Study on Multiscanning-Based Hyperspectral Image Classification Using Recurrent Neural Networks and Transformers

Weilian ZHOU

January 2024

Waseda University Doctoral Dissertation

Study on Multiscanning-Based Hyperspectral Image Classification Using Recurrent Neural Networks and Transformers

Weilian ZHOU

Graduate School of Information, Production, and Systems
Waseda University

January 2024

ACKNOWLEDGMENTS

This dissertation is the end of my journey in pursuing a PhD. Let me take this opportunity to convey my gratitude to the nicest people I owe for this work.

I express my deepest gratitude to my advisor, Professor Sei-ichiro Kamata, for his guidance, support, and invaluable insights throughout the research journey. His patience and dedication have shaped this study and pushed me to reach my full potential.

I also thank my thesis committee members for their valuable feedback, constructive criticism, and scholarly contributions. Their expertise and scholarly guidance have greatly enriched this research work.

I am grateful to my colleagues and friends who have provided encouragement, support, and fruitful discussions. Their insights and collaboration have contributed to developing and refining this research.

I want to acknowledge the financial support provided by our school. Without their support, this research would not have been possible.

Finally, I express my most profound appreciation to my family for their encouragement and understanding. Their constant support and belief in me have been the driving force behind my pursuit of knowledge.

This research work would not have been accomplished without the contributions and support of all those mentioned above and many others who have played a role, no matter how small, in this journey.

Thank you all.

ABSTRACT

Hyperspectral imaging (HSI) represents an advanced mode of remote sensing (RS), capturing high-resolution spectral information for each pixel in an image across a continuous spectrum of hundreds of bands. This dense, multidimensional information enables a high degree of differentiation between various land-cover types based on their unique spectral signals. HSI classification, a critical process in HSI analysis, involves assigning each pixel in the HSI to a specific class or category based on its spectral characteristics. The application of this technique spans a range of fields, such as environmental monitoring, mineralogy, precision agriculture, military defense, urban planning, and healthcare, driving advancements and insights in these domains. Despite the rich information provided by HSI, the high dimensionality of the data presents challenges that require sophisticated machine-learning techniques for effective and accurate classification.

In recent years, using machine learning or deep learning techniques for HSI classification has rapidly expanded due to their powerful capacity for feature representation and automatic learning. Traditional machine learning methods, which rely heavily on manual feature extraction, have been gradually replaced by sophisticated deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs have shown an ability to capture and utilize spatial information, while RNNs and their variants can handle the sequential nature of spectral responses in hyperspectral data. Using these models has substantially improved classification performance, offering more accurate and detailed insights from hyperspectral data. However, despite these advances, the pursuit of a model that is both highly accurate and computationally efficient remains a crucial challenge in the field of HSI classification. Therefore, the primary aim is to develop such a model that not only improves the classification accuracy but also optimizes computational resources, making it a valuable tool for real-world applications.

Therefore, the central focus of this dissertation is the development of robust and efficient models for HSI classification. It represents a critical initial step towards enhancing our capabilities to analyze hyperspectral data and harness its potential for various applications.

Despite the successes of deep learning in HSI classification, specific challenges persist, limiting the full realization of its potential. One significant concern is the inherent variability in the spatial distribution of HSI patches within the same class. Such variations in spatial complexity can lead to disparate spectral signals for the same type of land cover, thereby complicating the classification process. Traditional deep learning approaches (i.e., CNN-based and RNN-based), often guided by a singular view of an HSI patch, may fail to capture these nuances, limiting their ability to effectively serve as generative models, particularly when training data is scarce. Consequently, these models struggle to interpret other patches with different spatial distributions, even if they belong to the same class. This limitation impedes their ability to generalize effectively and restricts their performance in the face of diverse spatial complexities.

Under such constrained conditions, an enticing prospect for enhancing model performance involves the integration of multi-directional or multi-view perspectives into deep learning algorithms. The model becomes more generative and flexible by processing HSI data from various angles or views, enabling it to better recognize and understand the diverse spatial complexities within a class. This approach enriches the model with more comprehensive spectral-spatial feature representations and enhances the model's generalization capability. It unlocks the potential to improve classification accuracy, even when operating with limited HSI samples. In light of this, we observed that current state-of-the-art methods can still be enhanced, particularly when learning features from multiple directions. To address this, we introduce an innovative approach that combines a multiscanning strategy with specially designed sequential neural models, such as RNNs and Transformers. This approach uses multiple scanning orders to offer a diverse range of positional information, which can help us understand the spatial distribution from various angles or orientations. Consequently, it creates a potential pool of features that counter the low generative capabilities caused by the scarcity of samples.

Chapter 1 serves as the foundational guide for this dissertation. It begins with an overview of the research background followed by a review of prior work, highlighting specific challenges in the methods based on CNNs and RNNs. Subsequently, we introduce the central concept encapsulating our methodology for easier comprehension. This chapter also outlines the key contributions and the structure of this dissertation.

Chapter 2 presents the detailed scenario of proposing the multiscanning strategy. The chapter commences by highlighting the restrictions associated with using CNNs for HSI classification, which paves the way for adopting RNNs in HSI classification, followed by a review of the existing scanning techniques utilized for RNNs. Based on this groundwork, the chapter delves into a comprehensive discussion of our specific observations that shape the creation of the multiscanning strategy. The dialogue advances progressively through each section, constructing a context and framework for the strategy. Consequently, this chapter confirms the feasibility of employing a multiscanning strategy with RNNs for HSI classification, achieving satisfactory performance and laying the groundwork for future research.

Chapter 3 elaborates on our investigation of incorporating a multiscanning strategy with RNNs as well as attention mechanism for HSI classification. This chapter aims to further improve the Chapter 2 with an in-depth walkthrough of the designed networks including two novel proposals, 'scanning order-based attention' and 'RNN-based multiscanning feature fusion'. Subsequently, a series of experiments are conducted to assess the practicality of our approach, supported by a subsequent analysis that underpins our concepts. More specifically, our findings demonstrate a noticeable improvement in accuracy compared to other state-of-the-art methods, while the model size is considerably smaller. This results in less computational burden, underlining the greater efficiency of our approach. Meanwhile, compared with Chapter 2, the effectiveness of two proposals is verified by the overall improvements on classification accuracies. This study marks a significant step forward and opens new avenues for future research in this domain.

Chapter 4 aims to further enhance the previous Chapter 3. It addresses the constraints of RNNs by integrating Transformer models with our multiscanning approach. By harnessing the power of Transformers, renowned for their capability to model long-range dependencies and intricate correlations effectively, we strive to enhance the performance and robustness of our methodology. We begin by revisiting the limitations identified in our previous efforts and then introducing the Transformer and its self-attention mechanism. Subsequently, a high-level overview of our proposed methodology is provided, outlining the main stages and objectives. The subsequent section offers a detailed explanation of our approach, discussing the incorporation of Transformers into our multiscanning framework and underscoring the key enhancements and modifications. Lastly, we conduct a series of experiments to validate the feasibility of our approach, accompanied by an analysis that bolsters our propositions. As for the evaluation of generative capabilities, our approach outperforms other leading methods on several HSI datasets. Remarkably, this achievement is coupled with lighter model size and reduced processing time, thus demonstrating the efficiency and effectiveness of our method.

Chapter 5 wraps up this study, highlighting our significant findings and proposing future research directions. Our proposed methodology has demonstrated impressive outcomes, notably surpassing other methods in accuracy and computational efficiency. Furthermore, it has established itself as an innovative feature augmentation method for HSI classification, especially when dealing with limited sample sizes. Moving forward, developing the multiscanning strategy into a 3D version with a novel spectral-spatial 3D Transformer is an exciting prospect. Also, introducing hypergraph techniques, with their inherent sparsity, into the Transformer model holds considerable potential for achieving even better results in the future.

Keywords: Hyperspectral image classification; Multiscanning Strategy; Recurrent neural Network; Transformer; Deep Learning.

Contents

List of Figures

List of Tables

List of Main Symbols

$\mathbf{X} \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times C}$	The original HSI with 3D size
$\mathcal{H}, \mathcal{W}, C$	The size of HSI's height, width, and band (channel).
$\mathbf{X}^{(i,j)} \in \mathbb{R}^{p\times p\times C}$	The cropped HSI patch with 3D size centered at pixel $\mathbf{x}^{(i,j)}$.
$\mathbf{x}^{(i,j)} \in \mathbb{R}^{1\times 1\times C}$	The pixel with its spatial coordinate (i, j) in the original HSI
$\mathbf{S}_m^{(i,j)} \in \mathbb{R}^{p^2\times C}$	The scanned pixel-sequence from $\mathbf{X}^{(i,j)}$ with m-th ordering.
$\mathbf{H}_m^{(i,j)} \in \mathbb{R}^{p^2\times d}$	The hidden state features in RNNs at m-th ordering.
$\mathbf{Y}_m^{(i,j)} \in \mathbb{R}^{p^2\times d}$	The output features from RNNs at m-th ordering.
$\mathbf{PE}_m^{(i,j)} \in \mathbb{R}^{p^2\times d}$	The positional embeddings in Transformer for each pixel-sequence $\mathbf{S}_m^{(i,j)}$.
$\tilde{\mathbf{T}}_m^{(i,j),l} \in \mathbb{R}^{p^2\times d}$	The output features from one layer of Spectral Transformer.
$\mathbf{F}_m^{(i,j)} \in \mathbb{R}^{p^2\times d}$	The final output features from L-th layer of the Spectral Transformer.
$\mathbf{A}_m^{(i,j)} \in \mathbb{R}^{p^2\times 1}$	The attention weight for each pixel in the Spectral Transformer.
$\mathbf{y}_m^{(i,j)} \in \mathbb{R}^{1\times d}$	The feature representation from each RT encoder under m-th scanning order.
$\mathbf{M}_m^{spa} \in \mathbb{R}^{p^2\times p^2}$	The spatial-based soft self-attention mask at m-th scanning order.
$\mathbf{M}_m^{spe} \in \mathbb{R}^{p^2\times p^2}$	The spectral-based soft self-attention mask at m-th scanning order.
p	The cropping patch size.
d	The dimension of feature embeddings.
m	The m-th scanning order in the multiscanning strategy, $m = 1, 2, 3, ..., \mathcal{M}$.
s	The s-th step in the pixel-sequence or feature-sequence, $s = 0, 1, 2, ..., p^2 - 1$.
l	The l-th layer in Spectral Transformer, $l = 1, 2, ..., L$

Chapter 1

General Introduction

This chapter presents a general background of hyperspectral image with classification tasks and leads to the central concept of our proposed approaches, i.e., multiscanning strategy with sequential neural networks. We then summarize the organization of this dissertation with the correspondence between succeeding chapters and our preliminary work.

1.1 Research Background

1.1.1 Hyperspectral Imaging (HSI)

Hyperspectral imaging (HSI) is an advanced remote sensing (RS) technique that captures images in numerous dense and continuous spectral bands across the electromagnetic spectrum spanning from visible to infrared wavelength by satellites or unmanned aerial vehicles (UAV) [47]. Unlike traditional imaging techniques, hyperspectral imaging provides detailed spectral information for each pixel in a portrayed surface of the earth, allowing for more precise characterization of materials and objects [41]. This technology has found applications in various fields such as follows. 1) Agriculture [15]: hyperspectral imaging can reveal information about crop health, soil fertility,

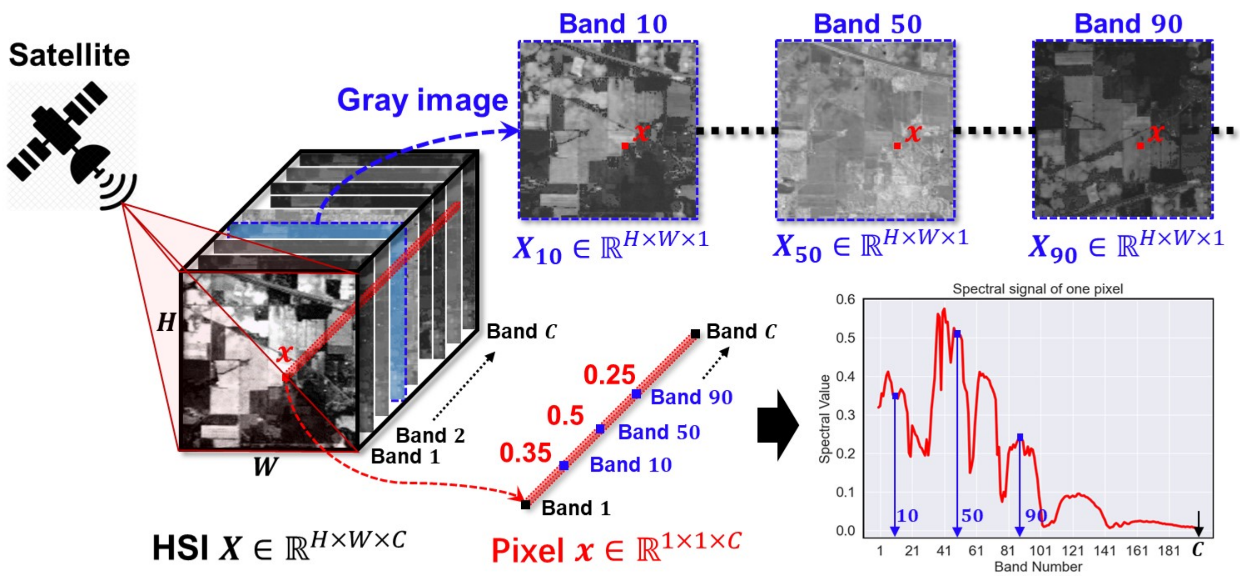

Satellite
Gray image
Band 10
Band 50
Band 90
x
$X_{10} \in \mathbb{R}^{H\times W\times 1}$
$X_{50} \in \mathbb{R}^{H\times W\times 1}$
$X_{90} \in \mathbb{R}^{H\times W\times 1}$
H
W
Band C
Band 2
Band 1
HSI $X \in \mathbb{R}^{H\times W\times C}$
Band C
0.25
Band 90
0.5
Band 50
0.35
Band 10
Band 1
Pixel $x \in \mathbb{R}^{1\times 1\times C}$
Spectral signal of one pixel
Spectral Value
0.6
0.5
0.4
0.3
0.2
0.1
0.0
10
50
90
1
21
41
61
81
101
121
141
161
181
C
Band Number

Figure 1.1 The illustration depicts the components of the HSI. Each pixel within the image is a high-dimensional vector. Each image band presents a grayscale depiction, each pixel recording the spectral reflectance value at that specific band.

and pest/disease presence that is not visible in a regular photograph; 2) environmental monitoring [68]: it can be used to monitor vegetation, track changes in ecosystems, assess water quality, and detect pollutants; 3) geology [50]: different minerals reflect light in different ways, and hyperspectral imaging can be used to detect and identify them; 4) urban planning [1]: HSIs can help identify materials used in buildings and roads, and assist in urban planning; 5) defense and security [57]: hyperspectral sensors can identify objects and materials based on their unique spectral signals, even if they are camouflaged or obscured.

As shown in Figure 1.1, HSI is often represented as a three-dimensional (3D) cube, where the two spatial dimensions correspond to the rows and columns of pixels in the image, and the third (spectral) dimension corresponds to the different spectral bands. It is often referred to as a hyperspectral data cube. For example, if we have an image of 145×145 pixels with 200 spectral bands, the size of the hyperspectral data cube would be $145\times145\times200$. Meanwhile, it contains a stack of grey-scale images, with each image in the stack representing a different spectral band.

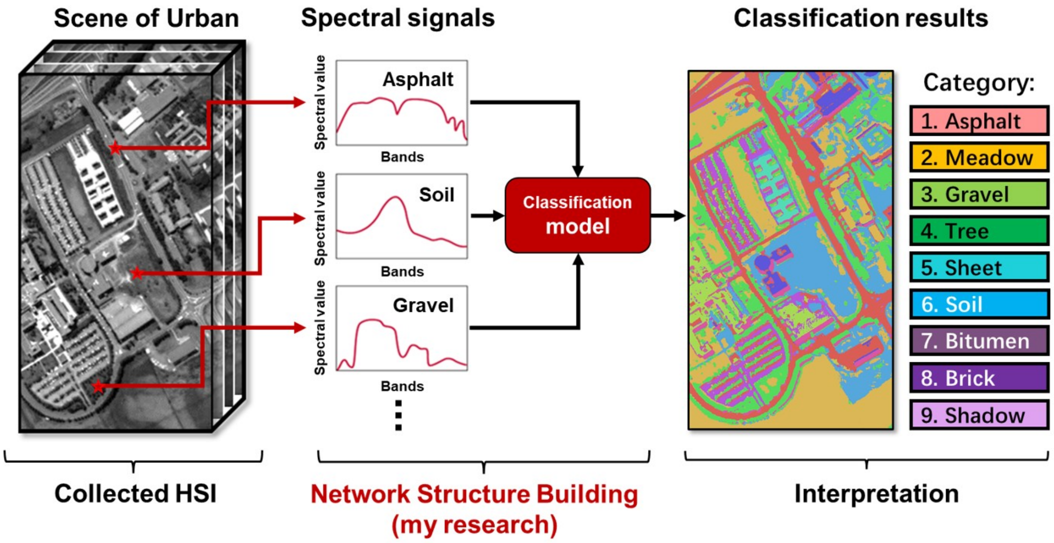


Figure 1.2 The flowchart of HSI classification. After obtaining HSI, we need to build a model to identify spectral signals from different land-cover classes in the HSI. This interpretation is applied to the entire HSI to yield classification results. Getting an accurate classification result for a better understanding of the land-cover scenario depends mainly on how the model is built. This dissertation aims to build a precise model to improve its applications.

When viewed together, these images provide a continuous spectrum for each pixel in the image. In each spectral band, the pixel's intensity value represents the reflectance or radiance of that portion of the scene at the specific wavelength of the band. The materials present at that location in the scene can be identified by looking at the spectrum for a single pixel (i.e., the set of intensity values across all bands). Different materials have unique spectral signals, which are variations in reflectance or absorption characteristics as a function of wavelength. These spectral signals are regarded as 'fingerprints,' allowing us to distinguish between different materials [19].

1.1.2 HSI Classification

HSI classification is essential in analyzing and interpreting hyperspectral data. It involves assigning each pixel in the HSI to predefined classes or categories based on its spectral signal [28]. The

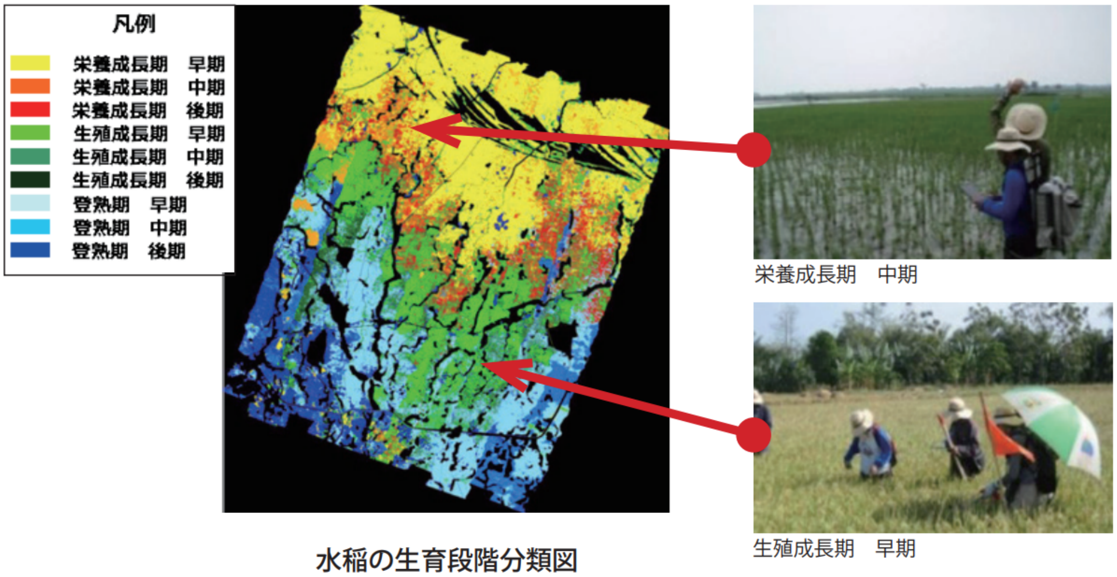


Figure 1.3 The applications of HSI classification in the agriculture by HISUI project in Japan.

goal is to accurately identify and distinguish different land cover types or materials in the scene, as shown in Figure 1.2. As the first step for practical utilization, HSI classification becomes critical because the high accuracy of classification results can enhance the quality of those applications. Consequently, this task has been widely studied in the RS field [91]. For example, Figure 1.2 shows the contributions of HSI classification on urban planning; we can easily find that the classification results can provide information to distinguish roofing materials, pavement types, vegetation, and other surface materials. This detailed understanding aids urban planners in understanding the current land use structure, which is crucial for making informed planning decisions. Furthermore, the green spaces within cities are essential for recreation, air quality, and biodiversity. HSI classification can quantify the extent of green spaces and even identify different vegetation types. It helps monitor the health and distribution of urban forests and parks, informing decisions about preserving and expanding such spaces.

As shown in Figure 1.3, the Hyperspectral Imager Suite (HISUI) project[1], led by Japan Space

[1] https://www.hisui.go.jp/en/project/index.html

System (JSS) and Japan Aerospace Exploration Agency (JAXA), revolutionizes remote sensing with its 185-band coverage from visible to short-wavelength infrared. It enables detailed mineral distribution, forest classification, and soil analysis, heralding a new stage in remote sensing technology. For more information, visit the HISUI project webpage[2].

1.2 Previous Works

1.2.1 Traditional Methods

Traditional HSI classification often uses handcrafted features and classic machine learning methods. Techniques like Principal Component Analysis (PCA) [54], Linear Discriminant Analysis (LDA) [74], Independent Component Analysis (ICA) [3], and wavelet transforms are used to reduce the dimensions of hyperspectral data and capture critical information. Then, classifiers such as support vector machines (SVM) [38] and random forests categorize pixels into specific classes. However, these methods have their drawbacks. Firstly, they heavily rely on selecting and designing the right features, which can be difficult and time-consuming [2]. Secondly, they might need help dealing with the high-dimensional nature of hyperspectral data, leading to computational and memory problems. Also, these dimension reduction methods may disrupt the spectral continuity of HSI data [48]. Lastly, traditional methods may fail to capture complex and nonlinear relationships in the data, limiting their classification accuracy.

1.2.2 Deep Learning-Based Methods

Deep learning has recently become a popular method for HSI classification. Techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown remarkable results in computer vision tasks [10]. With labeled data, these models can learn essential features

[2] https://www.jspacesystems.or.jp/jss/files/2021/06/HISUI_guidebook.pdf

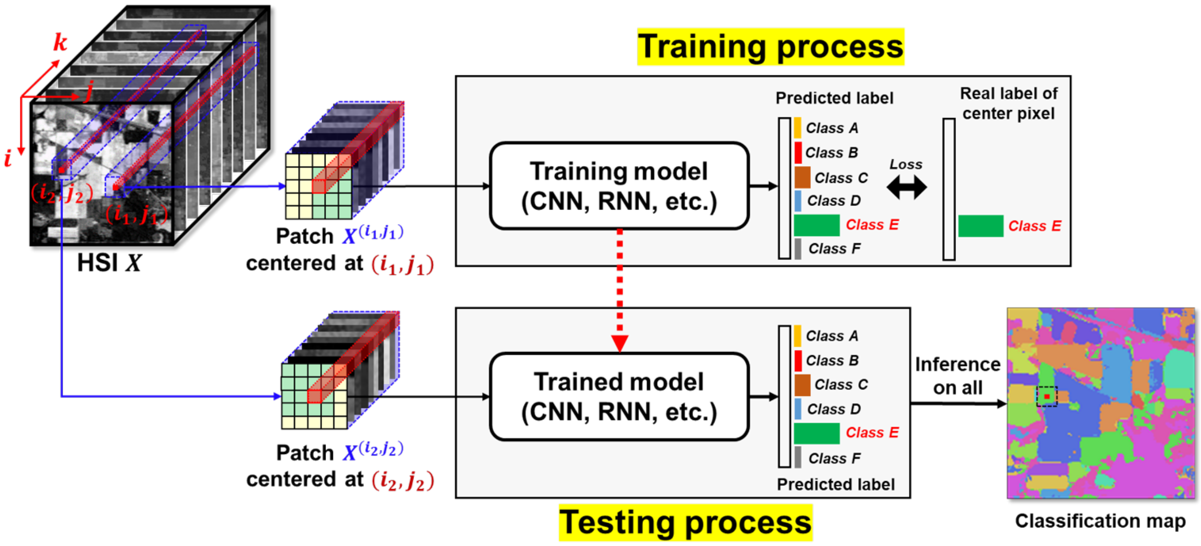


Figure 1.4 The general flowchart of patch-wise learning framework for HSI classification task.

from the HSI without manual feature engineering. They can also manage the high dimensionality of hyperspectral data by using the spatial and spectral links within the data [28]. With their advanced structures and optimization techniques, deep learning models can identify complex patterns and nonlinear relationships. It leads to better classification accuracy compared to traditional methods [93]. However, they often need much-labeled training data and can be computationally intensive [83]. Despite this, their potential for accurate HSI classification makes them an essential area for further study. Therefore, developing a quick, efficient, and accurate deep learning-based HSI classifier is now a primary focus in this field.

Due to the two assumptions that 1) the pixels close in the spectral feature space are highly likely to have the same label and 2) the spatially nearby pixels will be highly possible to share the same label, deep learning-based HSI classifiers mainly follow the patch-wise learning framework [87], in which a small neighborhood around the pixel (a ‘sub-image’) is cropped as a data sample for training model, as shown in Figure 1.4. The center pixel’s class is predicted based on the spectral characteristics of all pixels within the patch, thus incorporating local spatial information. It

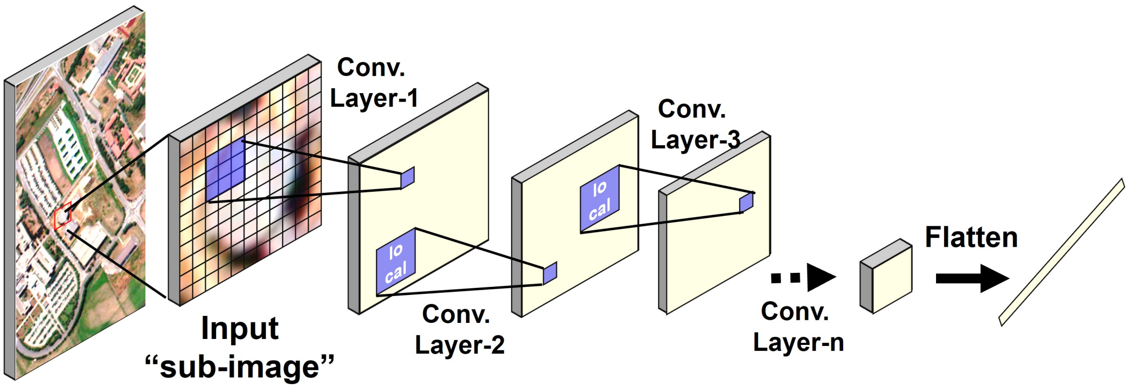


Figure 1.5 The general structure of multi-layer CNNs-based model. The input for the CNNs is a patch, sub-image, cropped from original HSI. The output generally is flattened into a feature vector.

allows the classifier to recognize spatial patterns (e.g., edges, textures) that can aid in distinguishing different land cover types. For example, in an urban area, a pixel representing a building might be surrounded by pixels representing streets, trees, or other buildings, which provide context for the central pixel.

