameters, respectively. Optionally, for some tasks, such as HSI classification, most methods need only one output, i.e., $\boldsymbol{P}[n]$. RNN is a biased model with loading the inputs sequentially; the positional information can be inherently encoded [39].

B. Transformer With Self-Attention

The Transformer architecture adopts a self-attention mechanism, enabling simultaneous modeling of the global dependencies among elements in the sequence, regardless of their positional relationship. This approach offers advantages in terms of parallel computation and addresses issues prevalent in traditional recurrent models, as discussed in [17].

The self-attention mechanism projects each element of sequential data $\boldsymbol{S}$ by applying three different linear transformations $\boldsymbol{W}^q$, $\boldsymbol{W}^k$, and $\boldsymbol{W}^v$ to obtain three features, i.e., query ($\boldsymbol{Q}$), key ($\boldsymbol{K}$), and value ($\boldsymbol{V}$). The output of the self-attention mechanism, $\boldsymbol{F}$, is then calculated as follows:

$$\boldsymbol{F} = \mathrm{SA}(\boldsymbol{S}) = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{36}$$

where d_k is the feature dimension of $\boldsymbol{K}$ and SA is noted as self-attention operation. By means of the abovementioned transformation, Transformer is capable of parallel computation of the long-range dependencies among the elements in the sequence, thereby effectively addressing the limitations of traditional RNN models in certain tasks.

It is important to note that the SA layer does not contain positional information and, therefore, fails to make use of sequential information. To address this issue, positional information is encoded into the original input through the following formulation:

$$\boldsymbol{S} = \gamma\boldsymbol{S} + \delta\boldsymbol{Pos} \tag{37}$$

where $\boldsymbol{Pos}$ serves as positional information, having the same size with $\boldsymbol{S}$. It is typically obtained by employing either the sine/cosine function with a fixed value or by incorporating them as additional, learnable parameters. Meanwhile, $\gamma = \delta$ is set commonly [18].

REFERENCES

[1] W. Lv and X. Wang, "Overview of hyperspectral image classification," *J. Sensors*, vol. 2020, pp. 1–13, Jul. 2020.

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

[3] M. Y. Teng, R. Mehrubeoglu, S. A. King, K. Cammarata, and J. Simons, "Investigation of epifauna coverage on seagrass blades using spatial and spectral analysis of hyperspectral images," in *Proc. 5th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2013, pp. 1–4.

[4] G. Notesco, E. Ben Dor, and A. Brook, "Mineral mapping of Makhtesh Ramon in Israel using hyperspectral remote sensing day and night LWIR images," in *Proc. 6th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2014, pp. 1–4.

[5] P. Ghamisi et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[6] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.

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

[8] 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.

[9] 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.

[10] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Scalable recurrent neural network for hyperspectral image classification," *J. Supercomput.*, vol. 76, no. 11, pp. 8866–8882, Feb. 2020.

[11] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral–spatial LSTMs," *Neurocomputing*, vol. 328, pp. 39–47, Feb. 2019.

[12] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[13] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, and H. Zhou, "Spatial sequential recurrent neural network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4141–4155, Nov. 2018.

[14] C. Shi and C.-M. Pun, "Multi-scale hierarchical recurrent neural networks for hyperspectral image classification," *Neurocomputing*, vol. 294, pp. 82–93, Jun. 2018.

[15] W. Zhou, S. Kamata, Z. Luo, and H. Wang, "Multiscanning strategy-based recurrent neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5521018.

[16] R. Jing, "A self-attention based LSTM network for text classification," *J. Phys., Conf. Ser.*, vol. 1207, Apr. 2019, Art. no. 012008.

[17] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.

[18] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

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

[20] 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.

[21] Y. Qing, W. Liu, L. Feng, and W. Gao, "Improved transformer net for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 11, p. 2216, Jun. 2021.

[22] S. Woo et al., "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 3–19.

[23] 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.

[24] Y. Zhou and Y. Wei, "Learning hierarchical spectral–spatial features for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1667–1678, Jul. 2016.

[25] K. Gao, B. Liu, Z. Xue, X. Zuo, Y. Sun, and M. Dai, "Deep transformer network for hyperspectral image classification," *Academic J. Comput. Inf. Sci.*, vol. 4, no. 7, pp. 11–17, 2021.

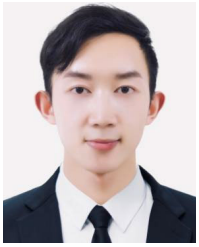
[26] D. Ibañez, R. Fernandez-Beltran, F. Pla, and N. Yokoya, "Masked auto-encoding spectral–spatial transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5542614.

[27] Z. Wang, Y. Ma, Z. Liu, and J. Tang, "R-transformer: Recurrent neural network enhanced transformer," 2019, *arXiv:1907.05572*.

[28] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," 2018, *arXiv:1807.03819*.

[29] K. Tran, A. Bisazza, and C. Monz, "Recurrent memory networks for language modeling," 2016, *arXiv:1601.01272*.