CNN-Based Methods

In these models, CNNs are the most popular methods for dealing with image features, and they have been widely adopted in HSI classification research. The general structure of multi-layer CNNs-based model is shown in Figure 1.5.

Chen *et al.* [9] applied PCA to reduce the dimensionality of the HSI and then implemented a 2D-CNN to process the spatial information in a local patch. Li *et al.* [40] introduced a novel fully CNN (FCN) for HSI classification that uses convolution and deconvolution on the first principle component. Similarly, Lee *et al.* [39] proposed a deep contextual CNN that fully exploits the local contextual correlations by combining a 2D-CNN and an FCN into the framework.

Pushing the boundaries further, a simple 3D-CNN structure was first proposed for HSI classification by integrating spectral and spatial information [43]. Hamida *et al.* [5] introduced a new

3D deep learning approach that processes spectral and spatial information jointly. Luo *et al.* [46] proposed a 3D-CNN with a novel pixel sampling strategy and included a unique reshape layer for classification. He *et al.* [29] and Qi *et al.* [55] developed a multi-scale 3D-CNN to learn 2D multi-scale spatial features and 1D spectral features jointly.

Sun *et al.* [65] proposed the combination of a 3D-CNN and attention. Similarly, Hang *et al.* [22] proposed an attention-aided 3D-CNN model focusing on more discriminative channels and positions. Furthermore, Ghaderizadeh *et al.* [18] proposed hybrid 3D and 2D CNNs to learn spatial-spectral features together.

RNN-Based Methods

Specifically, RNNs, popular deep learning models derived from the natural language processing (NLP) field, have also been adapted for the HSI classification task. Two kinds of RNNs-based models are shown in Figure 1.6 and 1.7, respectively.

Mou *et al.* [49] fed a pixel's spectral signal into an RNN, where the step length was set equal to the number of bands in the HSI data. Paoletti *et al.* [51] introduced a scalable RNN, simplifying the internal complexity of the original RNN model for HSI classification.

Zhou *et al.* [89] introduced a spatial RNN architecture where the first principle component of the patch is processed row by row. Each row is treated as a step for the RNN. Similarly, Hang *et al.* [23] divided a cropped HSI patch into several equal groups along the spectral domain, each serving as a step input for the RNN.

Zhang *et al.* [84] transformed an HSI patch into a single pixel-sequence by considering each pixel along with its spectral information as a separate step for the RNN. Additionally, Hao *et al.* [25] introduced a geometry-aware RNN for HSI classification, integrating U-Net [59], ResNets [27], and RNNs to build a model that is aware of the geometrical properties. Shi *et al.* [62] presented a hierarchical RNN architecture for HSI classification, incorporating a multi-scale CNN-

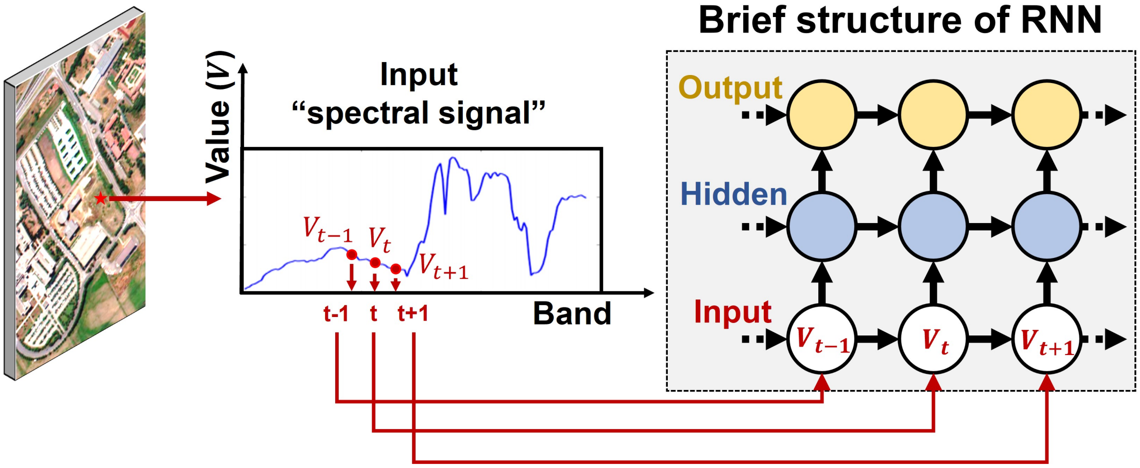


Figure 1.6 The general structure of spectral-only RNN. The input for RNN is defined as pixel's spectral signal. Each spectral value at each band index will be recurred by RNN.

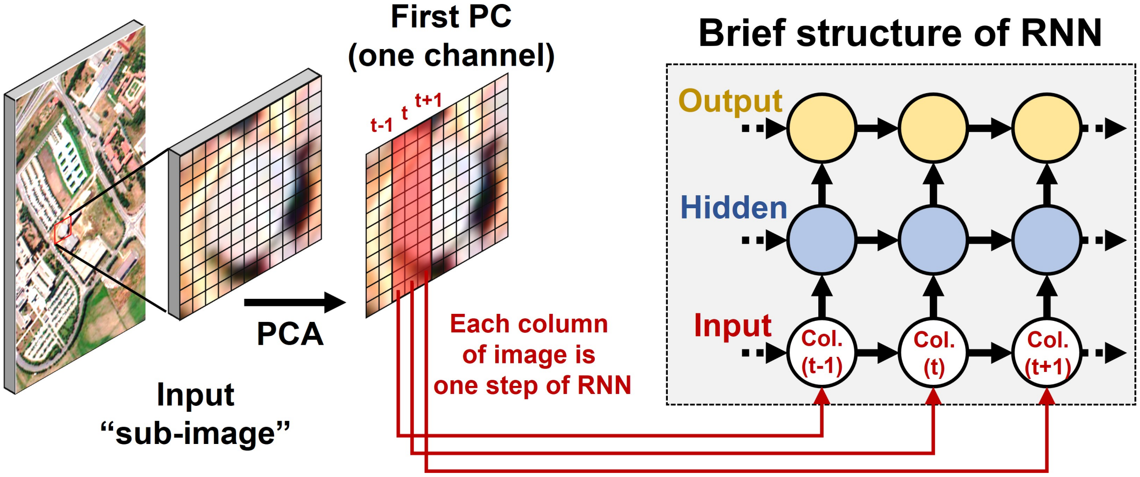


Figure 1.7 The general structure of spatial-only RNN. The input for RNN is defined as first PC. Each column or row of first PC will be recurred by RNN.

based approach.

1.3 Existing Problems

Despite previous works' achievements, they present a few significant limitations.

A common problem is the limited quantity or insufficiency of labeled training samples in HSI classification, which stems from the time-consuming and costly nature of data labeling manually, resulting in a training set that lacks the sense of sufficient multi-directional features. In such cases, the model tends to learn limited samples with fixed spatial orientations in the training data, erroneously treating these limited perspectives as comprehensive. It leads to severe model overfitting, where the model learns the training data exceedingly well and fails to generalize to new, unseen testing data.

More specifically, previous CNN-based methods locally operate the un-directional convolution on the input image. Hence, it suffers the following shortcomings: 1) Lack of rotational invariance: CNNs trained on unidirectional features might not recognize the same pattern if it appears in a different orientation in the test data, as shown in Figure 1.8; 2) Missing complex spatial relationships: Some spatial characteristics may only become evident when viewed from a certain angle. It could affect the CNN's ability to recognize meaningful patterns and features; 3) Difficulty in identifying edge and texture information: Important features like edges or textures, which might change across different orientations, may not be captured well with unidirectional features; 4) Poor generalization: If the CNNs are trained only on unidirectional features, it might not generalize well to different directional patterns in the test data; 5) Gridding Problem: In the context of dilated convolutions, using unidirectional features could lead to gridding artifacts which could negatively impact the model's performance; 6) Spatial reliance: CNNs generally convolves the image on the spatial aspect, the details of spectral variations may be neglected.

Furthermore, previous RNN-based methods deploy the single-directional feature for HSI classification. Also, they face the following problems: 1) Sequence Dependence: RNNs process sequences in a specific direction, i.e., they are sensitive to the order of data. Some important information that exists in other directions might be missed when using single-directional features; 2) Limited Context: RNNs using one-directional features are restricted to a specific perspective and

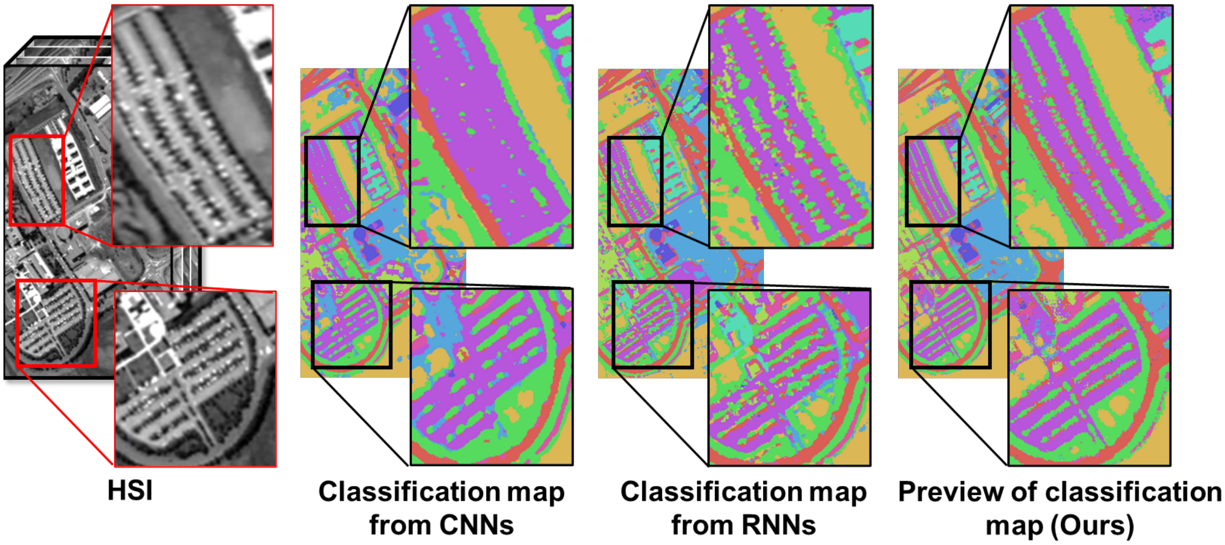

HSI
Classification map from CNNs
Classification map from RNNs
Preview of classification map (Ours)

Figure 1.8 Demonstrating the classification results from CNNs [29] and RNNs [84] for HSI classification, it shows unsatisfying performance, particularly at the edges or textures, when there is variation in the spatial orientation. Our method essentially handles this issue.

hence might fail to capture a broader, more holistic context of the spatial-spectral dependencies in the HSI; 3) Generalization issues: Similar to CNNs, RNNs trained only on single-directional features might have issues when trying to generalize to unseen data with different directional patterns; 4) Temporal Limitations: In an RNN, the influence of a given input on the hidden layer is shrunk or diluted over time (also known as the 'vanishing gradient problem'). Suppose a key feature is too far back in the input sequence (the 'wrong' direction from the current perspective). In that case, it may not significantly influence the output; 5) Spectral reliance: The process of RNNs handles images pixel-by-pixel, which may emphasize the spectral variations and result in noise-like classification, as shown in Figure 1.8.

To tackle these challenges, we could consider designing models accommodating a broad range of spatial complexities. It could involve developing models with larger receptive fields to capture more extensive spatial dependencies. Crucially, we should consider incorporating multi-directional features to capture spatial dependencies from various perspectives. We can enhance the model's

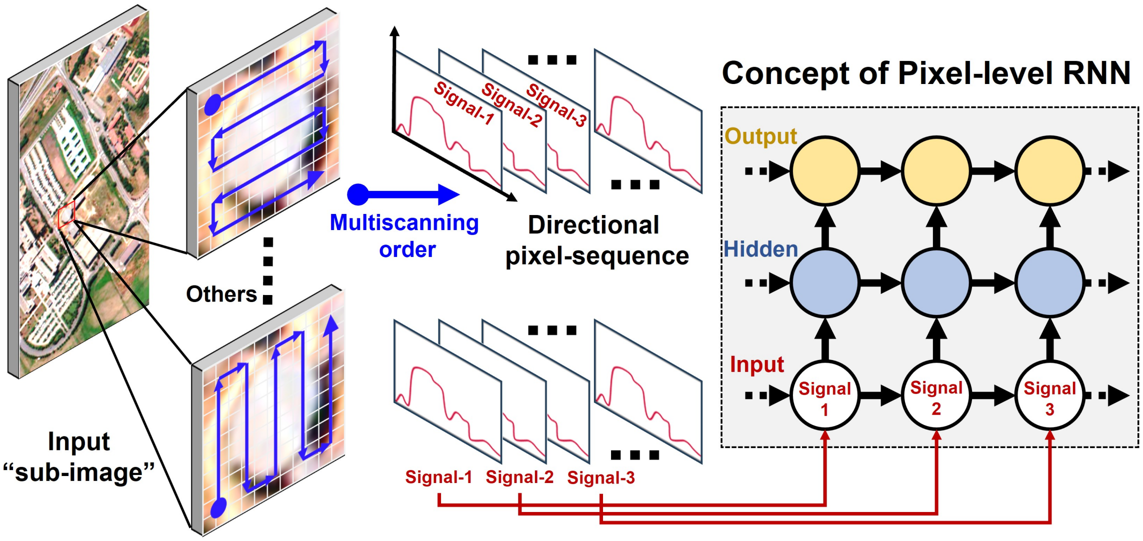


Figure 1.9 Illustration of the concept of multiscanning-based RNNs for HSI classification. The diagram demonstrates how features from multiple scanning strategies provide a comprehensive, multi-directional view of the spatial complexities inherent in the HSI patch, leading to a more informative, potential and knowledgeable feature.

ability to interpret diverse spatial patterns in unseen test data by exposing the models to a broader array of spatial complexities, including those from different directions.

1.4 Core Concept of Solution

In the realm of HSI classification, effectively capturing the diverse spatial complexities is a significant challenge. Previous methods using CNNs and RNNs often utilize a single scan direction, resulting in a limited viewpoint and consequently restricting their capacity to interpret spatial patterns accurately.

Therefore, our concept revolves around integrating multi-directional features with RNNs for HSI classification, as shown in Figure 1.9. This strategy, named 'multiscanning,' broadens the model's ability to understand and represent spatial dependencies, offering multiple perspectives and capturing a more comprehensive range of patterns and structures in the data.

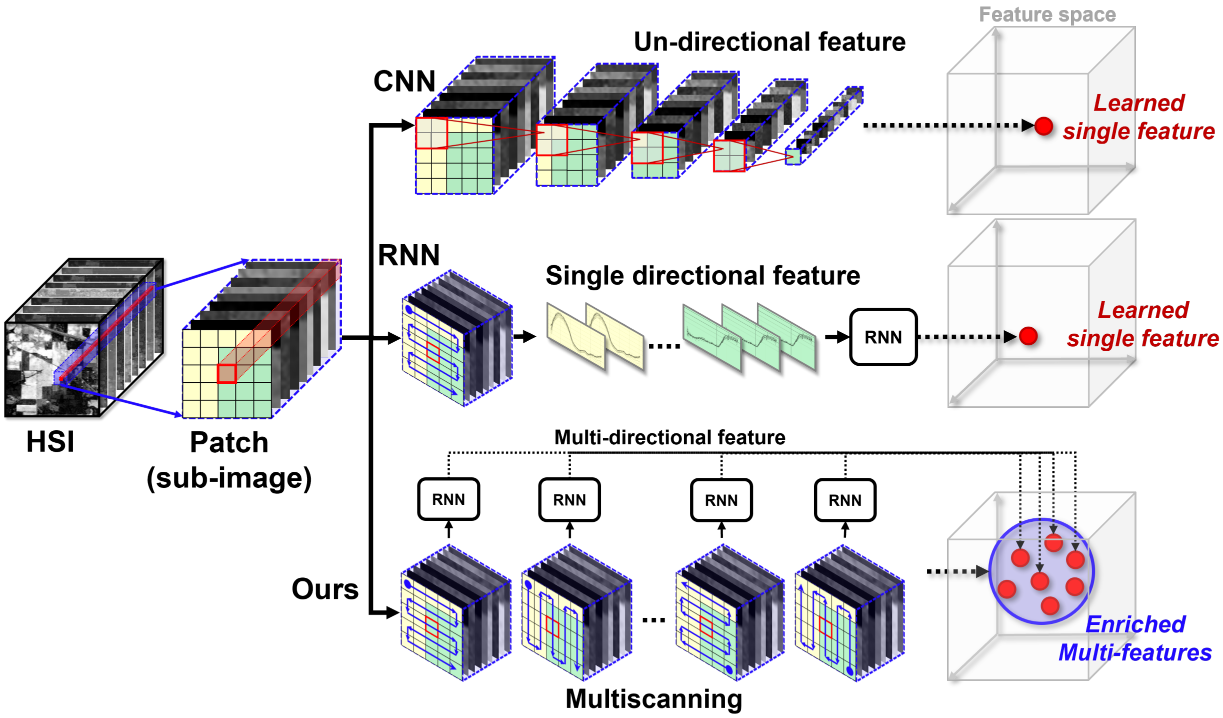


Figure 1.10 Depiction of the impact of multiscanning with RNNs in enriching the feature pool for potential feature selection, enhancing the generative ability of the model.

Specifically, we propose a design where we train RNNs with enlarged receptive fields. It enables the RNNs to incorporate a broader spatial context. However, what distinguishes our approach is the addition of multi-directional features derived from multiple scanning strategies.

Multiple scanning strategies allow us to extract features from the data in various directions and perspectives. It is akin to viewing a 3D object from multiple sides, resulting in a more holistic understanding of its structure and features. These multi-directional features are then fed into the RNNs, capable of processing sequential data, thus enabling the understanding of complex spatial situations.

The proposed strategy significantly enriches the feature set or space that the model learns from, thereby enhancing its ability to understand and interpret unseen test data, as shown in Figure 1.10. It also allows our model to be more adaptable and has the potential for different data distributions and spatial complexities that may be present in the HSI data.

In essence, our core concept involves leveraging the strengths of RNNs in processing sequential data and supplementing them with a broader view of the data via multi-directional scanning strategies; this combination provides a more robust and generalized model for HSI classification tasks, pushing the current achievable boundaries.

1.5 Organization of Dissertation

We summarize the succeeding chapters, conceptually describing their contents and contributions. It mainly discusses the 1) Multiscanning-based RNN for HSI classification in Chapter 2, 2) HSI classification using multiscanning-based RNN with attention in Chapter 3, and 3) HSI classification using multiscanning-based RNN-Transformer in Chapter 4. Figure 1.11 illustrates the organization of this dissertation and its corresponding preliminary work.

1.5.1 Multiscanning-Based RNN for HSI Classification

Chapter 2 details the concept of proposing the multiscanning strategy with RNNs.

It begins with a discussion of the limitations of CNN-based methods. We review RNNs and the limited scanning approaches that convert an image into a pixel-sequence. Subsequently, we introduce the observation of scanning manners and examine potential scanning patterns prevalent in the research community. Each pattern is further expanded into multiple orderings to the greatest extent possible, delineating the specific ordering directions later. Two fundamental questions are tackled: 1) How can an image patch be scanned into sequential data? and 2) What is the appropriate number of scanned pixel-sequence to utilize?

The key contributions of this chapter are briefly listed as follows:

1) We propose and attempt the multiscanning strategy with RNNs for HSI classification.

2) We discuss the scanning characteristics and optimize the number of scanning patterns with local scanning size for HSI classification.

3) We verify the possibility of using multiscanning strategy and RNNs for HSI classification and obtain satisfying performance, paving the road for further research.

4) The multiscanning strategy with RNNs is highlighted as a pioneering approach in the application of HSI classification.

The details of contributions are listed in Section 2.8.1.

1.5.2 HSI Classification Using Multiscanning-Based RNN with Attention

Chapter 3 presents the further research featured in multiscanning-based RNNs and attention mechanism for HSI classification.

Expanding on the idea of a multiscanning strategy, we refine its integration with RNNs by devising a harmonious approach. We further explore the spectral-spatial structure of HSI and how it interacts with RNNs. The detailed implementation of RNN for multiscanning sequences is outlined subsequently.

Besides, we propose various methods of feature fusion, including simple summation, length-level concatenation, and feature-level concatenation, to effectively incorporate the complementary features. Following this, we design integrated networks in two schemes. The first scheme feeds multiple scanned pixel-sequences individually into the RNN, while the second scheme pairs inverse pixel-sequences for a bidirectional RNN. The latter approach broadens the receptive fields, thus making our model more concise and compelling.

The experimental analysis focuses on several aspects, including 1) Evaluating the classification performance for each land-cover category and generating the overall classification map; 2) Investigating the outcomes of employing RNNs, specifically the Gated Recurrent Unit (GRU) and

Long Short-Term Memory (LSTM), in different scanning patterns; 3) Assessing the effectiveness of different scanning patterns; 4) Analyzing the performance when combining different scanning patterns; 5) Examining the impact of different feature combination methods; 6) Studying the effect of varying the number of scanned pixel-sequences; 7) Investigating the implications of using an attention mechanism; 8) Comparing the performance of RNN and 1D-CNN models for handling pixel-sequences. The major contributions of this chapter are briefly listed below:

1) To further improve the multiscanning-based RNNs featured in Chapter 2, we propose to validate the effectiveness of two approaches: ’Scanning Order-Based Attention’ and ’RNN-Based Multiscanning Feature Fusion’. These methods build upon the research presented in Chapter 2 and achieve improved results, with enhancements ranging from 2% to 5%.

2) Compared to other baseline methods, our approach achieves an overall accuracy improvement of 8% to 25%. Against state-of-the-art methods, we observe improvements ranging from 1% to 5%. Meanwhile, our approach can save approximately 2% to 62% in processing time, and our model size is significantly smaller—between 2 to 20 times lighter. This demonstrates the efficiency of our method.

3) This chapter delves deeper into the research on multiscanning-based RNNs, focusing on the attention mechanism and feature fusion. It demonstrates the potential for enhancing HSI classification results.

The details of contributions are listed in Section 3.7.1.

1.5.3 HSI Classification Using Multiscanning-Based RNN-Transformer

Chapter 4 extends the research into multiscanning-based RNN-Transformer for HSI classification.

This study introduces the renowned Transformer model into our study. We begin the chapter by addressing issues arising from using RNNs in HSI classification. It is followed by a brief overview of the capabilities of the Transformer model, focusing on its self-attention mechanism.

Next, we propose a novel RNN-Transformer encoder that combines the ordering bias of RNNs with the self-attention weights of the Transformer to generate features. This integration leads to the development of a pixel-oriented 'Spectral Transformer,' which effectively leverages the strengths of both models while mitigating their respective limitations. Additionally, we introduce a spectral-spatial-based soft masked self-attention mechanism to the encoder, enhancing attention allocation. This work departs from our previous endeavors and introduces a novel method for feature combination named the 'Multiscanning Transformer.'

The experimental analysis includes 1) Evaluation of classification performance for each land-cover category and generation of the overall classification map; 2) Analysis of the balance weight in positional embedding, exploring its influence on the performance; 3) Visualization of attention maps, providing insights into the attention allocation within the models; 4) Examination of the impact of the initial patch size on performance, determining the optimal patch size for the task; 5) Investigation of the number of training samples on the model's performance and generalization capabilities; 6) Comparative analysis of the proposed Spectral Transformer and Multiscanning Transformer, highlighting their respective strengths and contributions.

In addition, several critical ablation studies are conducted, which include the following aspects: 1) Capacity exploration with a multiscanning strategy, investigating the impact of utilizing multiple scanning strategies on the model's capacity and performance; 2) Analysis of the effects of spectral-spatial-based soft masked self-attention; 3) Comparison of different positional embedding methods, exploring the impact of different approaches to positional encoding on the model's performance; 4) Design considerations for the RNN-Transformer, analyzing the specific architectural choices that contribute to the overall performance.

The major contributions of this chapter are briefly listed below:

1) In this chapter, our goal is to build upon the research presented in Chapter 3 by introducing approaches such as the ’Spectral Transformer’ and ’Multiscanning Transformer.’ The experimental results validate their effectiveness. When compared to the findings in Chapter 3, we observe significant improvements, ranging from 2% to 4%..

2) Across four datasets, we achieve overall accuracy improvements of 6% to 11% compared to baseline methods, specifically Transformer-based approaches. Against state-of-the-art methods, our improvements range from 2% to 5%. Meanwhile, our approach can save approximately 50% in processing time and reduce the model size by 40%, demonstrating the significant efficiency of our method.

3) In this chapter, we introduce several innovative proposals, including the ’Spectral Transformer’ and ’Multiscanning Transformer,’ among others, all of which incorporate a multiscanning strategy. We also explore and verify their potential to enhance HSI classification results.

4) In this study, we reassess the effectiveness of employing the RNNs and Transformers (i.e., pure sequential models) for HSI classification tasks. While their integration has not been deeply explored in this community, we aim to contribute to advancing related studies with our research.

The details of contributions are listed in Section 4.9.1.

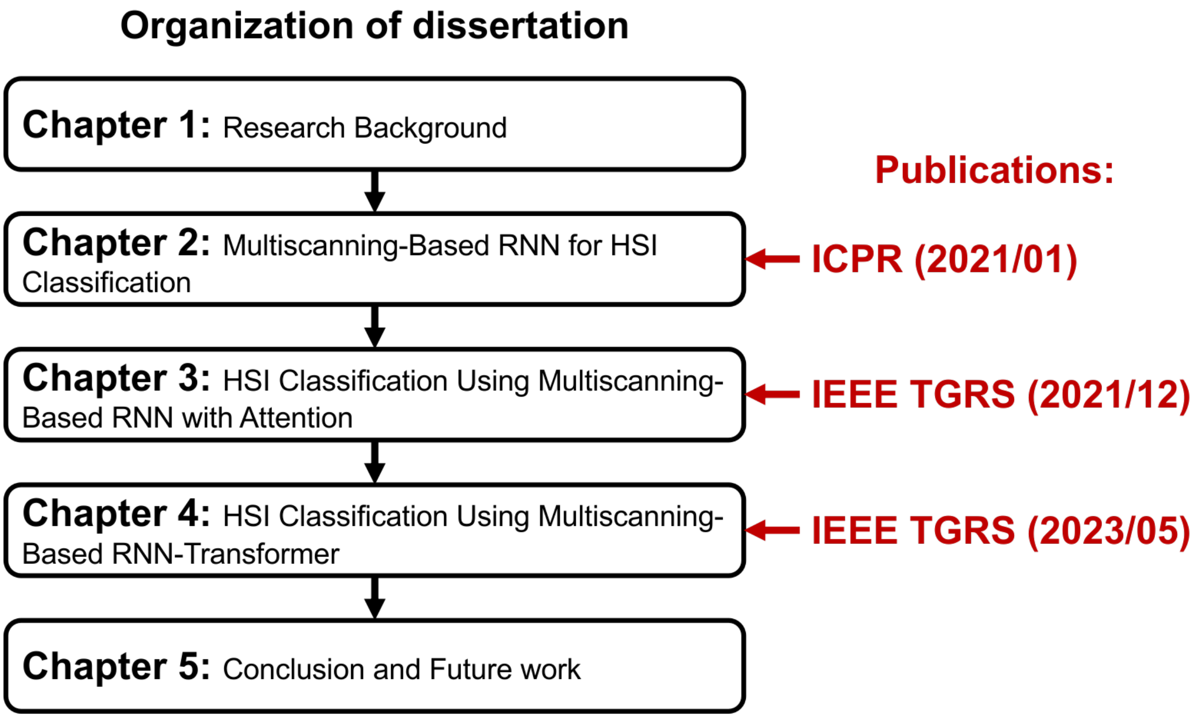

Organization of dissertation
Chapter 1: Research Background
Chapter 2: Multiscanning-Based RNN for HSI Classification
Chapter 3: HSI Classification Using Multiscanning-Based RNN with Attention
Chapter 4: HSI Classification Using Multiscanning-Based RNN-Transformer
Chapter 5: Conclusion and Future work
Publications:
ICPR (2021/01)
IEEE TGRS (2021/12)
IEEE TGRS (2023/05)

Figure 1.11 Organization of this dissertation and its correspondent publications.