[30] K. Tran, A. Bisazza, and C. Monz, "The importance of being recurrent for modeling hierarchical structure," 2018, *arXiv:1803.03585*.
[31] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, *arXiv:1803.02155*.
[32] M. Xu Chen et al., "The best of both worlds: Combining recent advances in neural machine translation," 2018, *arXiv:1804.09849*.
[33] J. Hao, X. Wang, B. Yang, L. Wang, J. Zhang, and Z. Tu, "Modeling recurrence for transformer," in *Proc. Conf. North*, 2019, pp. 1198–1207.
[34] X. Hu, W. Yang, H. Wen, Y. Liu, and Y. Peng, "A lightweight 1-D convolution augmented transformer with metric learning for hyperspectral image classification," *Sensors*, vol. 21, no. 5, p. 1751, Mar. 2021.
[35] X. He, Y. Chen, and Z. Lin, "Spatial–spectral transformer for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 3, p. 498, Jan. 2021.
[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[37] H. Zhang, C. Gong, Y. Bai, Z. Bai, and Y. Li, "3-D-ANAS: 3-D asymmetric neural architecture search for fast hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5508519.
[38] 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.
[39] K. Chen, R. Wang, M. Utiyama, and E. Sumita, "Recurrent positional embedding for neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1361–1367.
[40] J. Zheng, S. Ramasinghe, and S. Lucey, "Rethinking positional encoding," 2021, *arXiv:2107.02561*.
[41] X. Liu, H. Yu, I. Dhillon, and C. Hsieh, "Learning to encode position for transformer with continuous dynamical model," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6327–6335.
[42] 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.
[43] J. Rae and A. Razavi, "Do transformers need deep long-range memory?" in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7524–7529.
[44] I. Beltagy, M. E. Peters, and A. Cohan, "LongFormer: The long-document transformer," 2020, *arXiv:2004.05150*.
[45] Z. Manzil et al., "Big bird: Transformers for longer sequences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17283–17297.
[46] Z. Huang, P. Xu, D. Liang, A. Mishra, and B. Xiang, "TRANS-BLSTM: Transformer with bidirectional LSTM for language understanding," 2020, *arXiv:2003.07000*.
[47] X. Xue et al., "Convolutional recurrent neural networks with a self-attention mechanism for personnel performance prediction," *Entropy*, vol. 21, no. 12, p. 1227, Dec. 2019.
[48] A. Fahim, Q. Tan, M. Mazzi, M. Sahabuddin, B. Naz, and S. U. Bazai, "Hybrid LSTM self-attention mechanism model for forecasting the reform of scientific research in Morocco," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–14, May 2021.
[49] Y. Gao and T. Chua, "Hyperspectral image classification by using pixel spatial correlation," in *Proc. Int. Conf. Multimedia Modeling*, vol. 7732, 2013, pp. 141–151.
[50] R. Gogineni and (Dec. 13, 2019). *Hyperspectral Image Classification*. [Online]. Available: https://www.intechopen.com/chapters/70188



Weilian Zhou (Graduate Student Member, IEEE) received the B.S. degree from the Guilin University of Technology, Guilin, China, in 2017, and the M.S. degree from the Graduate School of Information, Production, and Systems (IPS), Waseda University, Fukuoka, Japan, in 2021, where he is currently pursuing the Ph.D. degree with the Image Media Laboratory.

He serves as a Research Associate with the IPS Research Center, Waseda University. His research interests include remote sensing, computer vision, pattern recognition, and deep learning, with a particular focus on the application of deep learning techniques in hyperspectral imaging.

Mr. Zhou is a member of the IEEE Geoscience and Remote Sensing Society.

Sei-Ichiro Kamata (Member, IEEE) received the M.S. degree in computer science from Kyushu University, Fukuoka, Japan, in 1985, and the Doctor of Computer Science degree from the Kyushu Institute of Technology, Kitakyushu, Japan, in 1995.

From 1985 to 1988, he was with NEC Ltd., Kawasaki, Japan. In 1988, he joined the faculty at the Kyushu Institute of Technology. In 1990 and 1994, he was a Visiting Researcher with The University of Maine, Orono, ME, USA. From 1996 to 2001, he has been an Associate Professor with the Department of Intelligent System, Graduate School of Information Science and Electrical Engineering, Kyushu University. Since 2003, he has been a Professor with the Graduate School of Information, Production and Systems, Waseda University, Fukuoka. His research interests include image processing, pattern recognition, image compression, and space-filling curve application.

Dr. Kamata is a member of The Institute of Image Information and Television Engineers (ITE) in Japan.

Haipeng Wang (Senior Member, IEEE) received the B.S. and M.S. degrees in mechanical and electronic engineering from the Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively, and the Ph.D. degree in environmental systems engineering from the Kochi University of Technology, Kochi, Japan, in 2006.

He was a Visiting Researcher with the Graduate School of Information, Production and Systems, Waseda University, Fukuoka, Japan, in 2008. He is currently a Professor with the Key Laboratory of Electromagnetic Wave Information Science (MoE), Department of Communication Science and Engineering, School of Information Science and Engineering, Fudan University, Shanghai, China. His research interests include signal processing, Synthetic Aperture Radar (SAR) image processing and analysis, speckle statistics, and applications to forestry and oceanography, machine learning, and its applications to SAR images.

Dr. Wang has been a member of Technical Program Committee of IEEE Geoscience and Remote Sensing Symposium (IGARSS) since 2011. He was a recipient of the Dean Prize of School of Information Science and Engineering, Fudan University, in 2009 and 2017. He serves as an Associate Editor for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.

Xi Xue (Student Member, IEEE) received the B.S. degree from Shanghai Ocean University, Shanghai, China, in 2018, and the M.S. degree from the Graduate School of Information, Production, and Systems (IPS), Waseda University, Fukuoka, Japan, in 2021, where she is currently pursuing the Ph.D. degree with the Image Media Laboratory.

Her research interests include image processing, computer vision, and deep learning, with a particular focus on the application of deep learning techniques in medical images.