Chapter 2

Multiscanning-Based RNN for Hyperspectral Image Classification

This chapter outlines the scenario of attempting the novel multiscanning strategy with RNN for HSI classification.

It starts by pointing out the limitations of using CNNs for HSI classification, which subsequently motivates the implementation of RNNs for HSI classification. It then reviews the current scanning methods employed for RNNs. From these foundations, the chapter transitions into a detailed discussion of our specific observations that inform the design of the multiscanning strategy. The discussion evolves progressively throughout each section, building the context and framework for the strategy.

2.1 Background

CNN-based methods have widespread use in HSI classification tasks owing to their effectiveness in local feature extraction. However, they suffer from certain limitations. The local contextual features they extract depend on small filter kernel sizes (commonly 3×3), which inhibit learning the spatial dependence between nonadjacent pixels in an image patch. For instance, regions at opposite corners of an image patch often do not communicate effectively during the standard convolution process [62]. It is a significant drawback since spatial dependencies are critical to interpreting spatial structural information [20, 63, 96].

Primarily, CNNs use unidirectional convolution operations that focus on local features from a single viewpoint. It means CNNs can overlook critical spatial dependencies and information in other directions. Simply put, CNNs lack the capacity for multi-directional feature learning due to their inherent unidirectional convolution mechanism. This limitation can hinder the network's understanding of HSI's complex spatial structures and patterns.

Hence, it naturally raises the following questions:

1) What methods can be employed to capture the global spatial dependencies among all pixels in an image patch, going beyond the limitations of local features?

2) How can we transform the image patch or feature map to a one-dimensional format without compromising their spatial continuity?

3) Are there alternative mechanisms besides conventional convolution that could efficiently capture and integrate spectral-spatial features from HSI?

In response to these questions, some studies [23, 49, 62, 84, 89], originally hailing from the natural language processing field, have sought to employ RNNs. These networks can capture global dependencies within an input sequence and are compatible with one-dimensional data structures. It makes RNNs viable alternatives to CNNs if adapted to handle image structures.

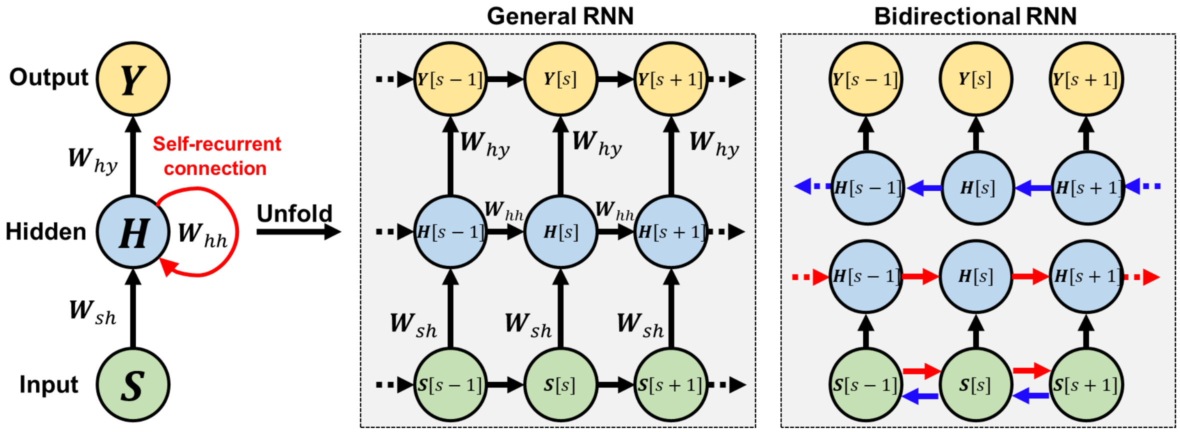


Figure 2.1 The illustration of RNN's operation with its variant, bidirectional RNN.

2.2 RNN

RNNs, a popular deep neural network, are highly effective in sequence learning [7, 37, 60]. They have a 'memory' function, which can recall the information from past states and apply it to future state learning, as illustrated in Figure 2.1. Given a sequential data $\mathbf{S}$ of length n, where each $\mathbf{S}[s]$, $s = 0, 1, ..., n$, signifies either a scalar or a vector. The hidden state $\mathbf{H}$, which denotes a sequential feature of identical length n derived from the RNN, can be computed as follows:

$$\mathbf{H}[s] = \begin{cases} 0, & \text{if } s = 0 \\ \mathcal{F}\Big(\mathbf{H}[s-1], \mathbf{S}[s]\Big), & \text{otherwise,} \end{cases} \tag{2.1}$$

where $\mathcal{F}()$ is a relevant function.

The update rule mentioned in $\mathcal{F}()$ is typically carried out as follows:

$$\mathbf{H}[s] = \phi\Big(\mathbf{W}_{sh} \odot \mathbf{S}[s] + \mathbf{W}_{hh} \odot \mathbf{H}[s-1] + \mathbf{b}_h\Big), \tag{2.2}$$

where $\mathbf{W}_{sh}$, $\mathbf{W}_{hh}$ and $\mathbf{b}_h$ represent the relevant input-to-hidden transformation parameters, respectively. ϕ is generally set as the non-linear activation functions, ReLU or Tanh. $\odot$ represents the matrix multiplication or dot-product.

Subsequently, the output from the last hidden state is obtained as:

$$\mathbf{Y}[s] = \phi\Big(\mathbf{W}_{hy} \odot \mathbf{H}[s] + \mathbf{b}_y\Big), \tag{2.3}$$

where $\mathbf{W}_{hy}$, and $\mathbf{b}_y$ represent the relevant hidden-to-output transformation parameters, respectively.

RNNs uniquely process each feature activation in their output about a specific position within the 'global' input, unlike traditional convolution+pooling layers that operate within a 'local' context window [75]. It allows RNNs to encompass the 'global' information from the input in a single, shallow feature extraction layer, resulting in fewer parameters than the deep structures and extensive feature extraction layers that CNNs need to capture similar 'global' information [53].

2.2.1 Long Short-Term Memory (LSTM)

The LSTM model addresses the shortcomings of basic RNNs, specifically the issues surrounding modeling long-term dependencies. The LSTM introduces a ‘memory block’ that replaces the recurrent hidden node in traditional RNNs, leading to notable performance improvements in various applications [31]. This memory block comprises an input gate, a forget gate, an output gate, and a self-recurrent connection. Information across current and previous time steps is stored within the memory cell, while the three gates control the inflow of information within the network. The computational procedure of LSTM at a time step s is outlined as follows:

$$\mathbf{I}[s] = \phi\Big(\mathbf{W}_{si} \odot \mathbf{S}[s] + \mathbf{W}_{hi} \odot \mathbf{H}[s-1] + \mathbf{b}_i\Big), \tag{2.4}$$

$$\mathbf{F}[s] = \phi\Big(\mathbf{W}_{sf} \odot \mathbf{S}[s] + \mathbf{W}_{hf} \odot \mathbf{H}[s-1] + \mathbf{b}_f\Big), \tag{2.5}$$

$$\mathbf{O}[s] = \phi\Big(\mathbf{W}_{so} \odot \mathbf{S}[s] + \mathbf{W}_{ho} \odot \mathbf{H}[s-1] + \mathbf{b}_o\Big), \tag{2.6}$$

$$\mathbf{M}[s] = \phi\Big(\mathbf{W}_{sm} \odot \mathbf{S}[s] + \mathbf{W}_{hm} \odot \mathbf{H}[s-1] + \mathbf{b}_m\Big), \tag{2.7}$$

$$\mathbf{C}[s] = \mathbf{F}[s] \times \mathbf{C}[s-1] + \mathbf{I}[s] \times \mathbf{G}[s], \tag{2.8}$$

$$\mathbf{H}[s] = \mathbf{O}[s] \times \phi\Big(\mathbf{C}[s]\Big), \tag{2.9}$$

where $\mathbf{W}_{si}, \mathbf{W}_{hi}, \mathbf{W}_{sf}, \mathbf{W}_{hf}, \mathbf{W}_{so}, \mathbf{W}_{ho}, \mathbf{W}_{sm}$, and $\mathbf{W}_{hm}$ are the relevant weight matrices. $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o$, and $\mathbf{b}_g$ are bias vectors. $\mathbf{S}[s], \mathbf{I}[s], \mathbf{F}[s], \mathbf{O}[s], \mathbf{M}[s], \mathbf{C}[s]$, and $\mathbf{H}[s]$ are the input, the input gate, the forget gate, the output gate, the candidate memory cell, the memory cell, and the hidden state at time step s, respectively. The operator × symbolizes element-wise multiplication or Hadamard-product. RNNs use the same model parameters across different time steps, so their parameter count does not increase with the number of time steps. They can be optimized using the Backpropagation Through Time algorithm [31]. Furthermore, LSTMs can also be extended into a bidirectional format to attain a more extensive receptive field.

2.3 RNN for Image with Scanning

However, conventional RNNs need to be better suited for image context learning due to the high dimensionality of image data, which can be either two-dimensional or three-dimensional. A major challenge when using RNNs for image data is finding a suitable sequence from the image. Consequently, various methods have been proposed to tackle this issue and effectively apply RNNs to image data.

ReNet [75] executes object recognition by substituting the convolution and pooling layers with four RNNs that traverse horizontally and vertically across the image in both directions. In [95], four directional scanning strategies were utilized: left-to-right, right-to-left, top-to-bottom, and bottom-to-top. Each direction was employed to scan an image or image region. In [94], four diagonal scanning methods were introduced in an acyclic manner, covering different diagonal directions to scan an image region. Zhou *et al.* [89] employed a scanning strategy where a single-channel image was scanned row by row in one direction (from top to bottom), with each row serving as one step input for the RNN. Zhang *et al.* [84] converted a 3D HSI patch into a pixel-sequence by calculating the Euclidean distance of each pixel to the central pixel. Each pixel, along with its spectral information, was considered as one step in the RNN. Moreover, Shi *et al.* [62] implemented diagonal, vertical, and horizontal scanning strategies on an image patch. The outputs from the RNNs were then aggregated through a general summation operation.

As previously discussed, an approach exemplified by [89] entails the contemplation and amalgamation of potential image-level scanning strategies (either in a column-to-column or row-to-row fashion) due to their inherent complementarity. However, these techniques usually focus on the first principal component of the HSI patch, thereby overlooking the crucial spectral information embedded within the original data.

A different method, like the one presented in [84], aims to determine an optimal sorting strategy at the pixel level (pixel by pixel). However, a solitary sorting strategy may not adequately encapsulate an image patch. Furthermore, as recognized widely, changing the input sequence for the RNN will result in divergent representations.

Moreover, the works [62, 94, 95] proposed to combine multiple scanning directions. However, these methods merely aggregate features from different directions without considering the correlation or weighting between them. It makes it challenging to discern which scanning direction offers superior or inferior results.

The strategies mentioned above adjust RNNs for image context learning, but they mainly focus on sequential data and overlook multi-directional features. While they offer many scanning methods, none explicitly explore the potential benefits of multi-directional feature learning. This omission is crucial as learning from multi-directional features could provide a broader context and richer information, potentially improving HSI classification performance.

Indeed, we need to tackle the following questions to maximize the potential of RNNs in HSI classification:

1) How can we augment the capabilities of RNNs to effectively process and understand image-

like features along with their spectral characteristics, mainly focusing on the spatial dependence among all pixels?

2) What would be the optimal strategy to transform or scan an image patch into a pixel-sequence that would retain as much contextual and spectral information as possible?

3) How can we best combine multiple pixel-sequences that may have different orderings? What strategy should we use to ensure a fair and meaningful combination of these sequences?

2.4 Observations

2.4.1 Scanning Orders Diversify Spatial Patterns of Sequences

Suppose we have an HSI represented by $\mathbf{X} \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times C}$, where $\mathcal{H}$, $\mathcal{W}$, and C correspond to height, width, and channels, respectively. Given a cropped HSI patch, denoted by $\mathbf{X}^{(i,j)} \in \mathbb{R}^{p\times p\times C}$, with a fixed spatial distribution in a $p \times p$ patch size - i, j signifying the spatial position (i, j) in the original $\mathbf{X}$.

It is crucial to convert $\mathbf{X}^{(i,j)}$ into a pixel-sequence $\mathbf{S}$ of size $\mathbb{R}^{p^2\times C}$ to work with the sequential model like RNNs. Here, p^2 denotes the step length for the RNNs, and C represents the feature dimension for each step, retaining spectral features.

However, no fixed and rigid rule dictates the arrangement of pixels in a pixel-sequence. Different arrangements can lead to distinct patterns in the pixel-sequence, capturing the spatial distribution of the underlying image patch. It is briefly illustrated in Figure 2.2, highlighting the potential for various pixel-sequences $\mathbf{S}_m^{(i,j)}$, $m = 1, 2, ..., \mathcal{M}$ to represent the spatial patterns of a patch.

For example, the first and second strategies start from the left-top pixel and end at the right-bottom pixel, implementing horizontal and vertical scanning, respectively. These distinct scanning

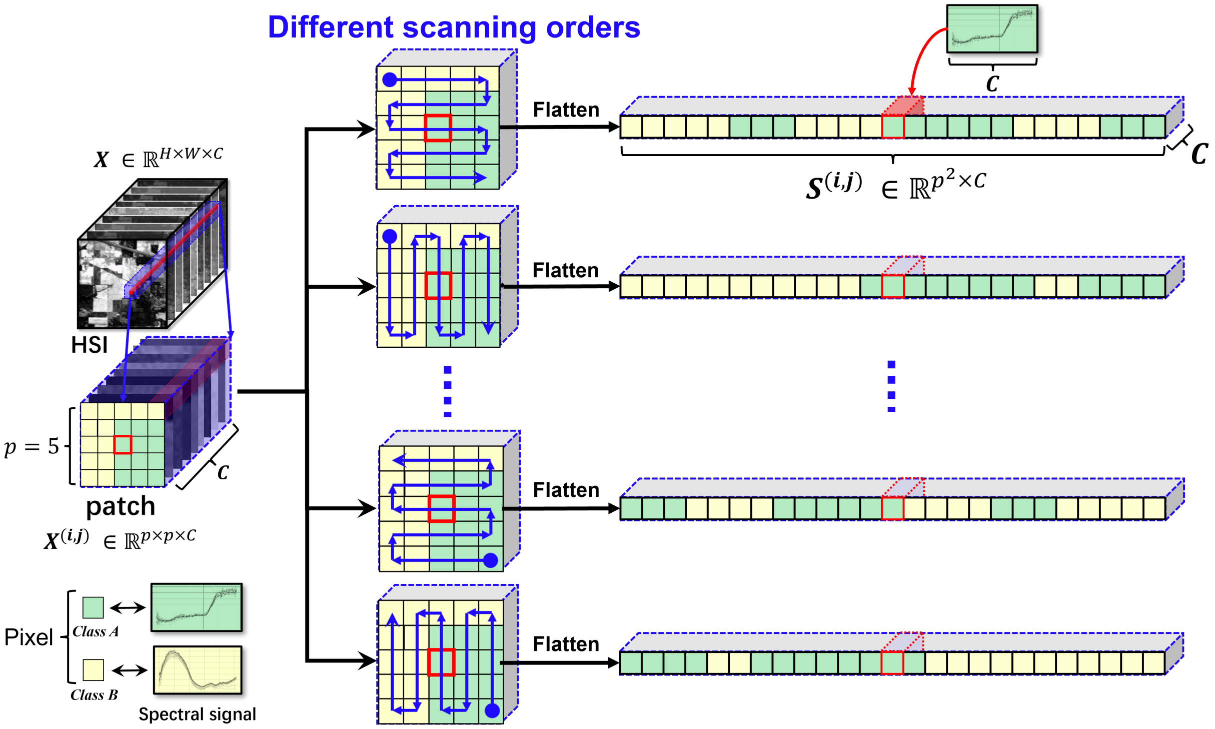

Different scanning orders
$X \in \mathbb{R}^{H\times W\times C}$
HSI
$p = 5$
C
patch
$X^{(i,j)} \in \mathbb{R}^{p\times p\times C}$
Pixel
Class A
Class B
Spectral signal
Flatten
Flatten
Flatten
Flatten
C
C
$S^{(i,j)} \in \mathbb{R}^{p^2\times C}$

Figure 2.2 A visualized explanation of the observation: Given a fixed spatial distribution of an HSI patch, different scanning orders will produce various spatial patterns within the pixel-sequences. These patterns can be interpreted as pixels of differing colors.

arrangements generate different patterns, both of which are valid options for representing the spatial information of the patch. Consequently, feeding these two sequences into the RNNs will result in two distinct representations for this image patch, as described in Eq 2.2.

The challenge arises in determining the scanning order. For instance, consider a patch size of $p = 5$. This results in the possibility of creating 25 permutations of 25 (P_{25}^{25}) different pixel-sequences by modifying the order of each pixel. However, many of these pixel-sequences are likely meaningless and redundant because they might fail to capture spatial distribution's continuity. Therefore, implementing restrictions on scanning manners becomes essential.

2.4.2 Multiple Scanning Orders Complement Feature Learning

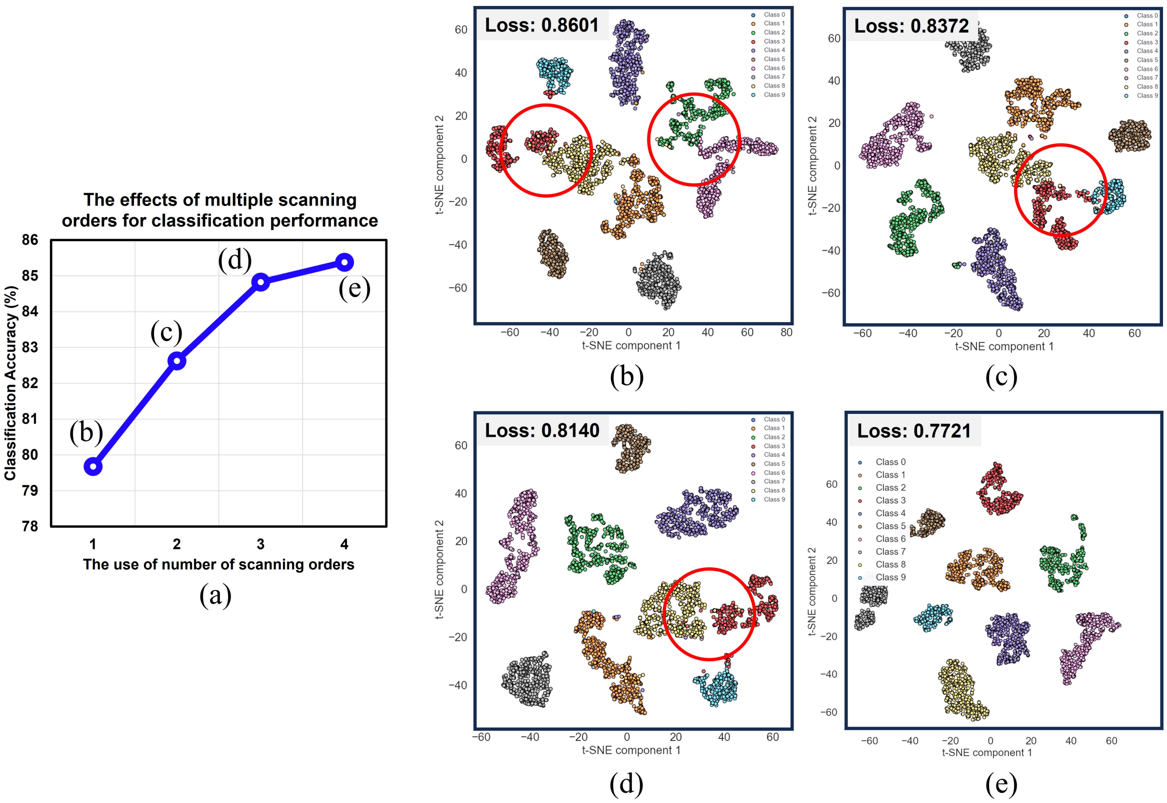
Multi-view learning [79, 85] has been proven effective in many machine learning tasks. It is beneficial when a single representation may only capture relevant or helpful information in the data. Hence, the advantages of using multi-view learning include: 1) Comprehensive Learning [86]: different views may capture different aspects or features of the data. The learning algorithm can get a more comprehensive understanding of the data by integrating these diverse views; 2) Learning efficiency: the learning algorithm may be able to converge faster or require fewer examples to achieve good performance by leveraging information from multiple views; 3) Improving performance: using multi-view learning can often achieve better performance, especially in scenarios where different views provide complementary information; 4) Noise reduction [42]: if one view of the data is noisy or contains errors, these can potentially be mitigated by other views that do not contain the same noise or errors.

Drawing inspiration from multi-view learning, we intuitively consider the application of multiple HSI patch views with multi-directional sense, employing different scanning orders with RNNs. Each unique sequence could capture various aspects or features of the image patch, such as distinct spatial dependencies or contextual information. Fusing these diverse perspectives can provide a more comprehensive representation of the image. Moreover, combining multiple sequences can enhance the model's robustness - should one sequence lead to inaccurate predictions, others may offset these errors when combined.

The multiscanning strategy serves as a novel form of feature augmentation for RNNs, improving their comprehension of the image from multi-directional views. This approach is also particularly suited to scenarios with limited HSI data samples.

To explore this concept, we conducted preliminary experiments to observe the classification performance of RNNs when we directly summed up one, two, three, and four scanning orders.

Our findings indicate that increasing the number of scanning orders on the PaviaU datasets



(a) (b) (c) (d) (e)

Figure 2.3 The preliminary experiments assessed the influence of various scanning orders on classification accuracy. (a) demonstrates the overall accuracy of the PaviaU dataset, indicating a general trend of improved performance with increased random scanning orders for RNNs. (b)-(e) show the T-SNE clustering of RNN outputs on the PaviaU dataset with one, two, three, and four scanning orders, respectively. With more scanning orders, the clusters become more cohesive and distinct. A consistent decrease in the losses of KL divergence also supports this observation.

leads to an overall improvement in classification accuracies. Additionally, we examine the output representations obtained by combining multiple scanned pixel-sequences using the T-SNE (t-Distributed Stochastic Neighbor Embedding) technique [72]. The results are further supported by the decrease in the total loss of Kullback-Leibler (KL) divergence [71], as illustrated in Figure 2.3. The visualization results demonstrate a general enhancement as the number of scanned pixel-sequences increases. This improvement is manifested by denser and more cohesive clusters in the T-SNE visualization. Thus, we posit that integrating multiple pixel-sequences enriches the vol-

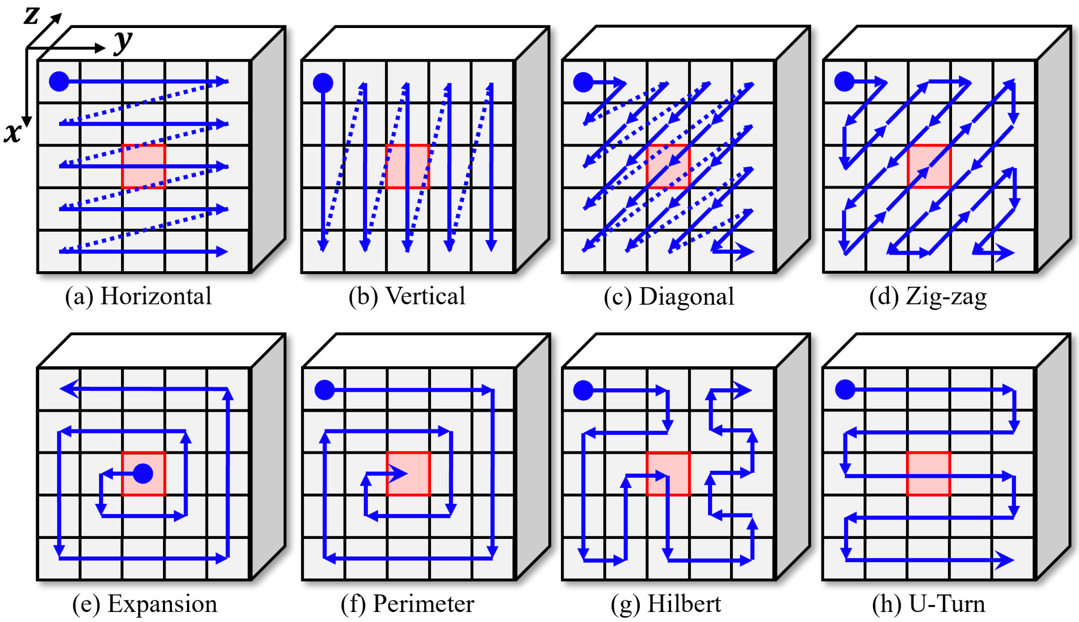


Figure 2.4 Single scanning patterns investigated include (a) horizontal, (b) vertical, (c) diagonal, (d) zig-zag, (e) expansion, (f) perimeter, (g) Hilbert, and (h) U-Turn. The dashed line represents the jump connection, while the center pixel is indicated in red.

ume of input information for the RNNs, leading to improved performance and more informative representations.

With these observations in mind, we are driven to generate different scanning pixel-sequences from a single HSI patch comprehensively and fittingly, thereby adaptively addressing various spatial dependencies.

2.5 Scanning Rules

We understand that a $p \times p$ image patch can be rearranged into many pixel-sequences. For example, when $p = 5$, there can be P_{25}^{25} pixel-sequences, each created by permuting the order of the 25 pixels. However, most of these pixel-sequences are redundant and meaningless because they do not capture spatial distribution continuity. It makes it essential to apply restrictions to scanning manners.

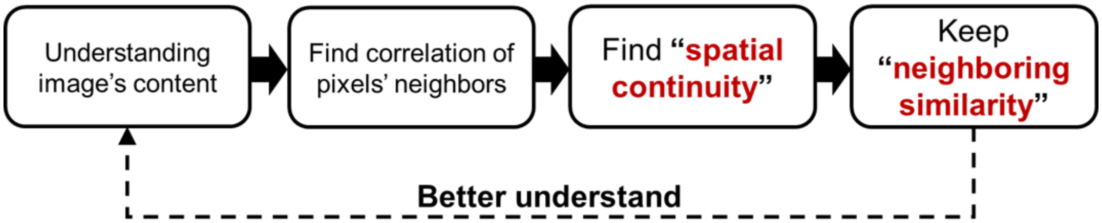

Understanding image's content
Find correlation of pixels' neighbors
Find "spatial continuity"
Keep "neighboring similarity"
Better understand

Figure 2.5 The illustration of understanding an image's content.

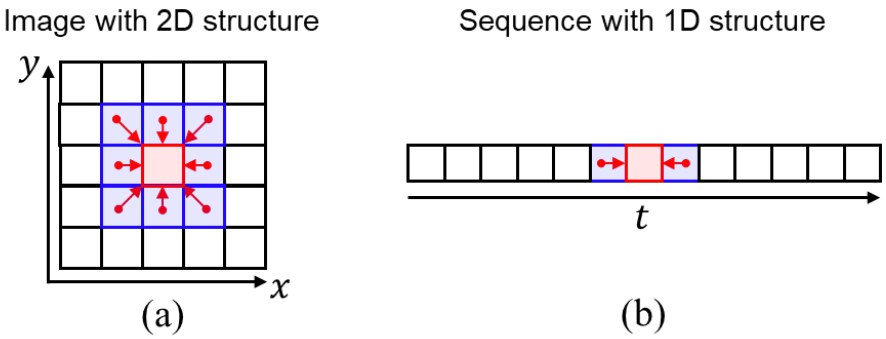


Figure 2.6 The illustration demonstrates that (a) in an image, one pixel can establish correlations with its eight nearest neighbors. Conversely, (b) in a sequence, a pixel (or a step, in sequence terms) can only correlate with two immediate neighbors - the ones on the left and right.

Hence, Figure 2.4 illustrates the possible scanning patterns by investigating the community.

Understanding an image's content requires identifying the close correlations between each pixel and its neighbors, as shown in Figure 2.5. This correlation can be analyzed through parameters like similarity. A high degree of similarity is typical for pixels within the same object or instance. Conversely, there might be a noticeable value difference at the edges of different objects or instances, reflected in sudden changes in similarity or entropy. In an image, one pixel can correlate with its eight closest pixels. However, for RNNs with a sequence, one pixel (i.e., one step) can only correlate with the two closest steps: left and right. This restriction implies that, compared to the image structure (2D structure), the sequence structure (1D structure) is more constrained in finding neighboring similarities that identify the data content's spatial continuity, as shown in Fig-

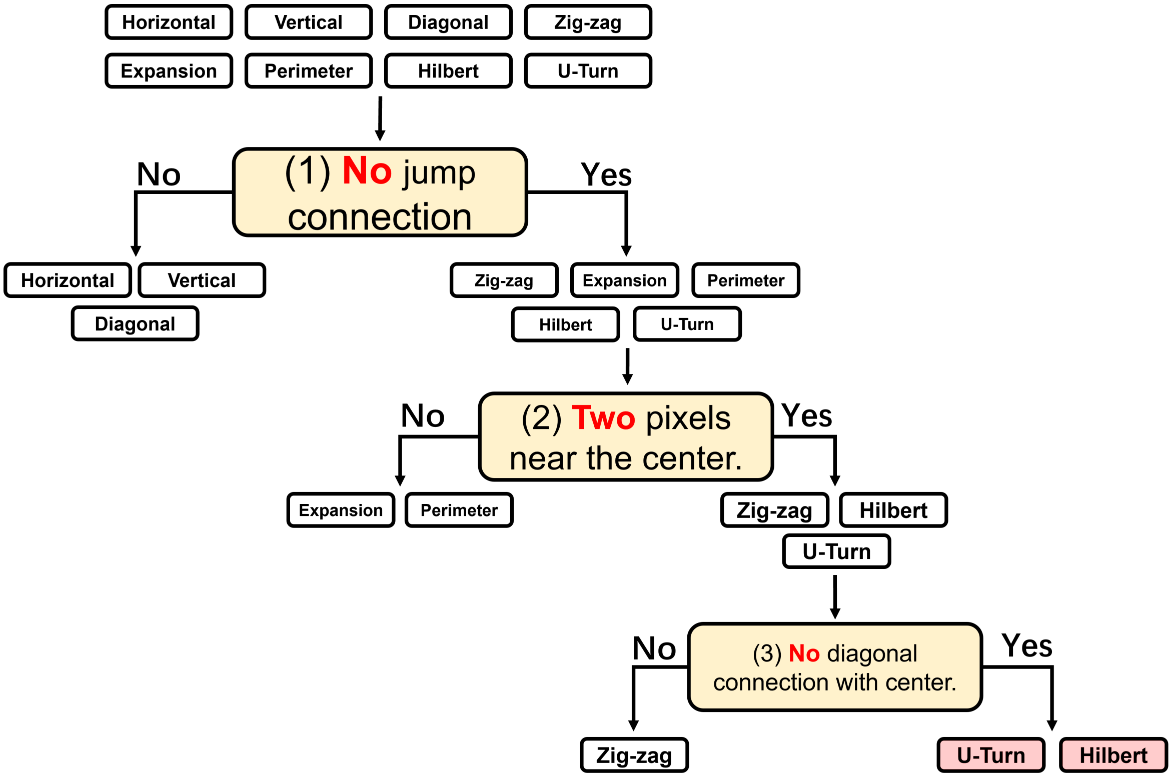


Figure 2.7 The screening procedure of scanning patterns.

ure 2.6. Considering these situations, we establish three rules for selecting the scanning patterns. Figure 2.7 illustrates the screening procedure.

Firstly, the scanned pixel-sequences must also be spatially continuous to maintain spatial continuity, incorporating as many adjacent neighbors as possible. Preserving spatial continuity enables us to capture and utilize the information from surrounding pixels since non-adjacent connections could disrupt the data's spatial coherence. Tapping into the data from neighboring pixels can help reduce intra-class variance and minimize the appearance of noise-like phenomena that affect classification performance [43]. We boost the quality and dependability of the extracted features by sustaining spatial continuity and integrating the information from neighboring pixels.

Secondly, in transforming a local image patch into a sequence, the sequence's central step can directly depend on the information from the preceding step and utilize the data from the subsequent

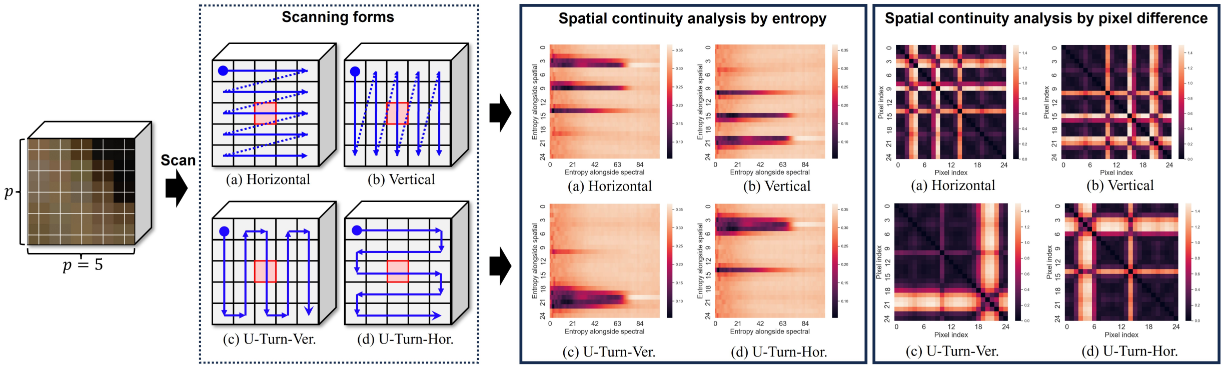


Figure 2.8 The spatial continuity of a scanned pixel-sequence, resulting from different scanning patterns of the same image patch, is analyzed by calculating element-wise entropy and pair-pixel difference. (a) represents horizontal scanning and its corresponding analysis results, (b) represents vertical scanning and its results, (c) shows the U-Turn-Vertical scanning and its outcomes, and (d) illustrates the U-Turn-Horizontal scanning and its results. Fewer abrupt changes in the results indicate spatial continuity. In this case, U-Turn scanning patterns appear more suitable for maintaining spatial continuity, resulting from no jump connections in the scanning pattern.

step. Viewing it from a different perspective, within an image patch, the central pixel can ascertain spatial dependencies not only from the nearest ‘left’ pixel but also from ‘right,’ ‘up,’ and ‘down’ adjacent pixels. Patch-based methodologies strive to classify the central pixel by considering its neighboring pixels. Therefore, when transforming an image into a sequence, it is essential to ensure that at least two neighboring pixels in the image’s location are positioned adjacent to the corresponding central pixel in the sequence.

The evaluation of spatial continuity can be visualized by calculating the difference between adjacent pixels or entropy. To illustrate this, we provide an example shown in Figure 2.8. The figure assumes a fixed spatial distribution for an image patch. This image patch is subjected to various scanning orders, each producing different ordered pixel-sequences. Following this, we calculate the entropy and pixel-difference along each scanned sequence. It becomes apparent that scanning patterns involving jump connections, such as Figures 2.8(a) and (b) result in significant inconsistencies in entropy along the spatial domain (y-axis)—noticeable by three instances of abrupt discontinuity. This phenomenon is also evident when calculating the pixel-pair difference, shown later. Conversely, if the scanning patterns are such that they spatially traverse the image patch, like in (c) and (d), the calculated entropy and pixel-difference tend to be smoother, with fewer abrupt changes.

Based on the three assumptions: 1) no jump connections should be included, 2) at least two nearest pixels should be arranged alongside the corresponding central pixel, and 3) no diagonal connections are with the center; only Hilbert scanning and U-Turn scanning can meet our rules.

For instance, let us consider the U-Turn scanning pattern. The scanning strategy is represented below. The original HSI symbolized as $\mathbf{X} \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times C}$ can be defined as follows:

$$\mathbf{X} = \left\{\mathbf{x}^{(i,j)} \in \mathbb{R}^{1\times1\times C}\right\}, \quad i = 0, 1, ..., \mathcal{H} - 1, \quad j = 0, 1, ..., \mathcal{W} - 1, \tag{2.10}$$

where $\mathbf{x}^{(i,j)}$ signifies the spectral signal at the spatial position (i, j). Consequently,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^{(0,0)} & \mathbf{x}^{(0,1)} & \dots & \mathbf{x}^{(0,W-1)} \\ \mathbf{x}^{(1,0)} & \mathbf{x}^{(1,1)} & \dots & \mathbf{x}^{(1,W-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}^{(H-1,0)} & \mathbf{x}^{(H-1,1)} & \dots & \mathbf{x}^{(H-1,W-1)}. \end{pmatrix}. \tag{2.11}$$

For one cropped HSI patch $\mathbf{X}^{(i,j)}$ centered at $\mathbf{x}^{(i,j)}$ with a patch size p, which is an odd number, as an illustration:

$$\mathbf{X}^{(i,j)} = \left\{\mathbf{x}^{(i+\alpha,j+\beta)}\right\}, \quad \alpha, \beta = -\frac{p-1}{2} \sim \frac{p-1}{2}, \tag{2.12}$$

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

Take $p = 5$ as an example:

$$\mathbf{X}^{(i,j)} = \begin{pmatrix} \mathbf{x}^{(i-2,j-2)} & \mathbf{x}^{(i-2,j-1)} & \mathbf{x}^{(i-2,j)} & \mathbf{x}^{(i-2,j+1)} & \mathbf{x}^{(i-2,j+2)} \\ \mathbf{x}^{(i-1,j-2)} & \mathbf{x}^{(i-1,j-1)} & \mathbf{x}^{(i-1,j)} & \mathbf{x}^{(i-1,j+1)} & \mathbf{x}^{(i-1,j+2)} \\ \mathbf{x}^{(i,j-2)} & \mathbf{x}^{(i,j-1)} & \mathbf{x}^{(i,j)} & \mathbf{x}^{(i,j+1)} & \mathbf{x}^{(i,j+2)} \\ \mathbf{x}^{(i+1,j-2)} & \mathbf{x}^{(i+1,j-1)} & \mathbf{x}^{(i+1,j)} & \mathbf{x}^{(i+1,j+1)} & \mathbf{x}^{(i+1,j+2)} \\ \mathbf{x}^{(i+2,j-2)} & \mathbf{x}^{(i+2,j-1)} & \mathbf{x}^{(i+2,j)} & \mathbf{x}^{(i+2,j+1)} & \mathbf{x}^{(i+2,j+2)} \end{pmatrix}, \tag{2.13}$$

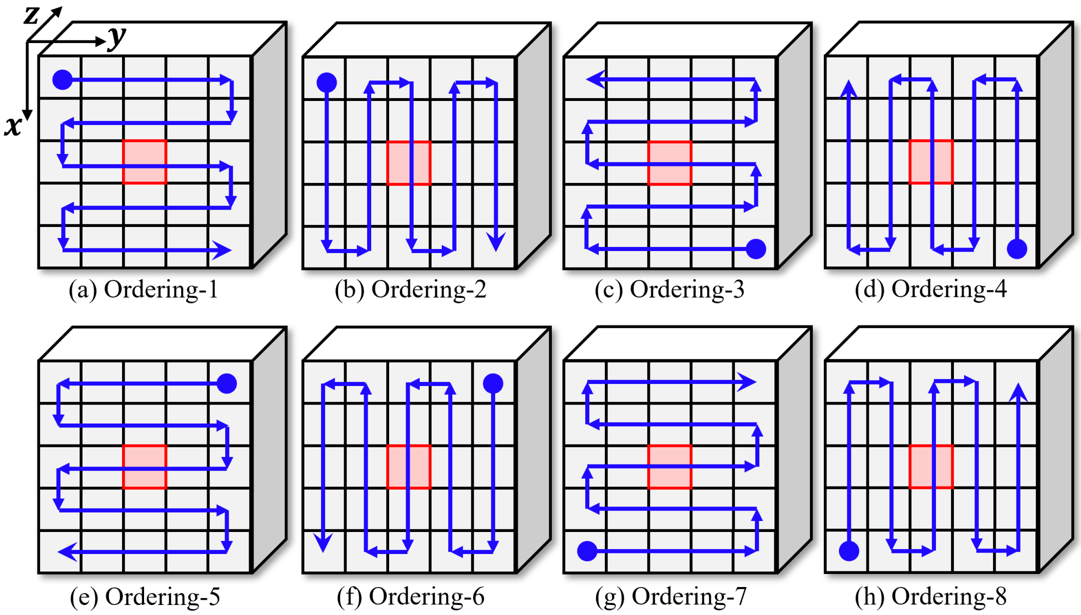


Figure 2.9 The illustration of extended multiscanning orders of the U-Turn scanning pattern on a 5×5 image patch.

where $2 \leq i \leq \mathcal{H}-3$, $2 \leq j \leq \mathcal{W}-3$ and $\mathbf{X}^{(i,j)} \in \mathbb{R}^{p \times p \times C}$. $\mathbf{X}^{(i,j)}[2,2] = \mathbf{x}^{(i,j)}$. Therefore, utilizing the multiscanning strategy, the HSI patch $\mathbf{X}^{(i,j)}$ is transformed into several pixel-sequences $\mathbf{S}_m^{(i,j)} \in \mathbb{R}^{p^2 \times C}, m = 1, 2, \ldots, \mathcal{M}$. Here, $\mathcal{M}$ represents the total count of scanning orders under one scanning pattern. For instance, let us consider U-Turn scanning with ordering-1, as illustrated in Figure 2.9(a), $\mathbf{S}_1^{(i,j)}$ could be arranged as follows:

$$\begin{array}{ccccccccccc}
 & \mathbf{x}^{(i-2,j-2)} & \rightarrow & \mathbf{x}^{(i-2,j-1)} & \rightarrow & \mathbf{x}^{(i-2,j)} & \rightarrow & \mathbf{x}^{(i-2,j+1)} & \rightarrow & \mathbf{x}^{(i-2,j+2)} \\
\rightarrow & \mathbf{x}^{(i-1,j+2)} & \rightarrow & \mathbf{x}^{(i-1,j+1)} & \rightarrow & \mathbf{x}^{(i-1,j)} & \rightarrow & \mathbf{x}^{(i-1,j-1)} & \rightarrow & \mathbf{x}^{(i-1,j-2)} \\
\rightarrow & \mathbf{x}^{(i,j-2)} & \rightarrow & \mathbf{x}^{(i,j-1)} & \rightarrow & \mathbf{x}^{(i,j)} & \rightarrow & \mathbf{x}^{(i,j+1)} & \rightarrow & \mathbf{x}^{(i,j+2)} \\
\rightarrow & \mathbf{x}^{(i+1,j+2)} & \rightarrow & \mathbf{x}^{(i+1,j+1)} & \rightarrow & \mathbf{x}^{(i+1,j)} & \rightarrow & \mathbf{x}^{(i+1,j-1)} & \rightarrow & \mathbf{x}^{(i+1,j-2)} \\
\rightarrow & \mathbf{x}^{(i+2,j-2)} & \rightarrow & \mathbf{x}^{(i+2,j-1)} & \rightarrow & \mathbf{x}^{(i+2,j)} & \rightarrow & \mathbf{x}^{(i+2,j+1)} & \rightarrow & \mathbf{x}^{(i+2,j+2)}
\end{array} \tag{2.14}$$

Thus, the pixel-sequence $\mathbf{S}_1^{(i,j)}$ can be created as follows:

$$\mathbf{S}_1^{(i,j)} = \left[\mathbf{x}^{(i-2,j-2)}, \mathbf{x}^{(i-2,j-1)}, \mathbf{x}^{(i-2,j)}, \mathbf{x}^{(i-2,j+1)}, \mathbf{x}^{(i-2,j+2)}, \mathbf{x}^{(i-1,j+2)}, \ldots\ldots, \mathbf{x}^{(i+2,j+2)}\right]^{\top}. \tag{2.15}$$

Table 2.1 Explored Horizontal, Vertical, Diagonal, and Zig-zag scanning patterns.

		scanning patterns							
		Horizontal		Vertical		Diagonal		Zig-zag	
Direction	Order	Start	End	Start	End	Start	End	Start	End
1	$(+x,+y) \rightarrow$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	-	-	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$
2	$(+x,+y) \downarrow$	-	-	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$
3	$(-x,-y) \leftarrow$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	-	-	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$
4	$(-x,-y) \uparrow$	-	-	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$
5	$(+x,-y) \leftarrow$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	-	-	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$
6	$(+x,-y) \downarrow$	-	-	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$
7	$(-x,+y) \rightarrow$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	-	-	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$
8	$(-x,+y) \uparrow$	-	-	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$

If $i, j = 2$ and $p = 5$, then $\mathbf{S}_1^{(2,2)}$ can be derived as:

$$\mathbf{S}_1^{(2,2)} = \left[\mathbf{x}^{(0,0)}, \mathbf{x}^{(0,1)}, \mathbf{x}^{(0,2)}, \mathbf{x}^{(0,3)}, \mathbf{x}^{(0,4)}, \mathbf{x}^{(1,4)}, ..., \mathbf{x}^{(4,4)}\right]^{\top}. \tag{2.16}$$

Additional scanning directions, given the same conditions ($i, j = 2$, and $p = 5$), are shown in Tables 2.1 and 2.2. In these tables, the 'Start' and 'End' pixels demonstrate each scanning pattern with various scanning orders. A '-' symbol marks an unexplored ordering scenario. The symbols ↑, ↓, ←, and → signify the initial direction of the scanning orders. For a more comprehensive understanding, it is recommended to cross-reference the U-Turn scanning pattern in Table 2.2 with Figure 2.9. Meanwhile, the total investigated scanning patterns are shown in Figure 2.10.

Upon further analysis and based on experience, it is evident that each scanning pattern can have up to eight scanning orders originating from the four corners of the image patch. The horizontal form has four scanning orders, the vertical form has four, the diagonal form has eight, the zig-zag form has eight, the expansion form has eight, the perimeter has eight, the Hilbert has eight, and the U-Turn has eight scanning orders.

Table 2.2 Explored Expansion, Perimeter, Hilbert, and U-Turn scanning patterns.

		scanning patterns							
		Expansion		Perimeter		Hilbert		U-Turn	
Direction	Order	Start	End	Start	End	Start	End	Start	End
1	$(+x,+y)\rightarrow$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{1\,(0,0)},\mathbf{x}^{2\,(4,0)}$	$\mathbf{x}^{1\,(0,4)},\mathbf{x}^{2\,(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$
2	$(+x,+y)\downarrow$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{1\,(0,0)},\mathbf{x}^{2\,(0,4)}$	$\mathbf{x}^{1\,(4,0)},\mathbf{x}^{2\,(4,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$
3	$(-x,-y)\leftarrow$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(2,2)}$	-	-	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$
4	$(-x,-y)\uparrow$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(2,2)}$	-	-	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,0)}$
5	$(+x,-y)\leftarrow$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$
6	$(+x,-y)\downarrow$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$
7	$(-x,+y)\rightarrow$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(4,4)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$
8	$(-x,+y)\uparrow$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(0,4)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(2,2)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,0)}$	$\mathbf{x}^{(4,0)}$	$\mathbf{x}^{(0,4)}$

2.6 Scanning Combination

Following the aforementioned guidelines, we generate eight valid U-Turn scanning orders to evaluate each neighboring pixel's impact on the central pixel. These can be viewed as encompassing scanning directions, each starting from the top-left, bottom-right, top-right, and bottom-left pixels of the original HSI patches.

A distinctive characteristic of this transformation is that the central pixel consistently maintains its position in the middle of the pixel-sequence, irrespective of the variations in the sequence orderings the patch undergoes. Consequently, it can extract significant spatial correlations from diverse orderings using identical time steps.

As depicted in Figure 2.2, varying the scanning order of a single image patch results in different patterns in the pixel-sequences. It indicates that a single image patch, with its spatial distribution fixed, can be represented in several ways. In CNN-based methods, one image patch is utilized as input, leading to a singular and fixed stream of information. However, converting one image patch

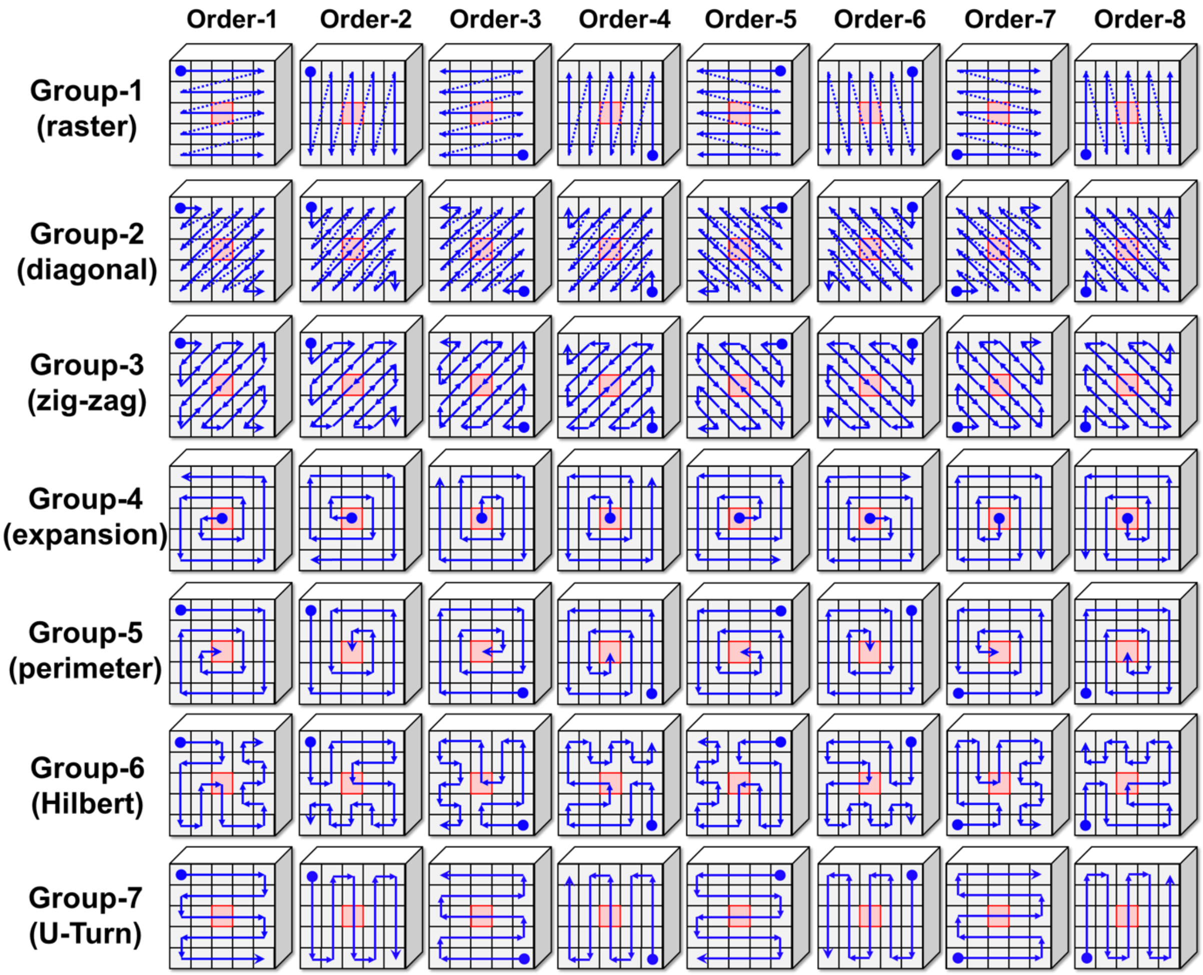


Figure 2.10 The illustration of all investigated scanning patterns. It can be divided into 7 scanning groups, and each group contains 8 scanning orders.

into multiple pixel-sequences allows these sequences to be introduced to RNN models. The multiscanning strategy effectively augments the quantity of sequential data, consequently enhancing the input information.

As noted in Figure 2.3, we discovered that randomly combining several pixel-sequences as input to RNNs produced more discriminative output representations. Therefore, the multiple generated pixel-sequences should be used comprehensively due to their complementarity for a single image patch.

However, it is intuitive that each pixel-sequence represents the original image patch in its unique way. While some might be suitable, others might not be. Therefore, scanning order-based attention among these scanning directions should be considered.

Furthermore, in the scanned pixel-sequences, we noticed that each sequence has a counterpart, such as ordering-1 and ordering-3; these pairs are reversed, as demonstrated in Figure 2.2. The forward and backward direction sequences are ideal for constructing a Bidirectional RNN (Bi- RNN), which provides more comprehensive reception steps and considers the entire sequence. Consequently, we group these paired sequences into four pairs for Bi-RNNs. The details are discussed in the upcoming Chapter 3.

2.7 Preliminary Experiments

We first assess the classification results of various scanning patterns to identify optimal scanning patterns for in-depth analysis. We perform individual experiments for each scanning pattern, maintaining a single scanning order, as depicted in Figure 2.4, and keeping all other conditions consistent with RNNs. These experiments are performed across three datasets, Indian Pines (IP), Pavia University (PU), and Salinas (SA), as illustrated in Figure 2.11. Table 2.3 outlines the number of training and testing samples used in each experiment.

2.7.1 Description of Datasets

The three publicly available HSI datasets are described below, and their corresponding original images, ground truths, and legends are depicted in Figure 2.11.

Indian Pines (IP) dataset: This dataset represents a mixed vegetation site and consists of 145 × 145 pixels with 220 spectral bands. After excluding water absorption bands, 200 bands are retained. The ground truth information contains 16 different land-cover classes.

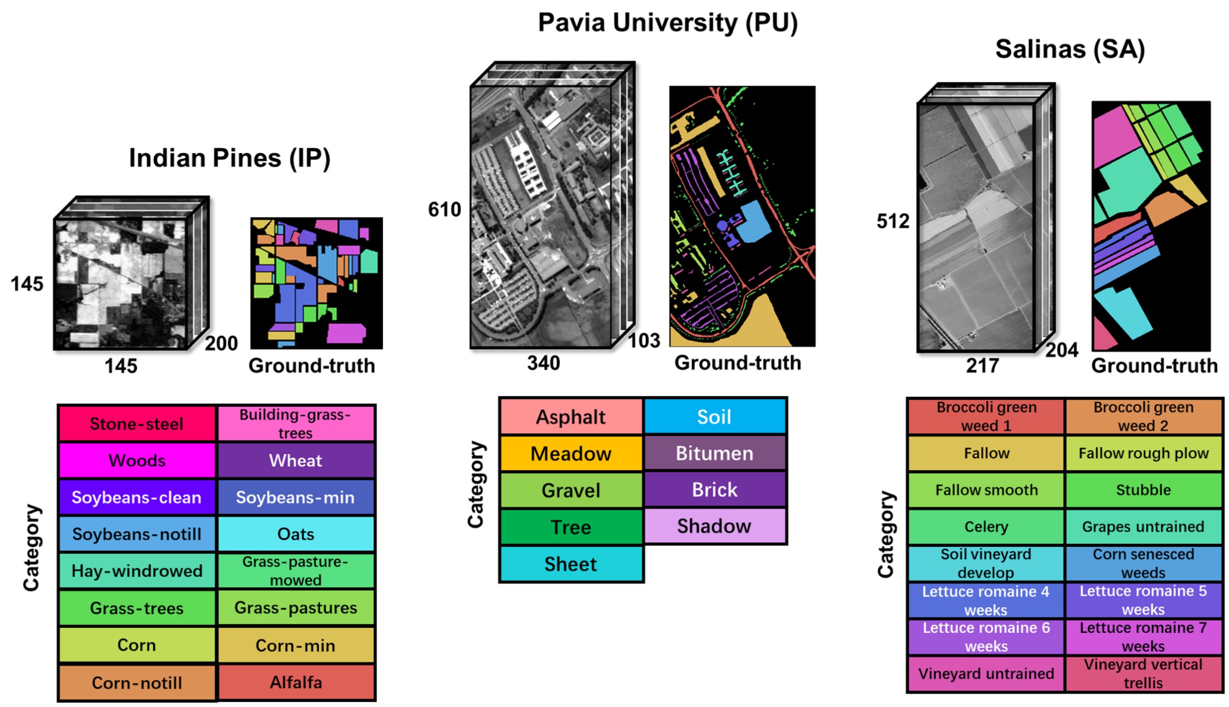


Figure 2.11 Three publicly HSI datasets: IP, PU and SA.

Pavia University (PU) dataset: The PU dataset comprises 115 bands with a spatial resolution of 610 × 340 pixels. In the experiment, 12 channels exhibiting noise were eliminated, leading to the use of 103 bands for the classification task. The dataset comprises nine types of ground cover.

Salinas (SA) dataset: The SA dataset consists of 224 bands with dimensions of 512×217 pixels. Like the IP dataset, 20 water absorption bands are excluded, resulting in 204 remaining bands. The ground truth information for this dataset contains 16 different classes.

2.7.2 Performances by Different Scanning Patterns

We delved into various scanning patterns and investigated their impact on classification results. As depicted in Figure 2.4, we adopted eight distinct scanning patterns, including horizontal, vertical, diagonal, zig-zag, expansion, perimeter, Hilbert, and U-Turn. Each scanning pattern can generate multiple scanning orders, as outlined in Tables 2.1 and 2.2. Initially, we conducted two experiments

Table 2.3 Types of Land-Cover and Pixel Count in Training and Testing Samples across Three Datasets

	Inidan Pines			PU			SA		
Type No.	Name	Training	Testing	Name	Training	Testing	Name	Training	Testing
1	Alfalfa	5	41	Asphalt	663	5967	Broccoli-g-w-1	200	1808
2	Corn-n	143	1285	Meadows	1864	16784	Broccoli-g-w-2	372	3353
3	Corn-m	83	747	Gravel	209	1889	Fallow	197	1778
4	Corn	24	213	Trees	306	2757	Fallow-r-p	139	1254
5	Grass-p	48	435	Metal Sheet	134	1210	Fallow-s	268	2410
6	Grass-t	73	657	Bare Soil	502	4526	Stubble	395	3563
7	Grass-p-m	3	25	Bitumen	133	1197	Celery	357	3221
8	Hay-w	48	430	Bricks	368	3313	Grapes-u	1127	10143
9	Oats	2	18	Shadows	94	852	Soil-v-d	620	5582
10	Soybean-n	97	875				Corn-s-w	327	2950
11	Soybean-m	246	2209				Lettuce-r-4-w	106	961
12	Soybean-c	59	534				Lettuce-r-5-w	192	1734
13	Wheat	20	185				Lettuce-r-6-w	91	824
14	Woods	126	1139				Lettuce-r-7-w	107	963
15	Building-g-t	39	349				Vineyard-u	726	6541
16	Stone-s-t	9	84				Vinyard-v-t	180	1626
Total		1025	9224		3860	38924		5404	48711

for each scanning pattern with RNNs to evaluate the use of single and multiple scanning orders. We must note that the single scanning order follows the illustration in Figure 2.4, and the combination of multiple scanning orders is designed as the summation of their outputs from respective RNNs. The results obtained for the three datasets are illustrated in Figures 2.13.

These two figures show that scanning patterns with non-adjacent pixel connections, such as horizontal and vertical forms, have lower classification accuracy. Conversely, the perimeter, Hilbert, and U-Turn scanning patterns, which are fully spatially continuous, achieve much higher accuracy. Hilbert and U-Turn scanning group, which arrange nearby pixels next to the central pixel in the sequence, typically outperform perimeter scanning.

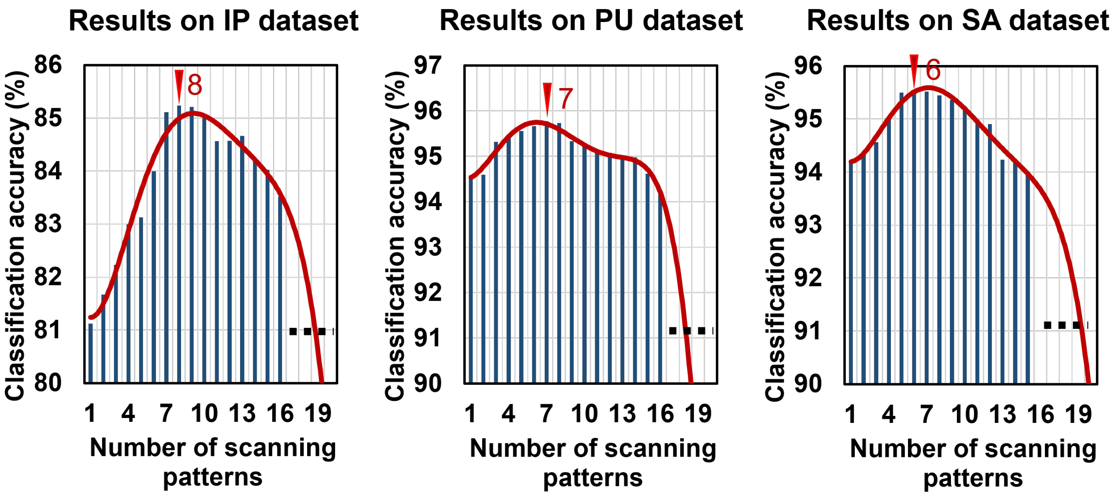


Figure 2.12 The experimental results of checking saturation point when adding multiple scanning orders gradually for model training. The saturation points for three datasets are 8, 7, and 6, respectively.

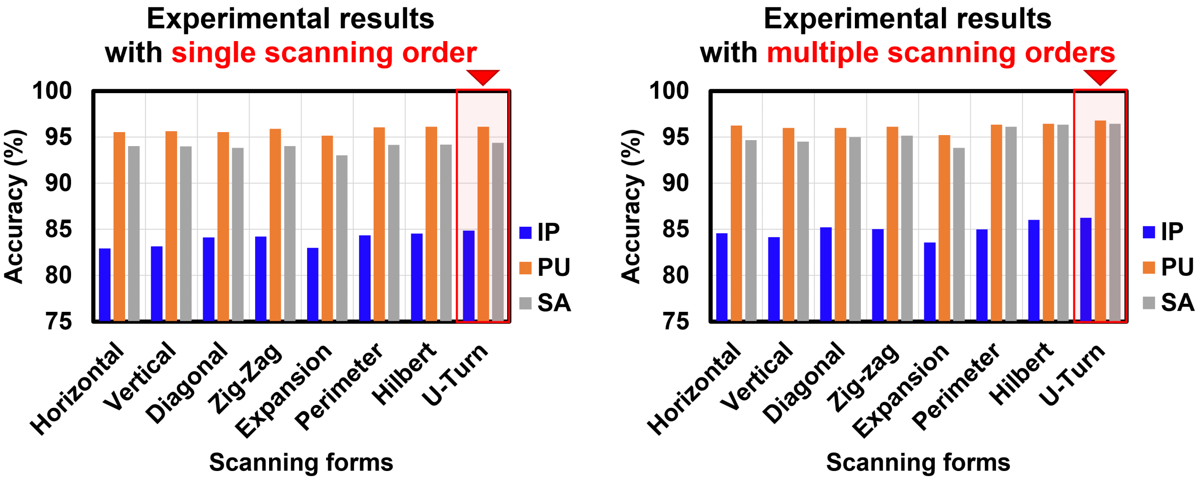


Figure 2.13 Classification outcomes using scanning patterns with single and multiple (eight) scanning orders by RNNs, across three datasets.

The perimeter and expansion scanning patterns are exact opposites, helpful in analyzing how much the first or last step influences the RNN sequence. In all three datasets, the perimeter scanning pattern performs better than the expansion form, suggesting that in RNNs, the sequence's last step has a more significant effect on the final output.

When comparing the diagonal and zig-zag scanning patterns, the zig-zag form yields better results. Figure 2.4 illustrates that the diagonal form has more non-adjacent connections than the zig-zag form. This comparison confirms that an effective scanning pattern should exhibit high spatial continuity.

Notably, the experimental results also verify and support the designated screening procedure and rules proposed before.

2.8 Conclusions with Further Discussion

2.8.1 Conclusions

This chapter elucidates our proposal for the multiscanning strategy. Initially, it discusses the limitations of CNNs, i.e., their inability to effectively handle global features in image encoding. Subsequently, it presents previous works on RNNs used for image scanning, providing a benchmark for the current state of research.

Motivated by our practical observations on implementing multiple scanning orders, we aim to generate diverse pixel-sequences from a single HSI patch comprehensively and adaptively, effectively addressing various spatial dependencies.

Scanning rules are designed to filter suitable scanning patterns that maintain the spatial consistency of image content. From these single scanning patterns, we extend to multiple scanning orders, a process we term our multiscanning strategy.

We conduct preliminary experiments on three datasets to assess our scanning rules and practical classification accuracy with various scanning patterns. The experimental results prove the effectiveness of our rules and screening process. Notably, the U-Turn scanning pattern preserves the spatial consistency of image content exceptionally well, aiding the model in better understanding the image and yielding improved results.

The main contribution of this chapter is summarised as:

1) We propose a multiscanning strategy with RNNs for HSI classification. Meanwhile, we investigate 54 types of scanning patterns, with local scanning region sizes ranging from 5×5 to 15×15, to evaluate classification performance on three HSI datasets: Indian Pines (IP), Pavia University (PU), and Salinas (SA).

2) We explore the characteristics of scanning methods and aim to optimize the number of scanning patterns based on local scanning region size. Specifically, we observe that the horizontal or vertical continuous scanning group, known as the ‘U-Turn’ group, is optimal for preserving spatial geometric properties and achieves better classification results compared to other groups. Furthermore, the ideal number of scanning patterns for the three datasets, IP, PU, and SA, is found to be 8, 7, and 6, respectively. Simultaneously, the optimal local scanning region size for these datasets is identified as 5×5 for IP, and 9×9 for both PU and SA.

3) We confirm the feasibility of employing a multiscanning strategy with RNNs for HSI classification, achieving satisfactory performance and laying the groundwork for future research.

4) We pioneer a new concept for the application of HSI classification.

2.8.2 Further Discussion

With the U-Turn scanning pattern identified as the optimal choice for converting an image into a sequence while maintaining high standards of spatial continuity for model learning, we are inspired to further extend the use of multiple ’U-Turn’ scanning orders as a multiscanning strategy for upcoming research, as outlined in Chapters 3 and 4. Future studies should also consider exploring more innovative and creative designs to combine RNNs with the multiscanning strategy.

Chapter 3

Hyperspectral Image Classification Using Multiscanning-Based RNN with Attention

This chapter further improves our previous research by integrating the multiscanning-based RNNs with attention mechanism as well as RNN-based feature fusion for HSI classification. It starts with a general introduction of RNNs, including the specific variant used in our model; then, it provides a detailed procedure for the designed integrated networks. We perform various experiments to validate the feasibility of our approach, followed by an analysis to support our ideas. The chapter concludes with a summary and a reference to the upcoming Chapter 4.

3.1 Introduction

3.1.1 Remaining Problems in CNN-Based Methods

While methods based on CNNs have demonstrated efficacy in local feature extraction, these techniques have certain limitations. The extraction of local contextual features primarily relies on the small filter kernel size (typically 3×3). This constraint does not adequately capture the spatial dependence between pixels that are not immediately adjacent in the image patch. For instance, during a standard convolution operation, the upper-left and lower-right regions of the image patch do not effectively communicate [62]. The spatial dependency is also essential to interpret the spatial structure information [20, 63, 96]. Therefore, the confined kernel size indeed restricts the receptive field of CNNs. Even though dilated convolutions [52, 53] have been introduced to extend this receptive field during convolutional operations, their efficacy is highly reliant on the dilation rate. It is not particularly suitable for small-sized HSI patches. Furthermore, such an approach tends to disrupt the continuity of adjacent pixels by incorporating nonadjacent ones, thus weakening the consistent spatial distribution necessary for feature extraction. In CNN-based approaches, the spatial dependencies of the HSI patches may also be lost during fully connected or pooling layers [64]. As such, in [63], it was argued that maintaining the spatial dependencies of all pixels is crucial for interpreting spatial structural information.

Moreover, a cropped patch is situated on the edge of a land-cover area. In that case, it often contains a fixed spatial orientation and multiple pixels whose land-cover labels differ from the center pixel. These pixels are referred to as interfering pixels. When such cropped patches are input into CNNs, which employ unidirectional convolution, the interfering pixels can disrupt the extraction of spectral-spatial features. It happens due to the heterogeneous mixture accumulating through the convolution process [65]. Consequently, the classification results may be unsatisfactory, often leading to unclear boundaries. The unidirectional nature of CNN's convolution operation contributes to this problem, as it cannot account for and handle multiple spatial orientations in the data.

Therefore, developing a solution that can manage global dependencies while maintaining spatial dependencies is essential and is inherently capable of multi-directional generation. This approach would optimally capture HSI data's diverse and complex dependencies.

3.1.2 Limitations in Previous RNN-Based Methods

Many previous studies [6, 21, 25, 62, 84] have proposed the use of RNNs as an alternative to CNNs. This shift is motivated by the unique strengths of RNNs. Unlike CNNs, which are primarily adept at capturing local spatial dependencies, RNNs excel at learning from data sequences of arbitrary length. They can effectively learn the long-term spatial dependencies present in an image. It enables them to capture the global context of an image, leading to potentially better results. By replacing CNNs with RNNs for HSI classification, researchers aim to harness these advantages and push the boundaries of what is possible in this field.

The RNNs mentioned previously do consider spectral-spatial features. However, they tend to overlook explicitly modeling dependencies in different spatial directions. As land-cover distributions change, these methods often misclassify pixels in heterogeneous regions, particularly at the cross-class region edges. Here, the relative pixel positions between two land-cover classes can shift orientation – from left to right, up to down, and so on. As discussed in Section 1.3, these models face several limitations, including 1) Single Sequence Dependence, 2) Limited Spatial Context, 3) Low Generalization Capabilities, 4) Temporal Limitations, and 5) Over-reliance on Spectral Features.

Given the limitations of previous work that only considers unidirectional dependency relationships, this study introduces a novel multiscanning strategy with RNNs to account for multi-directional spatial dependencies naturally. This new approach is designed to enhance the understanding and utilization of these relationships in HSI classification tasks.

3.1.3 Summarization

In this study, we propose a multiscanning strategy that leverages the unique sequential nature of HSI pixels and considers spatial dependencies at the pixel level. This strategy transforms a local HSI patch into multiple contiguous sequences based on different scanning directions. These scanning directions provide complementary perspectives on a single local spatial patch. We further devise a direction-features-based scheme that synergistically integrates these complementary scans using their correlation weights through an LSTM. The proposed network seamlessly facilitates the end-to-end HSI classification task in four stages. Compared to previous RNN-based methods, our approach offers three key advantages: 1) It effectively integrates spatial and spectral information; 2) It accommodates correlative dependencies through multiscanning approaches; 3) It fully addresses diverse spatial dependencies via the multiscanning strategy.

3.2 Spectral-Spatial Structure of HSI for RNN

The data is often divided into spatial and spectral domains in typical HSI experiments. It necessitates time-consuming and practical pre-processing skills with existing methodologies. We aim to overcome this by proposing the integration of these two domains via the use of RNNs and a multiscanning strategy, as sketched in Figure 3.1.

Consider an HSI image sized $\mathcal{H} \times \mathcal{W} \times \mathcal{C}$. Initially, we extract a local patch $\mathbf{X}^{(i,j)} \in \mathbb{R}^{p \times p \times C}$ with dimensions p, forming a 3D data cube. This patch is subsequently converted into several sequential data (pixel-sequences) $\mathbf{S}_m^{(i,j)} \in \mathbb{R}^{p^2 \times C}$ through the application of a multiscanning strategy. Here, m denotes the m-th scanning order within the strategy, with $m = 1, 2, 3, \ldots, \mathcal{M}$, where $\mathcal{M}$ is the total number of scanning orders.

In a pixel-sequence, each element is a feature vector containing the spectral information of a single pixel derived from the original data. Therefore, each element becomes a step in the RNN process, with the total number of steps equating to the number of pixels $\mathcal{N} = p \times p$ in the patch.

This conversion of the spectral-spatial structure into a sequential format allows the incorporation of neighboring relationships, thereby simultaneously integrating spectral and spatial information while effectively considering the relationships between neighboring local pixels. As a result, the proposed spectral-spatial structure of the HSI, both conceptually and structurally, is

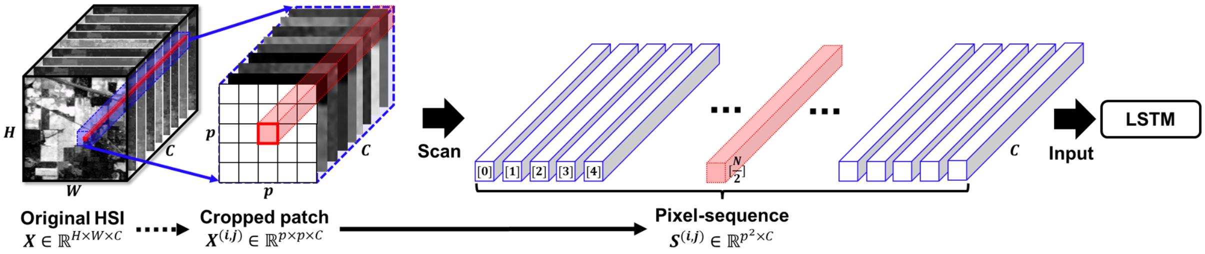

H
W
C
p
p
C
Scan
[0]
[1]
[2]
[3]
[4]
$[\frac{N}{2}]$
C
Input
LSTM
Original HSI
$X \in \mathbb{R}^{H\times W\times C}$
Cropped patch
$X^{(i,j)} \in \mathbb{R}^{p\times p\times C}$
Pixel-sequence
$S^{(i,j)} \in \mathbb{R}^{p^2\times C}$

Figure 3.1 The illustration of spectral-spatial structure of HSI patch for RNN.

well-adapted to serve as an input for the RNN model.

3.3 RNN for Multiscanning Pixel-Sequences

The RNN model excels at learning patterns within sequential data. With each step in the sequence, the activation of the previous step influences the current activation. Consequently, the output of the final step carries a 'memory' of previous steps, which impacts the network's ultimate decision [44].

When incorporating the RNN into our multiscanning strategy, for each scanned pixel-sequence $\mathbf{S}_m^{(i,j)}$, the s-th order pixel's hidden state $\mathbf{H}_m^{(i,j)}[s]$ is updated by referencing the previous $(s-1)$-th hidden state $\mathbf{H}_m^{(i,j)}[s-1]$ and the current s-th input $\mathbf{S}_m^{(i,j)}[s]$. This update is achieved via the following equation:

$$\mathbf{H}_m^{(i,j)}[s] = \begin{cases} 0, & \text{if } s = 0 \\ \mathcal{F}\left(\mathbf{H}_m^{(i,j)}[s-1], \mathbf{S}_m^{(i,j)}[s]\right), & \text{otherwise.} \end{cases} \tag{3.1}$$

In this case, $0 \le s \le \mathcal{N} - 1$ and $\mathcal{N} = p \times p$ where p represents the patch size. ϕ is a nonlinear function, such as a logistic sigmoid or hyperbolic tangent function. The recurrent hidden state's update rule from Equation (3.1) is executed as follows:

$$\mathbf{H}_m^{(i,j)}[s] = \phi\left(\mathbf{W}_{sh}^m \odot \mathbf{S}_m^{(i,j)}[s] + \mathbf{W}_{hh}^m \odot \mathbf{H}_m^{(i,j)}[s-1] + \mathbf{b}_h^m\right), \tag{3.2}$$

where $\mathbf{W}_{sh}^{m}$ denotes the input-to-hidden transformation matrix, and $\mathbf{W}_{hh}^{m}$ is the hidden-to-hidden transformation matrix that connects the hidden layer of two adjacent steps. $\mathbf{b}_{h}^{m}$ denotes the bias, all of which are associated with the m-th scanning order of the pixel-sequence $\mathbf{S}_{m}^{(i,j)}$.

The output $\mathbf{y}_{m}^{(i,j)}$, derived from the last hidden state $\mathbf{H}_{m}^{(i,j)}[N-1]$ for the m-th scanning order, can be computed as follows:

$$\mathbf{y}_{m}^{(i,j)} = \mathbf{Y}_{m}^{(i,j)}[N-1] = \phi\Big(\mathbf{W}_{hy}^{m} \odot \mathbf{H}_{m}^{(i,j)}[N-1] + \mathbf{b}_{y}^{m}\Big), \tag{3.3}$$

where $\mathbf{W}_{hy}^{m}$ is the hidden-to-output weight matrix and $\mathbf{b}_{y}^{m}$ is the output bias.

Consequently, in the multiscanning strategy, several scanned HSI pixel-sequences, $\mathbf{S}_{m}^{(i,j)} \in \mathbb{R}^{p^2 \times C}, m = 1, 2, ..., \mathcal{M}$, serve as separate inputs for the RNN model. The last output from each scanned pixel-sequence, denoted as $\mathbf{y}_{m}^{(i,j)} \in \mathbb{R}^{1 \times C}$, will be used as output representations for each scanning order $\mathcal{M}$. We denote this feature set as $\mathbf{Y}^{(i,j)}$. Hence, we can express it as follows:

$$\mathbf{Y}^{(i,j)} = \Big\{\mathbf{y}_{m}^{(i,j)} \Big| m = 1, 2, 3, ..., \mathcal{M}\Big\}. \tag{3.4}$$

Thus, $\mathbf{Y}^{(i,j)}$ comprises the entire enhanced feature set for learning the input patch $\mathbf{X}^{(i,j)}$.

In the experimental stages, we will examine the classification performance of the standard RNN and its variants, the Gated Recurrent Unit (GRU) [12] and LSTM [31].

3.4 Scanning Order-Based Attention

As previously mentioned, each scanning order in the multiscanning strategy offers a complementary view of an image patch. Hence, a suitable merging technique is required to leverage this supplementary information effectively. Figure 3.2 illustrates potential combination methods. We will explore and experiment with these different methods in later sections.

Typically, the simplest way to combine the features from each scan is by summing or averaging them. However, this implies that each scan contributes equally to the classification task,

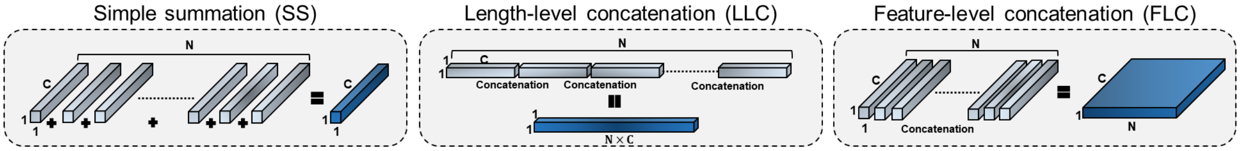


Figure 3.2 The diagram illustrates the possible combination manners of features in the RNN. The grey cube represents the output feature from RNN for each scanning order, while the blue cube represents the combined feature.

regardless of differing spatial dependencies. The truth is that the multiscanning strategy caters to different spatial dependencies within the HSI patch; spatial dependencies vary with different scanning orders. Consequently, specific scanning directions should be assigned lesser weight, while key scanning directions should carry more weight.

Motivated by [66], we propose introducing the attention mechanism, which intuitively constructs relationships between these scanning orders. Hence, it aims to combine all the output features from the RNN in a weighted fashion. Specifically, the weighted feature can be represented as follows:

$$\mathbf{Y}_c^{(i,j)} = Concat\left[\mathbf{y}_1^{(i,j)}, \mathbf{y}_2^{(i,j)}, \mathbf{y}_3^{(i,j)}, \mathbf{y}_4^{(i,j)}, \mathbf{y}_5^{(i,j)}, \mathbf{y}_6^{(i,j)}, \ldots, \mathbf{y}_{\mathcal{M}}^{(i,j)}\right]^{\top}. \tag{3.5}$$

$$\mathbf{E}^{(i,j)} = ReLU\left(\mathbf{Y}_c^{(i,j)} \odot \mathbf{W}_{ye} + \mathbf{b}_{ye}\right), \tag{3.6}$$

where $\mathbf{Y}^{(i,j)} \in \mathbb{R}^{\mathcal{M}\times d}$ and Equation (3.6) represents a single linear projection layer. This layer reorders the output of each RNN in its current vector space. Afterward, it applies the $ReLU$ activation function to transform and obtain $\mathbf{E}^{(i,j)}$ as a new feature representation of $\mathbf{Y}_c^{(i,j)}$. Here, $\mathbf{W}_{ye}$ is the transformation matrix and $\mathbf{b}_{ye}$ is the bias.

The attention weight $\mathbf{w}^{(i,j)}$ is produced through the $Softmax$ layer, formulated as follows:

$$\mathbf{A}^{(i,j)} = \left(\mathbf{E}^{(i,j)} \odot \mathbf{W}_{ea} + \mathbf{b}_{ea}\right), \tag{3.7}$$

$$\mathbf{w}^{(i,j)} = Softmax\left(\mathbf{A}^{(i,j)}\right) \tag{3.8}$$

where $\mathbf{w}^{(i,j)} \in \mathbb{R}^{\mathcal{M}\times 1}$. $\mathbf{W}_{ea}$ represents the transformation matrices, while b_{ea} is the bias term. The $Softmax$ function converts the non-normalized output into a probability distribution, constraining the output values to fall within the $(0,1)$ range.

Once we have determined the updated attention weights $\mathbf{w}^{(i,j)}$, we use them to broadcast the feature representation $\mathbf{Y}_c^{(i,j)}$ at each element (scanning order) m, as follows:

$$\tilde{\mathbf{Y}}_c^{(i,j)}[m] = \tilde{\mathbf{y}}_m^{(i,j)} = \mathbf{w}^{(i,j)}[m] \times \mathbf{Y}_c^{(i,j)}[m]. \tag{3.9}$$

These weights can be effortlessly integrated into the entire network, with their optimal values being automatically learned from the data. Consequently, the output features $\tilde{\mathbf{Y}}_c^{(i,j)} \in \mathbb{R}^{\mathcal{M}\times d}$ obtained through the scanning order-based attention module can be expressed as follows:

$$\tilde{\mathbf{Y}}_c^{(i,j)} = Concat\left[\tilde{\mathbf{y}}_1^{(i,j)}, \tilde{\mathbf{y}}_2^{(i,j)}, \tilde{\mathbf{y}}_3^{(i,j)}, \tilde{\mathbf{y}}_4^{(i,j)}, \tilde{\mathbf{y}}_5^{(i,j)}, \tilde{\mathbf{y}}_6^{(i,j)}, \ldots, \tilde{\mathbf{y}}_{\mathcal{M}}^{(i,j)}\right]^{\top}. \tag{3.10}$$

Our model can achieve a more coherent interpretation by assigning weights to different scanning orders, highlighting the significance of specific scanning orders while deeming others less critical. Moreover, our model can effectively capture the spectral-spatial features of HSI patches, establish correlations among neighboring pixels, and prioritize suitable transformations of HSI patches. These enhancements contribute towards improving the accuracy of our training model.

3.5 Integrated Network

This section introduces the combined network structures comprising two schemes, as depicted in Figures 3.3 and 3.4. The first scheme treats each directional sequence as an individual input to a single LSTM unit. In the second scheme, Bidirectional LSTM (Bi-LSTM) is deployed to process four forward and backward pixel-sequences pairs. Subsequently, we concatenate the learned features and scanning order-based attentions to take advantage of their complementary properties for a single HSI patch.

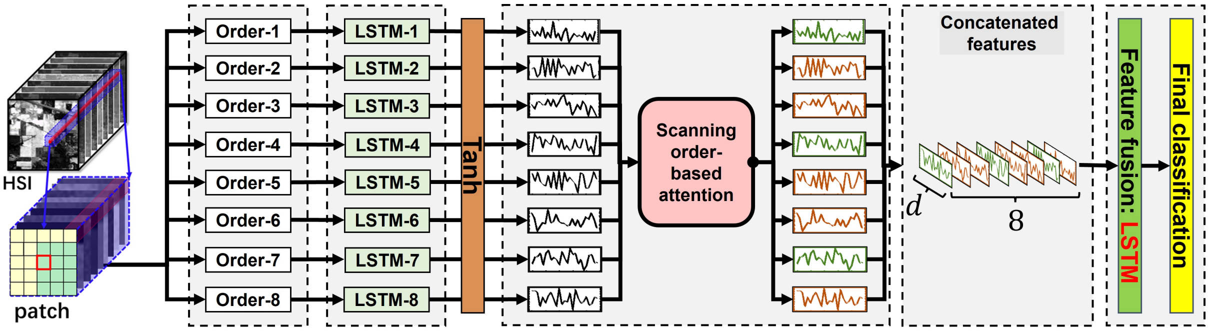


Figure 3.3 Scheme-1 depicts a multiscanning-based model with a separate procedure for each direction. This approach involves processing the HSI patch from multiple scanning directions independently. Each scanning direction follows its distinct pathway, enabling the extraction of directional-specific features. These features are then combined or fused to represent the input comprehensively.

3.5.1 Scheme-1: Separating Multiscanning Feature Fusion by RNN

In this scheme, the multiscanning directional sequences are individually input into a single LSTM unit, and the resulting output representations are concatenated to capture their complementarity, as depicted in Figure 3.3.

The procedure converts a cropped HSI patch into multiple pixel-sequences via a multiscanning strategy. Each directional pixel-sequence is then independently introduced into an LSTM unit. The corresponding feature representation $\mathbf{y}_m^{(i,j)}$ for each scanned pixel-sequence encapsulates the spatial dependencies amongst all pixels within the patch at scanning order m. Subsequently, the learned features from all pixel-sequences are integrated with their corresponding learnable fusion weights through feature-level concatenation, as follows:

$$\tilde{\mathbf{Y}}_c^{(i,j)} = Concat\left[\tilde{\mathbf{y}}_1^{(i,j)}, \tilde{\mathbf{y}}_2^{(i,j)}, \tilde{\mathbf{y}}_3^{(i,j)}, \ldots, \tilde{\mathbf{y}}_m^{(i,j)}\right], m = 1, 2, 3, \ldots, \mathcal{M} \tag{3.11}$$

where $\tilde{\mathbf{Y}}_c^{(i,j)}$ is a new sequential data.

Finally, the concatenated feature $\tilde{\mathbf{Y}}_c^{(i,j)}$ is input into the subsequent LSTM to capture the interdependence and complementary nature of these scanning orders, following the procedure described below: $\tilde{\mathbf{y}}_1^{(i,j)} \rightarrow \tilde{\mathbf{y}}_2^{(i,j)} \rightarrow \tilde{\mathbf{y}}_3^{(i,j)} \rightarrow \tilde{\mathbf{y}}_4^{(i,j)} \rightarrow \tilde{\mathbf{y}}_5^{(i,j)} \rightarrow \tilde{\mathbf{y}}_6^{(i,j)} \rightarrow \ldots \rightarrow \tilde{\mathbf{y}}_{\mathcal{M}}^{(i,j)}$, therefore referring to the Equations (3.1) and (3.2):

$$\mathbf{H}_c^{(i,j)}[m] = \phi\Big(\mathbf{W}_{yc}^m \odot \tilde{\mathbf{Y}}_c^{(i,j)}[m] + \mathbf{W}_{hc}^m \odot \mathbf{H}_c^{(i,j)}[m-1] + \mathbf{b}_c^m\Big). \tag{3.12}$$

In this context, the hidden state $\mathbf{H}_c^{(i,j)}[m]$ represents the state of the LSTM at the m-th scanning order, where the input to this state is $\tilde{\mathbf{Y}}_c^{(i,j)}[m]$. Specifically, we set the initial input as $\tilde{\mathbf{Y}}_c^{(i,j)}[1] = \tilde{\mathbf{y}}_1^{(i,j)}$. The hidden state is computed using the input-to-hidden transformation matrix $\mathbf{W}_{yc}^m$, the hidden-to-hidden transformation matrix $\mathbf{W}_{hc}^m$, and the bias term $\mathbf{b}_c^m$.

The classification procedure employs the output representation derived from the last scanning order, symbolized as $\mathbf{H}_c^{(i,j)}[M]$. The formulation is calculated as follows:

$$\mathbf{y}_c^{(i,j)} = \tilde{\mathbf{Y}}_c^{(i,j)}[M] = \phi\Big(\mathbf{W}_{cc}^m \odot \mathbf{H}_c^{(i,j)}[M] + \mathbf{b}_{cc}^m\Big), \tag{3.13}$$

Here, $\mathbf{W}_{cc}^m$ represents the weight matrices, and $\mathbf{b}_{cc}^m$ denotes the biases. The output undergoes an activation function ϕ to further process the results.

As a result, for the input HSI patch $\mathbf{X}^{(i,j)}$, we obtain a final feature representation $\mathbf{y}_c^{(i,j)}$ that incorporates the multi-directional features to gain a comprehensive understanding of the input. This approach aims to enhance the feature pool and improve the generative ability when dealing with unseen test data.

3.5.2 Scheme-2: Pairing Multiscanning Feature Fusion by RNN

As mentioned earlier, each scanning pattern can generate four pairs of forward and backward scanning orders within the multiscanning strategy. We intuitively feed these pairs into a Bi-RNN to leverage the benefits of a wider receptive field and enhance the model's efficiency and effectiveness. This approach allows the model to capture information from both past and future contexts. The Scheme-2 model, illustrated in Figure 3.4, demonstrates the incorporation of the Bi-RNN into our framework.

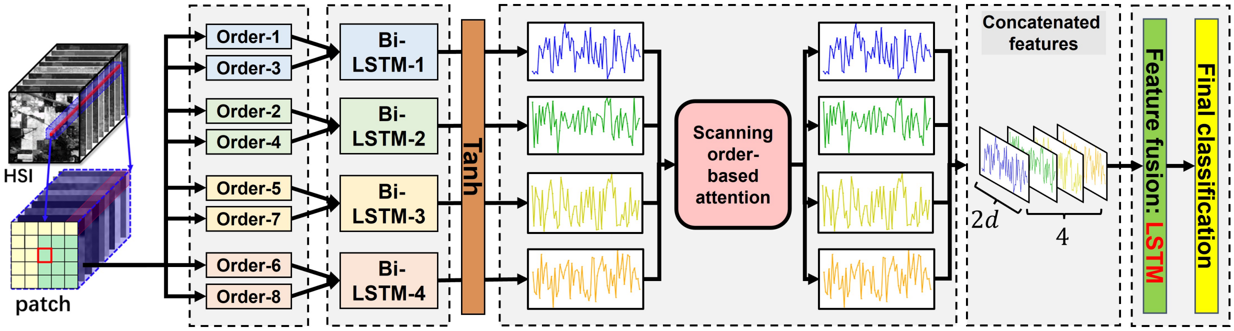


Figure 3.4 Scheme-2 showcases a framework incorporating a Bi-LSTM network for pairs of scanning directions. In this approach, the input is processed by the Bi-LSTM network in both forward and backward directions, allowing the model to capture contextual information from different perspectives. The model can effectively capture dependencies between adjacent directions by considering pairs of scanning directions and enhance the overall feature representation. The output features from the Bi-LSTM network are combined or fused to represent the input comprehensively.

RNNs have demonstrated impressive performance in analyzing sequential data. Unlike standard RNNs, Bidirectional RNNs (Bi-RNNs) incorporate two hidden layers to simultaneously process data in both forward and backward directions, generating a combined output; this allows the model to consider information from both past and future states, resulting in a wider receptive field. The calculations involved in a Bi-RNN are as follows:

$$\overrightarrow{\mathbf{H}_m^{(i,j)}[s]} = \phi\left(\overrightarrow{\mathbf{W}_{sh}^m} \odot \mathbf{S}_m^{(i,j)}[s] + \overrightarrow{\mathbf{W}_{hh}^m} \odot \mathbf{H}_m^{(i,j)}[s-1] + \overrightarrow{\mathbf{b}_h^m}\right), \tag{3.14}$$

$$\overleftarrow{\mathbf{H}_m^{(i,j)}[s]} = \phi\left(\overleftarrow{\mathbf{W}_{sh}^m} \odot \mathbf{S}_m^{(i,j)}[s] + \overleftarrow{\mathbf{W}_{hh}^m} \odot \mathbf{H}_m^{(i,j)}[s+1] + \overleftarrow{\mathbf{b}_h^m}\right), \tag{3.15}$$

Here, $\overrightarrow{\mathbf{H}_m^{(i,j)}[s]}$ and $\overleftarrow{\mathbf{H}_m^{(i,j)}[s]}$ represent the forward and backward outputs at the s-th order pixel in the pixel-sequence. The weight coefficient matrices for the forward and backward directions are denoted as $\overrightarrow{\mathbf{W}_{sh}^m}$, $\overleftarrow{\mathbf{W}_{sh}^m}$, $\overrightarrow{\mathbf{W}_{hh}^m}$, and $\overleftarrow{\mathbf{W}_{hh}^m}$. The biases for the forward and backward directions are represented by $\overrightarrow{\mathbf{b}_h^m}$ and $\overleftarrow{\mathbf{b}_h^m}$, respectively.

As an example, $\overrightarrow{\mathbf{H}_m^{(i,j)}[s]}$ can be considered as the output from $\mathbf{H}_1^{(i,j)}[s]$, while $\overleftarrow{\mathbf{H}_m^{(i,j)}[s]}$ can be regarded as the output from $\mathbf{H}_3^{(i,j)}[s]$. In the subsequent step, we concatenate the outputs from the

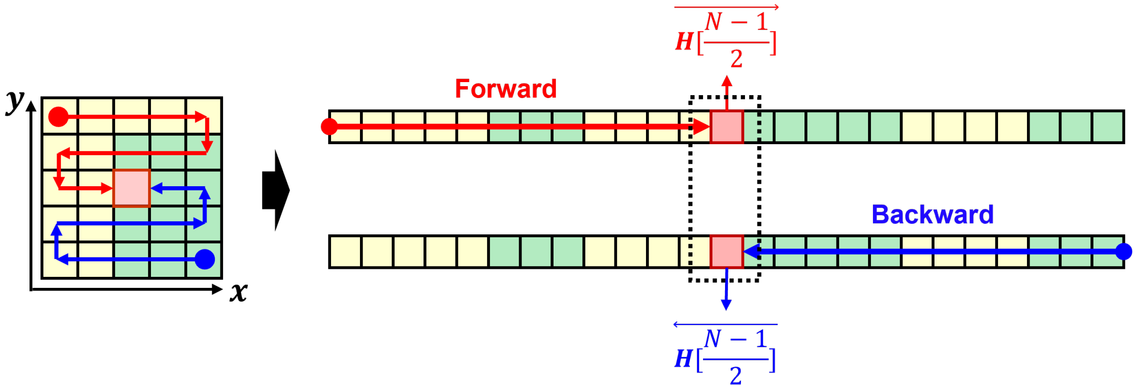

y
x
Forward
Backward
$\overrightarrow{H}[\frac{N-1}{2}]$
$\overleftarrow{H}[\frac{N-1}{2}]$

Figure 3.5 The concept of employing a Bi-RNN with two inverse sequences. In this setup, the central step, denoted as $\frac{N-1}{2}$, uniformly learns features from its adjacent neighbors within the same steps. Two complementary outputs from central step, $\overrightarrow{\mathbf{H}\left[\frac{N-1}{2}\right]}$ and $\overleftarrow{\mathbf{H}\left[\frac{N-1}{2}\right]}$, are concatenated for subsequent process.

Bi-RNN of these two inverse sequences using the following operation:

$$\mathbf{H}_m^{(i,j)}[s] = Concat\left[\overrightarrow{\mathbf{H}_m^{(i,j)}[s]}, \overleftarrow{\mathbf{H}_m^{(i,j)}[s]}\right]. \tag{3.16}$$

The output $\mathbf{y}_{m_2}^{(i,j)}$ from Bi-RNN of two inverse sequences is obtained at $\frac{N-1}{2}$-th step, as illustrated in Figure 3.5:

$$\mathbf{H}_{m_2}^{(i,j)}\left[\frac{N-1}{2}\right] = Concat\left[\overrightarrow{\mathbf{H}_m^{(i,j)}\left[\frac{N-1}{2}\right]}, \overleftarrow{\mathbf{H}_m^{(i,j)}\left[\frac{N-1}{2}\right]}\right]. \tag{3.17}$$

$$\mathbf{y}_{m_2}^{(i,j)} = \phi\left(\mathbf{W}_{hy}^{m_2} \odot \mathbf{H}_{m_2}^{(i,j)}\left[\frac{N-1}{2}\right] + \mathbf{b}_y^{m_2}\right), \tag{3.18}$$

where $\mathbf{W}_{hy}^{m_2}$ represents the hidden-to-output transformation matrix and $\mathbf{b}_y^{m_2}$ is the added bias. In this place, $m_2 = 1, 2, ..., \frac{M}{2}$.

In the experiments, multiple pairs of pixel-sequences are inputted into the Bi-RNN. The resulting learned feature representations, denoted as $\mathbf{y}_1^{(i,j)}, \mathbf{y}_2^{(i,j)}, ..., \mathbf{y}_{\frac{M}{2}}^{(i,j)}$, for each pair are passed through the hyperbolic tangent (tanh) activation function. Subsequently, these features are combined with their respective learnable fusion weights $w_1^{(i,j)}, w_2^{(i,j)}, ..., w_{\frac{M}{2}}^{(i,j)}$ using Equations (3.6) and (3.7), as follows:

$$\tilde{\mathbf{y}}_{m_2}^{(i,j)} = w_{m_2}^{(i,j)} \times \mathbf{y}_{m_2}^{(i,j)}. \tag{3.19}$$

$$\tilde{\mathbf{Y}}^{(i,j)} = \left\{ \tilde{\mathbf{y}}_{m_2}^{(i,j)} \middle| m_2 = 1, 2, 3, ..., \frac{\mathcal{M}}{2} \right\} \tag{3.20}$$

$$\tilde{\mathbf{Y}}_c^{(i,j)} = Concat \left[\tilde{\mathbf{y}}_1^{(i,j)}, \tilde{\mathbf{y}}_2^{(i,j)}, ..., \tilde{\mathbf{y}}_{m_2}^{(i,j)} \right] \tag{3.21}$$

Additionally, each $\tilde{\mathbf{y}}_m^{(i,j)}{}_2$ is activated using the Rectified Linear Unit (ReLU) function, which is effective in creating sparse representation features. Sparse representations are advantageous and effective in representation learning compared to dense representations [90].

Finally, the concatenated feature $\tilde{\mathbf{Y}}_c^{(i,j)}$ is input into the subsequent LSTM to capture the complementarity and correlation between these scanning orders in the following sequence: $\tilde{\mathbf{y}}_1^{(i,j)} \rightarrow \tilde{\mathbf{y}}_2^{(i,j)} \rightarrow ... \rightarrow \tilde{\mathbf{y}}_{\frac{\mathcal{M}}{2}}^{(i,j)}$.

3.6 Experiments and Analysis

3.6.1 Experiment Design

The experiments are conducted on three datasets (IP, PU, and SA), as shown in Figure 2.11 before. The labeled samples from the HSI datasets are randomly divided into training and testing sets to evaluate the proposed and compared methods' performance and feasibility. In this process, we designate 10% of the pixels as training samples, while the remaining 90% are allocated for testing. The numbers of samples selected for training and testing are summarized in Table 2.3 before.

To compare the performance of our proposed method, we include several well-known methods as baselines. These methods include spectral-based approaches such as Spectral CNN [33] and Deep RNN [48], spatial-based method FCN [40], and spectral-spatial-based methods such as 3D CNN [43], M3D-DCNN [29], and SSAN [65]. The parameter settings for these compared methods are obtained from their respective references.

During the training phase, we employ a batch size of 100 and utilize the Adam optimization algorithm with an initial learning rate of 0.001. The models are trained for 200 epochs. To enhance the training process, we incorporate a dropout layer with a probability of 0.5 and a batch normalization layer before the fully connected layer. All LSTM layers are set to 1, and each LSTM layer consists of 64 hidden units. For the Bi-LSTM, we use two layers, one for the forward direction and another for the backward direction, with 64 hidden units in each layer.

All experiments use the PyTorch 1.1 platform and the GeForce RTX 1080 GPU, providing computational acceleration for efficient training and evaluation.

The classification performance from all methods is evaluated by overall accuracy (OA), average accuracy (AA), Kappa coefficient ($Kappa$), and accuracy for each class.

3.6.2 Experiment Results

IP Dataset

Figure 3.6 presents the classification maps achieved by different methods on the IP dataset. The corresponding quantitative results for each method are in Table 3.1.

Figure 3.6 and Table 3.1 demonstrate that the spectral-based methods, spectral CNN and Deep RNN, which only utilize a single pixel's spectral data without its neighboring pixels, lead to poor classification results. These misclassifications manifest as noise-like and scattered. In an HSI, spatially close pixels likely belong to the same class. As indicated by the marked flat regions in Figure 3.6, pixels in these areas should be assigned the same label. However, due to spectral signal similarities among different materials, and variations within the same material, misclassifications occur unpredictably and widely. In Table 3.1, spectral-based methods perform unsatisfactorily for classes such as '4', '9', and '15', mainly due to unbalanced training samples and complex spectral signals.

Despite FCN incorporating spatial information during training, it uses a dimension reduction

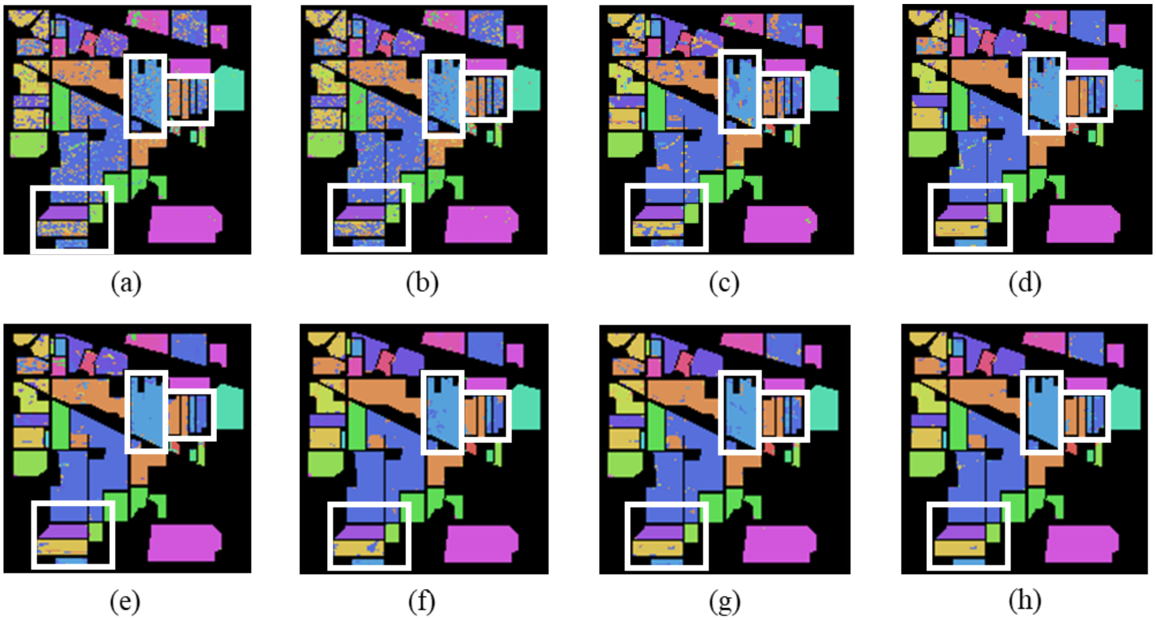


Figure 3.6 Various method-based classification maps for the IP image: (a) Spectral CNN. (b) Deep RNN. (c) FCN. (d) 3D CNN. (e) M3D-DCNN. (f) SSAN. (g) First Proposed Scheme (Ours). (h) Second Proposed Scheme (Ours).

pre-process that disrupts the correlation between spectral bands. Figure 3.6 (c) shows fewer scattered misclassifications, but we can still observe misclassifications resembling objects, indicating FCN's attention to spatial pixel correlations. A closer look at the white boxes reveals that these object-like errors are mostly found at the edges of class regions.

Combining spectral and spatial information leads to insightful results, making 3D CNN an appropriate choice. Better outcomes are achieved by 3D CNN, M3D-DCNN, and SSAN due to their consideration of both types of information. Specifically, Figure 3.6 (d), (e), and (f) display their results, which are generally satisfactory with smoother classifications in more significant regions. However, small-sized HSI patches may contain mixed variations, potentially skewing 3D convolution results. It can lead to unclear boundaries and misclassifications between neighboring classes. SSAN mitigates this issue by applying spatial attention, reducing the influence of interfering pixels, and focusing more on similar class pixels within an HSI patch.

Table 3.1 Quantitative performance metrics, including OA, AA, and Kappa, along with the accuracies for each class, were evaluated for various classification methods for the IP dataset. Best results are highlighted.

Classes	Methods							
	spectral CNN	Deep-RNN	FCN	3D CNN	M3D-DCNN	SSAN	Scheme-1(ours)	Scheme-2(ours)
1	21.78	76.71	40.82	80.05	63.33	**88.68**	86.66	87.27
2	62.77	74.90	75.31	82.57	88.75	93.34	93.58	**95.68**
3	63.31	61.44	57.88	72.19	81.16	90.75	92.17	**94.75**
4	54.94	56.30	61.29	87.23	84.09	84.37	90.33	**91.77**
5	36.80	92.99	91.62	93.72	93.28	97.21	96.99	**98.58**
6	86.33	94.72	95.38	97.81	97.79	**98.67**	97.36	98.55
7	80.02	92.68	60.66	**95.48**	81.19	86.20	88.91	91.51
8	93.05	97.95	95.16	98.41	98.41	**100**	**100**	**100**
9	0	60.05	56.02	60.88	86.77	78.89	94.16	**100**
10	52.71	70.13	67.68	75.67	90.30	**95.33**	93.02	95.30
11	70.35	75.06	75.33	83.42	91.64	93.85	94.11	**97.05**
12	57.79	76.89	73.79	80.57	82.07	95.52	95.31	**97.38**
13	88.52	98.81	98.88	99.66	**100**	**100**	**100**	**100**
14	84.36	94.37	93.32	96.49	97.66	97.40	96.18	**98.91**
15	43.69	71.89	68.37	75.23	68.25	**82.29**	79.58	80.33
16	91.44	92.23	88.59	97.88	93.88	**99.03**	98.37	98.34
OA	69.80	79.94	79.12	85.49	90.22	**95.57**	95.13	95.38
AA	61.74	80.45	75.01	86.08	88.41	94.26	93.55	**95.45**
Kappa	0.64	0.74	0.75	0.85	0.88	**0.95**	0.93	**0.95**

Our proposed schemes learn correlations pixel by pixel, unlike 3D CNN, which collectively convolves them. Scheme-2 outperforms Scheme-1, achieving results comparable to SSAN. This improvement is due to local spatial sequences that account for pixel relationships in nearby areas, often overlooked in CNN-based methods. Close examination of Figure 3.6 (g) and (h) reveals that our models produce more precise boundaries and more accurate classifications, especially in areas highlighted by the white boxes.

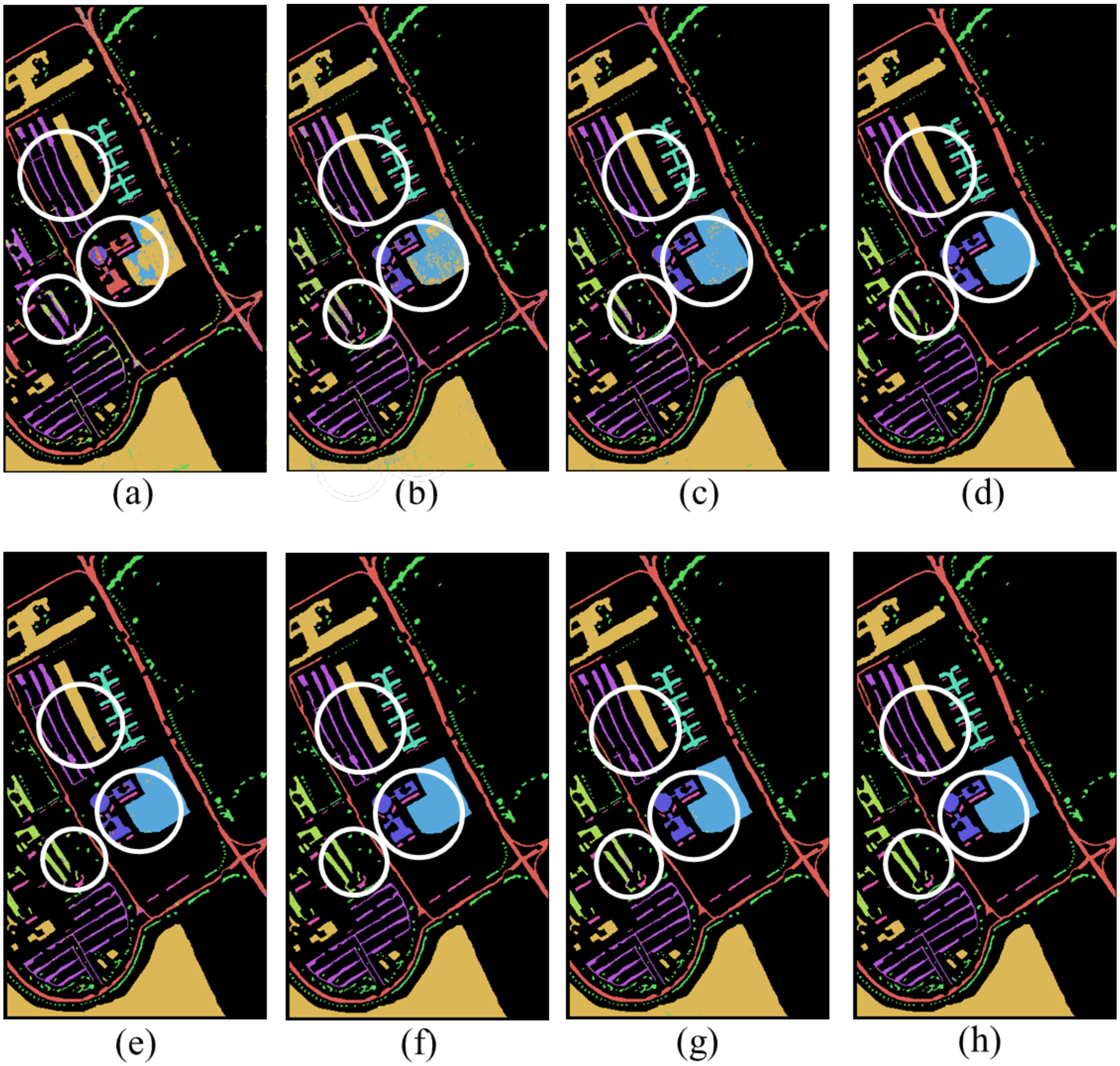


Figure 3.7 Classification maps for the PU image utilizing different methods: (a) Spectral CNN. (b) Deep RNN. (c) FCN. (d) 3D CNN. (e) M3D-DCNN. (f) SSAN. (g) First Proposed Scheme (Ours). (h) Second Proposed Scheme (Ours).

PU Dataset

Figure 3.7 displays the classification maps achieved by all methods on the PU dataset. Table 3.2 shows corresponding quantitative results for each method.

Table 3.2 confirms that methods incorporating local spatial features of an HSI patch yield better

Table 3.2 Quantitative performance metrics, including OA, AA, and Kappa, along with the accuracies for each class, were evaluated for various classification methods for the PU dataset. Best results are highlighted.

Class	Methods							
	spectral CNN	Deep-RNN	FCN	3D-CNN	M3D-DCNN	SSAN	Scheme-1(ours)	Scheme-2(ours)
1	92.60	92.23	96.64	96.05	98.61	98.65	99.19	**99.88**
2	92.91	95.91	96.24	97.99	98.10	**98.83**	97.29	97.97
3	76.01	70.89	92.68	88.98	90.19	93.65	94.11	**95.25**
4	92.00	95.33	97.15	97.35	99.45	98.09	99.31	**99.96**
5	99.33	99.91	99.91	**100**	**100**	**100**	**100**	**100**
6	77.28	90.19	97.08	96.12	98.29	98.93	99.02	**99.48**
7	85.88	79.87	95.72	89.68	98.44	96.75	96.71	**98.46**
8	84.52	80.88	94.95	90.65	95.88	**98.34**	98.04	98.29
9	99.88	99.88	99.88	99.88	**100**	98.55	99.33	**100**
OA	90.02	91.99	95.45	96.36	98.07	99.02	98.99	**99.18**
AA	88.97	89.45	96.69	95.19	97.66	97.85	98.11	**98.81**
Kappa	0.83	0.86	0.95	0.95	0.97	**0.98**	**0.98**	**0.98**

accuracy than classifiers using only a single pixel's spectral data. 3D CNN-based models outperform spectral CNN, Deep RNN, and FCN due to their use of local pixel relationships, improving results for classes '1', '2', and '7', with class '5' reaching 100% accuracy.

Our proposed method outshines all others, with Scheme-1 and Scheme-2 models demonstrating effectiveness. The Scheme-2 model achieves the highest accuracy for classes '1', '3', '4', '6', '7', and '9', and overall scores (OA, AA, and Kappa) exceed all comparison methods, nearing a Kappa of 1.

Figure 3.7 (a) and (b) highlight the poor performance of spectral-based methods. Models using local area data like FCN, 3D CNN, M3D-DCNN, and SSAN yield cleaner results, as marked by white circles and boxes. Local feature consideration enhances the training of discriminative models. Our proposed models reduce noise-like misclassifications, maintain clear boundaries, and improve classifications in small regions. In summary, our Scheme-2 model is superior.

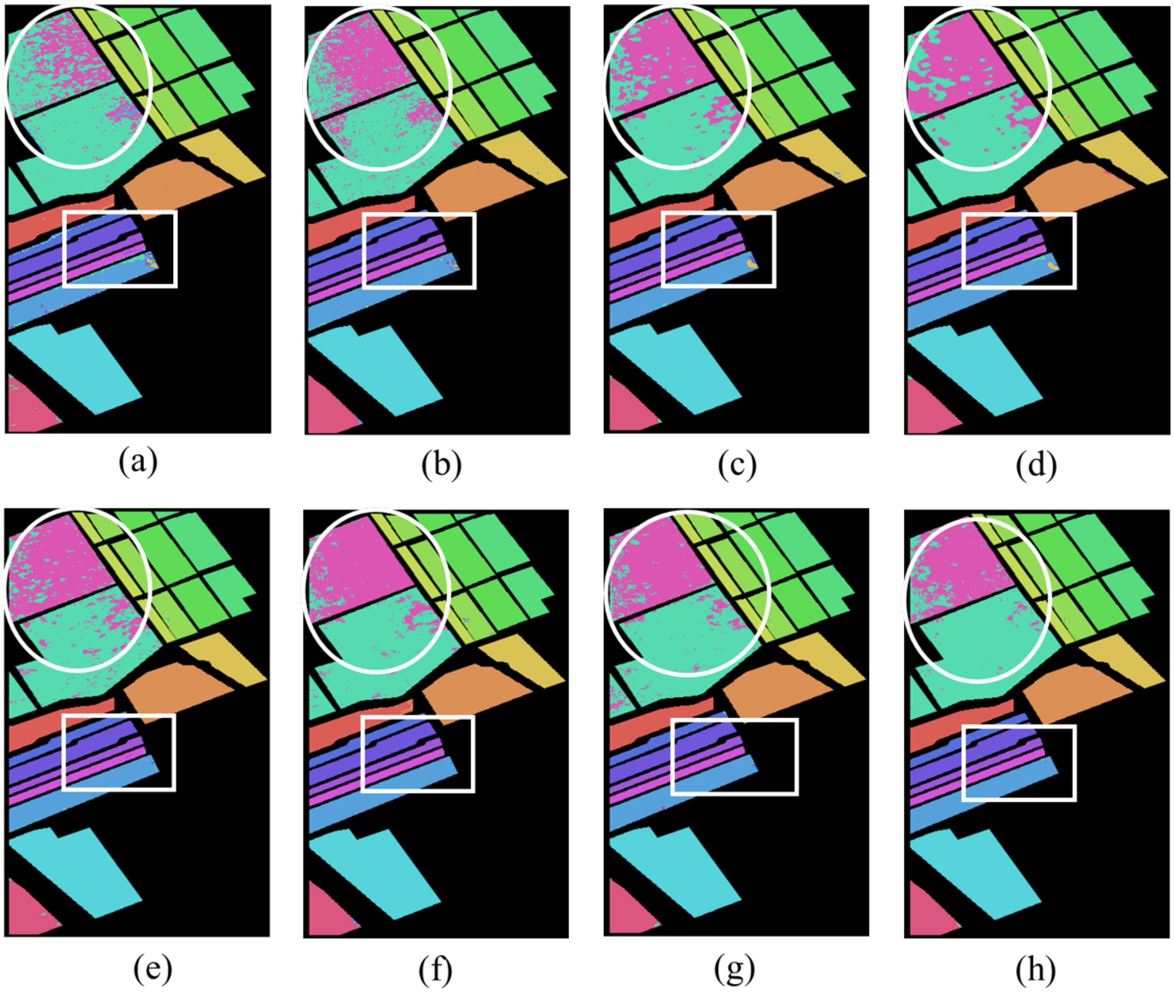


Figure 3.8 Classification maps for the SA dataset via various methods: (a) Spectral CNN. (b) Deep RNN. (c) FCN. (d) 3D CNN. (e) M3D-DCNN. (f) SSAN. (g) First Proposed Scheme (Ours). (h) Second Proposed Scheme (Ours).

SA Dataset

Figure 3.8 and Table 3.3 present the classification outcomes of various methods, highlighting the best results. For this dataset, which is highly homogeneous, we set the patch size to 7 for all methods, including 3D CNN, FCN, SSAN, and our proposed models. Other settings remain the same as previously mentioned.

Table 3.3 shows that spectral-based methods like spectral CNN and deep RNN, which input

Table 3.3 Quantitative performance metrics, including OA, AA, and Kappa, along with the accuracies for each class, were evaluated for various classification methods for the SA dataset. Best results are highlighted.

Classes	Methods							
	spectral CNN	Deep-RNN	3D CNN	FCN	M3D-DCNN	SSAN	Scheme-1(ours)	Scheme-2(ours)
1	98.30	**99.88**	99.01	95.61	98.21	98.42	98.13	98.32
2	98.91	99.81	98.90	99.69	**99.96**	99.97	98.97	98.99
3	96.98	98.36	98.02	98.13	99.64	98.14	**100**	**100**
4	98.78	98.71	98.89	97.71	**99.78**	97.95	99.11	99.35
5	97.86	98.48	99.11	98.48	**99.46**	98.09	99.08	99.34
6	99.89	99.89	**100**	98.3	99.61	**100**	99.79	**100**
7	99.23	99.74	99.13	98.32	99.59	**99.94**	98.98	99.63
8	80.15	84.15	88.87	**94.91**	92.37	91.82	93.07	93.18
9	98.71	99.43	99.83	99.39	99.80	**99.97**	99.65	99.76
10	88.39	96.51	96.89	95.92	**98.41**	96.82	96.79	97.71
11	88.88	97.26	96.99	96.28	97.68	**99.24**	98.03	98.23
12	97.73	98.99	99.33	97.53	98.75	**99.47**	99.01	98.85
13	96.17	99.43	98.79	97.77	99.08	**100**	**100**	**100**
14	94.08	97.61	98.48	97.24	**98.73**	98.66	98.23	98.16
15	56.12	72.61	82.26	**90.42**	86.67	89.15	88.69	89.08
16	97.69	**99.01**	97.75	85.28	94.23	97.80	93.67	93.66
OA	89.46	92.95	95.06	94.71	95.83	97.23	97.19	**97.92**
AA	91.99	95.24	97.01	96.31	97.62	**97.84**	97.58	97.67
Kappa	0.87	0.92	0.94	0.94	0.94	**0.97**	**0.97**	**0.97**

spectral signals directly, do not yield satisfying results, especially for class 8' and 15'. These outcomes stem from unbalanced training samples and complex spectral situations. Despite this, deep RNN achieves the best result in class '16' due to its handling of band-to-band correlation in a spectral signal, showcasing its strength in managing sequences with global dependency.

HSI features pixels from different classes with similar spectral signals and pixels from the same class with significant variations. As Figure 3.8 (a) and (b) demonstrate, disregarding neighboring pixel correlations leads to noise-like misclassification. However, methods incorporating spatial data yield promising results, as seen in Figure 3.8 (e) and (f), where white circle-marked regions

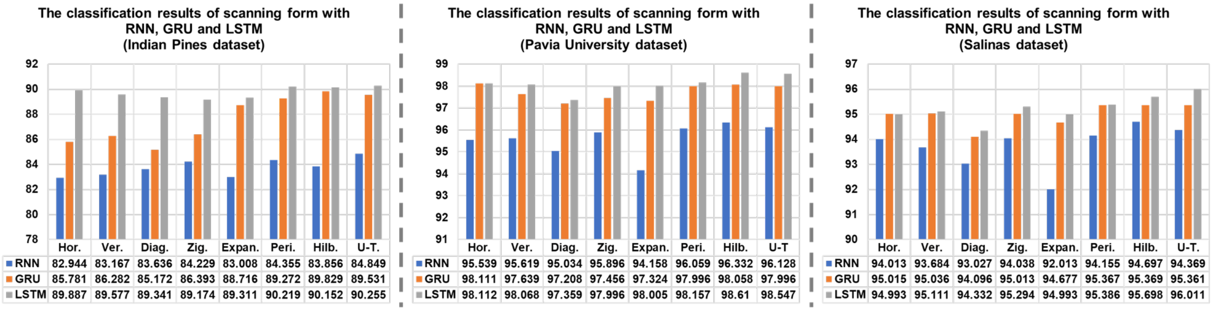


Figure 3.9 Classification outcomes using scanning patterns of RNN, GRU, and LSTM across three datasets.

are clean, and errors appear object-like instead of noisy. Nevertheless, adding spatial information sometimes reduces classification accuracy, as class ‘16’ results show.

Our approach exhibits superior OA and Kappa, achieving 100% accuracy for classes ‘3’, ‘6’, and ‘13’. As displayed in Figure 3.8, our methods consistently yield accurate and smooth results. Local spatial information use leads to cleaner region outlines. The SSAN method, which accounts for spatial attention, also yields excellent results in many classes, indicating that spatial attention reduces the negative impact of interfering pixels within an HSI patch. Our methods and SSAN yield similar results, with only about 0.7% OA difference, as shown in Table 3.3.

3.6.3 Other Analysis 1) to 8)

1) Performance on Different scanning patterns with RNN, GRU and LSTM

Unlike previous studies, we additionally employed separate RNN, GRU, and LSTM units for our experiments. The results obtained for the three datasets are conducted in single scanning order following Figure 2.4. The experimental performances are illustrated in Figure 3.9.

Figure 3.9 mirrors the findings of earlier research, demonstrating that regardless of whether we use GRU or LSTM, the scanning patterns with non-adjacent pixel connections, such as the horizontal and vertical forms, yield lower classification accuracy. In contrast, the perimeter, Hilbert, and U-Turn scanning patterns, which maintain complete spatial continuity, achieve significantly higher accuracy.

LSTM, one of the novel variants of recurrent units, shows better results. It can be attributed to its unique design, which incorporates gates for suitable feature selection during the recurrent procedure. It is the primary reason we replaced all simple RNN units with LSTM units.

2) Performance on Combining Different scanning patterns

In this study, we investigated a total of eight different scanning patterns. Among them, the U-Turn scanning pattern with eight orders consistently yielded the best results. Based on this finding, we explored the impact of combining different scanning patterns. Each scanning pattern was extended into eight scanning orders, as listed in Tables 2.1 and 2.2. Subsequently, we combined these multiple scanning orders with RNNs for evaluation. For instance, the U-Turn and Horizontal scanning patterns have eight orders each. We processed them separately using the same operation and then combined them, resulting in sixteen scanning orders in this scenario.

The results of these scenarios are presented in Table 3.4. This analysis allowed us to examine the performance of the different combinations and gain insights into the effectiveness of leveraging multiple scanning patterns to improve classification results.

Our findings indicate that randomly mixing different scanning patterns can decrease the classifier's accuracy. Furthermore, incorporating more scanning patterns increases the number of training parameters, showing that such a combination is not an efficient way to improve the classifier. As a result, we chose to proceed with only the U-Turn form for the subsequent experiments.

3) Effects of Different Feature Combination Manners

As discussed earlier, each scanning order in a scanning pattern relates to a single image patch. It is crucial to amalgamate all features from the potential scanning orders appropriately. We evaluated

Table 3.4 Performance outcomes when merging various scanning patterns. Best results are highlighted.

scanning patterns								Datasets		
U-Turn	Hilbert	Perimeter	Expansion	Zigzag	Diagonal	Vertical	Horizontal	IP	PU	SA
✓								**95.38**	**99.18**	**97.92**
✓	✓							95.31	99.05	97.81
✓		✓						95.04	98.85	97.56
✓			✓					94.05	96.55	96.08
✓				✓				95.22	99.04	97.80
✓					✓			95.04	98.82	96.99
✓						✓		95.01	98.33	97.67
✓							✓	94.97	98.91	97.89

the existing combination methods (SS, LLC, and FLC) available in the discipline, as shown in Figure 3.2. Additional experiments were conducted to ascertain the optimal combination approach for managing the complementary information among scanning orders. Note that this experiment employs eight complementary scanning orders for each scanning pattern.

This examination employs all these methods with LSTM for various combinations. The classification results are individually depicted in Figure 3.10. The FLC combination approach persistently outperforms the SS and LLC methods, emphasizing the importance of the correlational relationship among various scanning features in generating more discernible representations. Compared with the LLC combination approach, FLC involves fewer eight steps for the LSTM than the considerably longer (hidden size × 8) steps in the LLC method. This extended sequence for the LSTM to handle detrimentally affects the outcomes. The SS combination method directly amalgamates the weighted features, overlooking the correlation among these features. Consequently, the SS method generally fails to deliver satisfactory results.

Analysis from points 4) to 8) that follows is based on the Scheme-2 model using the U-Turn scanning pattern.

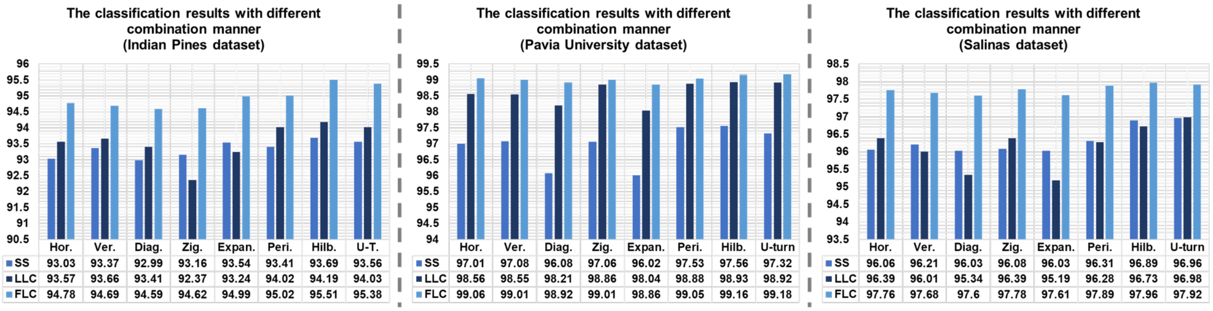


Figure 3.10 Classification outcomes from different combinations of scanning patterns across three datasets.

Table 3.5 Total KL-Divergence Loss associated with the usage of varying numbers of scanning sequences. Best results are highlighted.

	The use of number of scanning directions				
	0	1	2	4	8
IP	2.2780	1.3256	0.9935	0.8841	**0.7846**
PU	3.7278	1.8250	1.4513	1.2117	**1.0374**
SA	2.3706	0.9871	0.9698	0.8211	**0.7917**

4) The Feature Visualizations by T-SNE

Deep learning approaches strive to forge novel feature representations as a substitute for the original input, a process termed representation learning. Most prevailing methods train a model to morph the input into a unique output representation for HSI classification.

To validate the effectiveness of our method, we use the T-SNE [72] technique to visualize how the learned features from our model distribute. The visualization results for the IP, PU, and SA datasets are shown in Figure 3.11. Two points to note from these figures are: firstly, in the T-SNE results, clusters of different colors may be close, but this does not necessarily imply similarity. It is due to the inner workings of T-SNE. Secondly, clusters of the same color may be distant from each other because T-SNE aims to maintain local structure and thus achieves a local minimum.

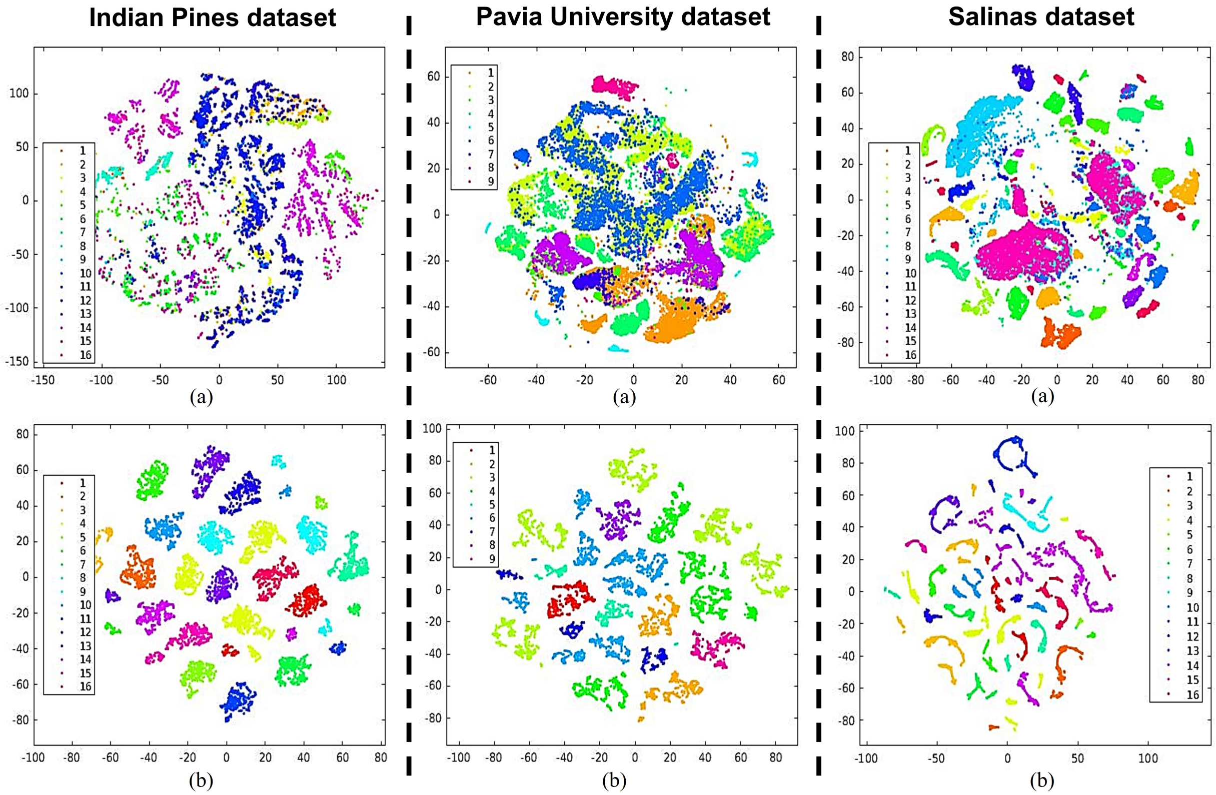


Figure 3.11 T-SNE visualizations for three datasets: a) Original samples and b) Feature representations learned via our method, depicted through T-SNE.

These figures clearly show that our model can distinguish the learned feature representations of the original pixels. Even with unbalanced samples selected for visualization, the results are encouraging and satisfactory. Through T-SNE, the new feature representations become grouped and distinct, with several clusters marked by the same color appearing more cohesive and smooth.

Additionally, we calculated the total KL divergence loss for each dataset using varying numbers of scanning directions, and the results are shown in Table 3.5. As the number of scanning sequences increases, the loss tends to decrease, supporting the feasibility of the multiscanning strategy.

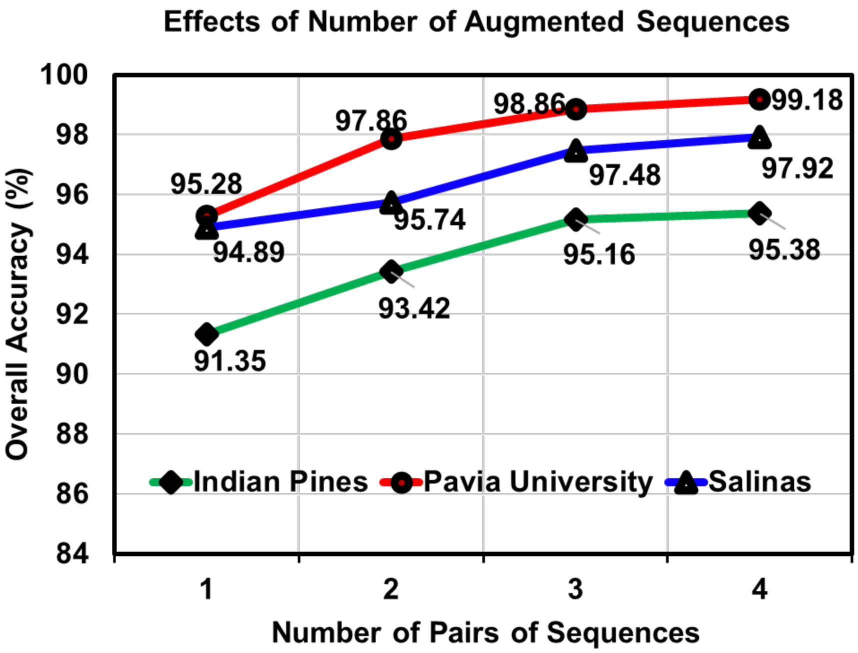


Figure 3.12 Impact of varying the quantity of augmented sequences across three datasets.

5) Effects of Multiple Scanned Pixel-Sequences

In our experiments, we prefer ‘scanning’ over ‘sorting,’ as ‘sorting’ could potentially neglect the continuity between individual pixels. For instance, when a 5×5 patch is reshaped into a 1×25 sequence, there would be P_{25}^{25} sorting strategies, many meaningless and superfluous. Therefore, we employ the scanning operation to maintain the intrinsic structure and continuity within a spatial patch and among each pixel.

Data augmentation, commonly used in computer vision, enhances network performance by expanding input information. Methods like rotation, mirroring, and flipping augment images for image-based networks like CNN. Similarly, our multiscanning approach can be seen as a feature augmentation operation, producing multi-directional views for sequence-based networks like RNNs. These augmented sequences increase the information amount for these networks.

Further experiments explore how the number of pixel-sequences used affects system performance. In the Scheme-2 model, we progressively add four pairs of spatial sequences into the training, each containing two pixel-sequences. Figure 3.12 illustrates how the quantity of input data influences the final classification results.

By increasing the number of input sequences, we observe an improvement in the classification accuracy, confirming the feasibility of our methods and ideas. However, we also note that the accuracy of our approach on the SA dataset does not significantly increase with the increased number of sequences. This observation can be attributed to the high homogeneity and large size of the SA dataset. In this dataset, small HSI patches often contain pixels of the same class with a higher probability, which reduces the impact of additional sequences on classification accuracy.

6) Effects of Using Attention Mechanism

Our models integrate an attention layer inspired by the concept of attention mechanisms. This approach assigns distinct weights to various scanning orders based on their significance. A single-layer neural network and a $Softmax$ layer, as specified in Equation (3.6) and (3.7), determine these weights. To assess the enhancements provided by the attention mechanism, we carried out additional experiments, the results of which are presented in Table 3.6.

Including the attention layer enables the network to accord higher importance to vital scanning orders while reducing the emphasis on the less crucial ones instead of treating them equally. It results in more distinguishable outputs and enhances classification accuracy. Introducing an attention layer leads to increased accuracy in both Scheme-1 and Scheme-2. This improvement is particularly evident in the IP dataset, likely due to its relatively small size. With a patch size of five, there will be more interfering pixels, suggesting that certain scanned pixel-sequences negatively influence training. The attention mechanism can mitigate the impact of these adverse pixel-sequences, leading to superior performance.

Table 3.6 Implications of Implementing Attention Mechanism. Best results are highlighted.

	The use of attention layer			
	Scheme-1		Scheme-2	
	without	with	without	with
IP	94.33	**95.13**	94.57	**95.38**
PU	98.51	**98.99**	99.01	**99.18**
SA	96.76	**97.19**	97.59	**97.92**

7) Comparison Between RNN and 1D CNN for Dealing with Pixel-Sequence

We suggest using RNNs like GRU and LSTM for processing pixel-sequences, which have an input channel of spectral data dimension C and a sequence length equivalent to the number of flattened pixels ($patchsize \times patchsize$). Alternatively, the sequence could be processed using a spectral-spatial 1D CNN, with an input channel size of $patchsize \times patchsize$ and a sequence length of spectral dimension C.

To demonstrate the advantage of RNNs, we conducted further experiments. We manually set the number of parameters nearly equal in these experiments for the proposed method and the 1D CNN. Moreover, we only used one feature extraction layer for LSTM and 1D CNN. The settings for the 1D CNN are as follows: The kernel size is 3, with 128 kernels and a kernel stride of 1. In the case of LSTM, a single layer with 128 hidden units is employed. The experimental results are displayed in Table 3.7, and the comparison of validation accuracy on three datasets are shown in Figure 3.13.

The results show that RNNs can process pixel-sequences more efficiently with fewer parameters than 1D CNN, demonstrating RNN's superiority when handling sequence-like data.

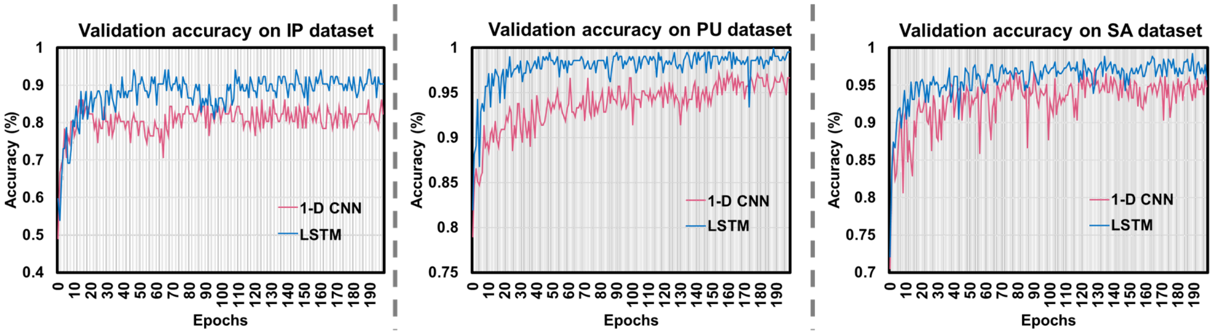


Figure 3.13 Comparison of validation accuracy between RNN (LSTM) and 1D CNN when handling a pixel-sequence across three datasets.

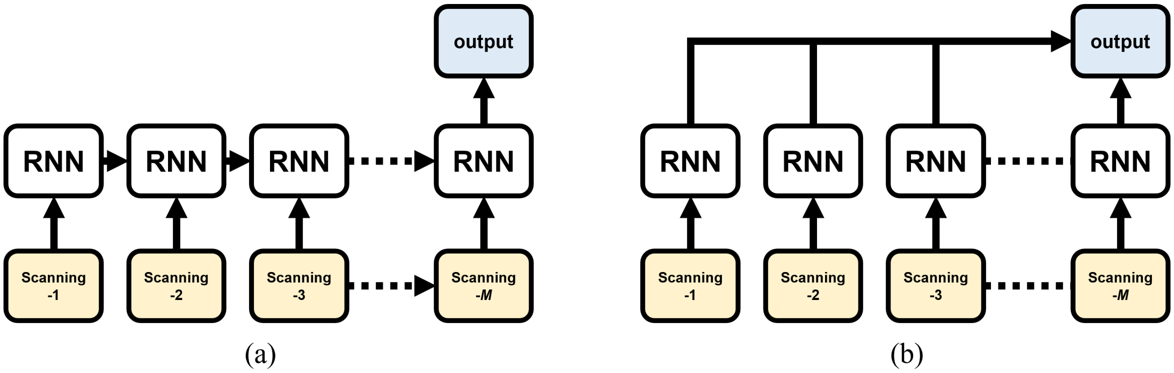
Table 3.7 Comparative analysis between RNN and 1D CNN in the context of handling pixel-sequences. Best results are highlighted.

	1D CNN		LSTM	
	Number of Parameters	OA(%)	Number of Parameters	OA(%)
IP	491,281	81.267	416,618	**90.324**
PU	164,874	96.214	116,607	**97.676**
SA	501,009	95.629	431,478	**96.834**

8) Comparison of Cascaded and Parallel Design for Multiscanning

In order to evaluate the combination methods within the multiscanning strategy, we have proposed two approaches, as illustrated in Figure 3.14: 1) Cascaded manner, as proposed in [92]: each scanning order is fed in succession, meaning the output from one scanning order, processed through one RNN, becomes the input for the next RNN, which is prepared for the subsequent scanning order. 2) Parallel manner: each scanning order is separately processed with its respective RNN, and the output from each RNN is concatenated.

To verify the performance of the cascaded and parallel methods of feature combination for multiscanning, we performed a series of experiments using 8 U-Turn scanning orders. The results of these experiments are presented in Table 3.8. It is important to note that these experiments were



(a)

(b)

Figure 3.14 The comparison of (a) cascaded and (b) parallel design of feature combination for multiscanning strategy

Table 3.8 The overall accuracy of deploying eight U-Turn scanning orders for multiscanning feature combination. Best results are highlighted.

	Multiscanning combination manner	
	Cascaded manner	Parallel manner
IP	92.99	**94.33**
PU	97.56	**98.51**
SA	95.21	**96.76**

conducted purely without additional attention mechanisms. Additionally, all RNNs were replaced with LSTM for this set of tests, and Scheme-1, which separates multiscanning, was employed.

The results demonstrate that the parallel method outperforms the cascaded approach. We hypothesize that the parallel method's superiority stems from its ability to process each scanning order independently and equally, thereby preserving the unique information from each order. The parallel method ensures a more unbiased feature representation by treating each scanning order with equal importance. This unbiased representation, in turn, amplifies the effectiveness of the multi-directional approach, expanding the diversity of the feature pool, thereby leading to superior results.

An additional concern regarding the cascaded method is the difficulty in determining the order of the scanning orders. It could inadvertently introduce bias into the process, potentially skewing the results. Since the cascaded method processes the scanning orders in a sequence, the decision on the order in which they are processed could disproportionately influence the outcome, with the later orders having more weight in the final representation. It introduces an undesired source of variability and potential bias into the process.

3.7 Conclusions with Further Discussion

3.7.1 Conclusions

This chapter introduces an RNN-centric approach for HSI classification that leverages a multi-scanning strategy. This strategy ingeniously transforms a localized image patch into a plethora of complementary directional sequences via RNN, thereby capably handling the spectral-spatial information. Moreover, we conducted assessments on the attention of diverse scanning patterns on the outcomes of HSI classification. Through Bi-RNN, four pairs of local directional sequences were processed and fused using concatenation. We conducted a comparative analysis of various concatenation methods to understand their impacts on classification results.

The multiscanning strategy is a feature augmentation operation that enhances the RNN model's data volume. The proposed networks are adept at extracting spectral and spatial information from HSI patches, harnessing the spatial dependencies within the spatial patch in conjunction with the spectral data from each pixel.

This chapter considers our methodology highly efficient, primarily due to its usage of fewer parameters, especially when juxtaposed with 3D CNN-based methods. Further, we propose our strategy as a potential substitute for CNNs, hinting at its capacity to pave the way for future data analysis and research endeavors centered on constructing spatial and spectral information using solely RNN.

The main contributions of this chapter can be summarized as:

1) We propose to validate the effectiveness of two approaches: 'Scanning Order-Based Attention' and 'RNN-Based Multiscanning Feature Fusion'. These methods build upon the research presented in Chapter 2 and achieve improved results, with enhancements ranging from 2% to 5%.

2) Compared to other baseline methods, our approach achieves an overall accuracy improvement of 8% to 25%. Against state-of-the-art methods, we observe improvements ranging from 1% to 5%. Notably, the enhancements are more pronounced in the urban scene dataset, particularly the PU dataset. We attribute this to the geometric layout of urban features like buildings and roads, which our multiscanning-based RNN is better able to recognize. However, in agricultural scenes, such as the IP dataset and SA dataset, where geometric design is less prominent, our method does not show significant improvements.

3) Compared to other methods, our approach can save approximately 2% to 62% in processing time, and our model size is significantly smaller—between 2 to 20 times lighter. This demonstrates the efficiency of our method.

4) This chapter delves deeper into the research on multiscanning-based RNNs, focusing on the attention mechanism and feature fusion. It demonstrates the potential for enhancing HSI classification results.

3.7.2 Further Discussion

While our proposed multiscanning-based RNNs demonstrate improved performance, it is essential to acknowledge the intrinsic limitations of RNNs, particularly their ordering mechanism. In this context, we highlight three issues for future research in combination with popular Transformer [13, 73] models:

1) RNNs exhibit bias, which can weaken the influence of the central pixel's information, despite its importance in determining the final feature.

2) RNNs face challenges when dealing with larger HSI patches, as the pixel-sequences may contain more pixels with different class labels than the central pixel. They can dominate the final decision of the output features.

3) RNNs need to learn the correlation among different scanning features effectively. The feature fusion by RNNs is not appropriate.

Addressing these issues through integrating Transformer models could be a promising direction for future work. Transformers have shown remarkable performance in capturing long-range dependencies, addressing bias, and modeling correlations, making them a potential solution for overcoming the limitations of RNNs in HSI classification tasks.

Therefore, Chapter 4 addresses the abovementioned issues and further develops our ideas.

Chapter 4

Hyperspectral Image Classification Using Multiscanning-Based RNN-Transformer

In this chapter, we aim to tackle the limitations of RNNs used in previous research by incorporating Transformer models into our multiscanning-based approach. We intend to improve the performance and robustness of our method by leveraging the strengths of Transformers, such as their ability to capture long-range dependencies and model correlations effectively.

It starts with a review of limitations in our previous work. After that, it continues with a general introduction of the Transformer with its self-attention mechanism. Following the introduction, a brief preview of the proposed procedure is presented, outlining the key steps and objectives.

Then, it delves into the detailed progress of our approach. It describes the integration of Transformers into our multiscanning-based framework, highlighting the modifications and enhancements.

Finally, we perform various experiments to validate the feasibility of our approach, followed by an analysis to support our ideas.

4.1 Introduction

4.1.1 Remained Problems in Previous Chapter

Although RNN-based HSI classifiers have shown promising results in previous studies, specific critical issues still need to be addressed.

A prominent concern is the dependency on the output of the final step as the ultimate feature for classification, while the other steps' importance is overlooked. Most have implemented attention-weighted summation of each step's output, while these may inadvertently favor later pixels, as indicated in [37]. It could dilute the significance of information from the central pixel, which should play a key role in defining the final feature. This dilution could contribute to misclassifications, particularly when the cropped patch's central pixel shares the same label [88].

Additionally, handling larger HSI patches presents another challenge. In such scenarios, RNN models may falter as the pixel-sequence may contain more pixels with different class labels than the central pixel. This imbalance could influence the final decision regarding the output features disproportionately. The rigidity of the RNN due to its sequence-dependent nature compounds this issue. For example, interfering pixels are predominantly located towards the pixel-sequence's end. In that case, the final step's output features might be negatively affected and fail to represent the HSI patch's land-cover characteristics accurately.

Furthermore, as employed in previous work [91], the multiscanning strategy processes an HSI patch with multiple scanning orders to capture complementary contextual features. However, the importance of each scanning order needs to be adequately accounted for, intensifying the issues discussed above. Some scanning orders may place interfering pixels later in the pixel-sequence, which can negatively influence the extracted features. Solely averaging the outputs from different scanning orders might decrease feature discriminability.

Consequently, it is paramount to address these challenges by developing innovative methods that aptly consider the importance of each step, manage larger HSI patches efficiently, and fine-tune the multiscanning strategy to enhance feature discriminability.

4.1.2 Transformer

Recently, the Transformer model [73] has gained significant attention and applied to various domains because it captures interdependencies within sequences using self-attention mechanisms. In the field of HSI classification, the vision Transformer (ViT) [13] has been adopted and improved upon from different perspectives, such as pixel-sequence, patch-sequence, or band-sequence.

The initial ViT-based HSI classification model was suggested by Hang *et al.* [32], featuring an enhanced Transformer encoder specializing in band features. The Transformer was then amalgamated with the CBAM attention block [77] by Qing *et al.* [56] to augment spectral attention. An innovative deep Transformer-in-Transformer (TNT) module was introduced by Gao *et al.* [16] for patch-level and pixel-level feature extraction. The concept of a bidirectional Transformer encoder enabling dynamic and versatile cropping regions was put forward by He *et al.* [26]. The masked auto-encoding spectral-spatial Transformer (MAEST), encompassing both reconstruction and classification paths to enhance Transformer features, was put forth by Ibañez *et al.* [36]. Lastly, the HSI Transformer (HiT) was conceived by Yang *et al.* [80], wherein convolution operations were fused into the Transformer to allow the inclusion of both spectral and spatial features [93].

Unlike the sequential processing of HSI pixel-sequences by RNN models, the Transformer model handles all pixels concurrently; it grants individual attention to each pixel from all others, facilitating the creation of a weighted representation that encapsulates comprehensive information. This approach enables the Transformer to assign more precise and diverse attention weights to all pixels, thereby circumventing the neglect of crucial information [76]. Consequently, compared to RNN models, which rely on the output of the last step, the Transformer is better positioned to allow the central pixel to influence the final classification feature significantly.

Moreover, the Transformer assigns rational and unequal weights based on the specific scanning order when handling pixel-sequences with varying scanning orders. It empowers the model to evaluate the impact of each order and produce a more discriminative feature fusion. Consequently, the Transformer presents a more flexible and efficient strategy for merging information from different scanning orders in HSI classification tasks.

These advancements in utilizing the Transformer model for HSI classification demonstrate its potential to address the limitations associated with RNN models and further enhance the performance and discriminative power of HSI classification algorithms.

4.1.3 Limitations in Transformer

While the Transformer model has demonstrated remarkable performance in various tasks, including HSI classification, recent studies have highlighted its lack of recurrent modeling limitations. It has been shown that incorporating recurrent modeling in the Transformer can lead to further improvements [11].

Recurrence is essential for capturing crucial properties of input sequences, such as structural representations and positional embeddings. RNN-based models have successfully captured these properties, as they naturally encode sequential information and dependencies [61, 69, 70]. These are areas where the self-attention mechanism in the Transformer falls short. The absence of recurrence in the Transformer may hinder its ability to fully capture and exploit the inherent sequential nature of specific data.

Furthermore, research studies [8, 24] have suggested that the representations learned by Transformer-based and RNN-based encoders can complement each other. It is possible to achieve a more comprehensive and practical modeling approach by leveraging the strengths of both architectures.

Considering these findings, it becomes evident that while the Transformer model has shown significant promise, there is still value in exploring the combination of recurrent modeling with the self-attention mechanism. Integrating recurrence into the Transformer framework may allow capturing both the global context and the local dependencies within sequences, leading to enhanced performance and more comprehensive representations.

4.1.4 Summarization

This research aims to augment the multiscanning strategy [91] by integrating a uniquely tailored RNN-Transformer model adept at encapsulating the multi-sequential attributes of HSI pixels. We proposed an innovative multiscanning controlled positional embedding to accommodate spatial contextual dependencies at varying positions.

The RNN-Transformer model combines the strengths of both RNN and Transformer architectures, leveraging the ordering bias of RNN and the self-attention weights of the Transformer for feature generation. The model can determine the positive or negative impact of different scanning orders by incorporating scanning order-based attention.

We introduce a spectral-spatial-based soft mask in the self-attention layer to address the influence of interfering pixels, effectively mitigating their effects.

Through these innovations, this study aims to improve the accuracy of HSI classification, enhance the multiscanning strategy, and provide a robust framework for analyzing hyperspectral data.

4.2 Self-Attention Mechanism

At the core of the Transformer model resides the self-attention mechanism. This mechanism processes each constituent of sequential data **S**, yielding three distinctive features through the application of three varied linear transformations, specifically, Query (**Q**), Key (**K**), and Value (**V**). The resulting output from the self-attention mechanism, denoted by **F**, is subsequently computed as follows:

$$\mathbf{F} = \mathcal{SA}(\mathbf{S}) = Softmax\Big(\frac{\mathbf{Q} \odot \mathbf{K}^\top}{\sqrt{d_k}}\Big) \odot \mathbf{V}, \tag{4.1}$$

where d_k is the feature dimension of **K**, and $\mathcal{SA}()$ is denoted as self-attention operation.

The self-attention mechanism can be enhanced using multi-head attention, where the attention operation is performed in parallel. It allows the model to focus on different aspects of the input sequence.

The output of the self-attention mechanism is processed through a stack of feedforward neural networks (FFNs). It consists of several linear projection layers with a non-linear activation function $\phi()$, such as ReLU, as:

$$\mathcal{FFN}(\mathbf{F}) = \phi\Big(\mathbf{F} \odot \mathbf{W}_1 + \mathbf{b}_1\Big) \odot \mathbf{W}_2 + \mathbf{b}_2 \tag{4.2}$$

where **F** is the input to the $\mathcal{FFN}()$, and $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are the learnable parameters.

The Transformer model also includes residual connections [27] and layer normalization. The residual connection adds the input of the layer to its output, and layer normalization normalizes the output of each layer. Through the transformation, the Transformer can concurrently compute the long-range dependencies amongst sequence elements, thereby overcoming the restrictions of conventional RNN models in specific tasks.

It is essential to highlight that the SA layer lacks positional information and cannot leverage sequential information. Hence, to rectify this, positional information is encoded into the original input through the following formulation:

$$\mathbf{S} = \gamma \times \mathbf{S} + \delta \times \mathbf{PE}, \tag{4.3}$$

where **PE** represents positional embedding features, maintaining the same size as **S**. It is typically acquired using either the sine or cosine function with a fixed value, as follows:

$$\mathbf{PE}\Big(pos, 2i\Big) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{4.4}$$

$$\mathbf{PE}\Big(pos, 2i+1\Big) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{4.5}$$

where pos represents the position in the sequence, i represents the dimension index, and d_{model} represents the dimension of the model. Some works set $\mathbf{PE()}$ as additional, learnable parameters. Meanwhile, the $\gamma = \delta$ combination term is set commonly [13].

The Transformer model gains the ability to capture positional information and understand the order of elements within the sequence by adding positional encoding to the input sequence. It allows the model to handle sequential data better and effectively process sequences of varying lengths.

4.3 Multiscanning Controlled Positional Embedding

Correctly encoding positions is critical as positional embedding greatly affects the Transformer's comprehension of sequence structure and order [45]. Our observation, inspired by the multiscanning strategy, suggests that different input arrangements in a sequence can yield diverse output features for sequential models. As we previously demonstrated, altering the current input $\mathbf{S}_m^{(i,j)}[s]$ during training results in unique $\mathbf{Y}_m^{(i,j)}[s]$ values. This setup of RNNs in the example resembles the positional embedding in the Transformer. Therefore, modifying the positional information in the Transformer is expected to generate different output features. We may achieve more distinguishable outputs by suitably combining these features. Therefore, we improve the positional embedding using the multiscanning strategy.

We aim to leverage different scanning orders to capture diverse spatial contextual dependencies and improve the discriminative power of the Transformer model by integrating the multiscanning strategy into the positional embedding. This approach allows us to exploit input arrangement variations and enhance representation learning. We can better understand the input sequence and generate more discriminative output features by effectively blending the output features from different scanning orders.

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

$$\mathbf{PE}^{(i,j)} = \left\{\mathcal{RNN}\left(\mathbf{S}_m^{(i,j)}\right)\right\}, \quad m = 1,2,...,\mathcal{M}, \tag{4.6}$$

where $\mathcal{RNN}()$ is the recurrent operation, listed in Chapter 3.

In this research, we introduce the concept of multiscanning controlled positional embedding ($\mathbf{PE}^{(i,j)}$) generated using the U-Turn scanning pattern with $M = 8$, as illustrated in Figure 2.9. This choice of scanning pattern aligns with previous works and serves as the basis for incorporating positional information in our approach.

4.4 RNN-Transformer Encoder (RT)

The inherent ordering bias in RNNs can lead to the attenuation of central pixel information in HSI classification tasks.

We incorporate the Transformer's self-attention mechanism to address this limitation, which permits a reevaluation of the significance of various attributes within the RNN's output [14, 78].

Figure 4.1 provides a comprehensive illustration of the RT encoder. The sequence of operations in a single layer of the RT encoder can be summarized as follows.

Initially, RNN is employed for both positional and feature embeddings, as demonstrated below:

$$\mathbf{PE}_m^{(i,j),l} = \mathcal{RNN}\left(\mathbf{S}_m^{(i,j),l}\right), \tag{4.7}$$

where $l = 1,...,L$ denotes the l-th layer within the RT encoder.

Subsequently, we implement layer normalization (denoted as $\mathcal{LN}$) and a skip connection to normalize and boost the features.

$$\tilde{\mathbf{S}}_m^{(i,j),l} = \mathcal{LN}\left(\gamma_m^l \times \mathbf{S}_m^{(i,j),l} + \delta_m^l \times \mathbf{PE}_m^{(i,j),l}\right). \tag{4.8}$$

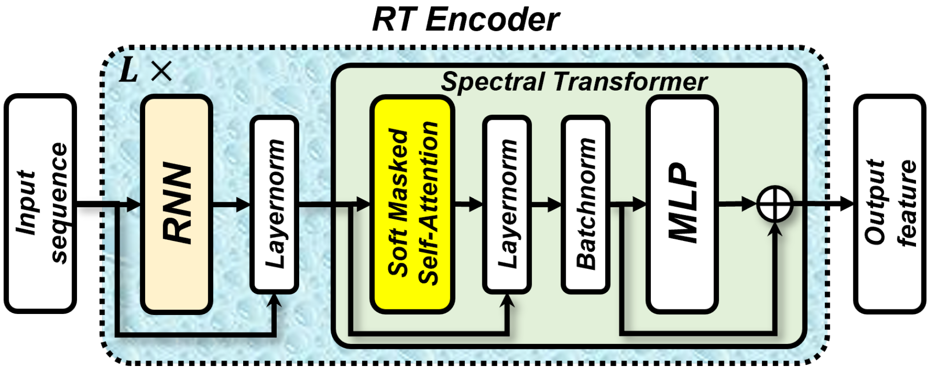


Figure 4.1 In-depth structure of the RT encoder.

where $\gamma_m^l, \delta_m^l \in \mathbb{R}^1$ are learnable parameters involved in the linear combination between two matrices, with the constraint that $\gamma_m + \delta_m = 1$.

Following this, the Spectral Transformer (ST) is deployed on $\tilde{\mathbf{S}}_m^{(i,j),l}$ with the purpose of recalculating the attention weights as follows:

$$\tilde{\mathbf{T}}_m^{(i,j),l} = \mathcal{ST}\left(\tilde{\mathbf{S}}_m^{(i,j),l}\right), \tag{4.9}$$

where $\mathcal{ST}()$ symbolizes the procedures executed within the Spectral Transformer.

4.4.1 Spectral Transformer (ST)

The Spectral Transformer includes various operations, particularly the Soft Masked Self-Attention module (SMSA). The embedded feature $\tilde{\mathbf{S}}_m^{(i,j),l}$ initially passes through the SMSA as follows:

$$\mathbf{F}_m^{(i,j),l} = \mathcal{SMSA}\left(\tilde{\mathbf{S}}_m^{(i,j),l}\right), \tag{4.10}$$

where $\mathbf{F}_m^{(i,j),l} \in \mathbb{R}^{p^2 \times d}$ represents the output feature from the $\mathcal{SMSA}()$ module. More details of $\mathcal{SMSA}()$ are provided in Section 4.6. We then conduct layer normalization and implement a skip connection like the previous steps:

$$\mathbf{T}_m^{(i,j),l} = \mathcal{LN}\left(\mathbf{F}_m^{(i,j),l} + \tilde{\mathbf{S}}_m^{(i,j),l}\right). \tag{4.11}$$

Finally, following the application of batch normalization ($\mathcal{BN}()$), these features are passed through a feed-forward layer to generate the output of the l-th layer:

$$\tilde{\mathbf{T}}_m^{(i,j),l} = \mathcal{MLP}\Big(\mathcal{BN}\Big(\mathbf{T}_m^{(i,j),l}\Big)\Big), \tag{4.12}$$

where $\mathcal{MLP}()$ represents multi-layer perceptron.

Optionally, the output from the l-th layer can be used as the input for the subsequent layer in the RT encoder, calculated using the following expression:

$$\mathbf{S}_m^{(i,j),l+1} = \tilde{\mathbf{T}}_m^{(i,j),l} + \mathbf{T}_m^{(i,j),l}. \tag{4.13}$$

Thus, for the last layer (L) of the RT encoder, the linear transformations are applied to $\tilde{\mathbf{T}}_m^{(i,j),L}$ to derive a novel feature representation $\mathbf{F}_m^{(i,j)}$:

$$\mathbf{F}_m^{(i,j)} = \mathcal{FFN}(\mathbf{F}) = \phi\Big(\mathbf{W}_1 \odot \tilde{\mathbf{T}}_m^{(i,j),L} + \mathbf{b}_1\Big) \odot \mathbf{W}_2 + \mathbf{b}_2 \tag{4.14}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are the corresponding parameters. The resulting $\mathbf{F}_m^{(i,j)}$ will then be utilized for the subsequent processes.

The RT encoder possesses a distinctive characteristic whereby each pixel is assigned more nuanced and diverse weights. It ensures that important information is preserved while irrelevant information is effectively disregarded. Moreover, the positional bias introduced by the RNN is effectively integrated with the Transformer in this encoder architecture. This combination enhances the model's ability to capture both sequential dependencies and global context, resulting in improved feature representations for HSI classification.

4.5 Feature Selection Layer (FS)

The objective of the FS layer is to generate a representation of the feature $\mathbf{F}_m^{(i,j)}$ that concentrates on the center step. It is achieved by assigning greater influence to the central element of $\mathbf{F}_m^{(i,j)}$ in the final feature representation $\mathbf{y}_m^{(i,j)}$ for the m-th scanning order. The intensified feature $\mathbf{y}_m^{(i,j)} \in \mathbb{R}^{1\times d}$ is computed as a weighted sum of all elements in $\mathbf{F}_m^{(i,j)}$:

$$\mathbf{y}_m^{(i,j)} = \sum_{s=0}^{p^2-1} \left(\mathbf{A}_m^{(i,j)}[s] \odot \mathbf{F}_m^{(i,j)}[s] \right), \tag{4.15}$$

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

$$\mathbf{A}_m^{(i,j)}[s] = Softmax\left\{ \left(\mathbf{F}_m^{(i,j)}[s] \right)^\top \odot \mathbf{F}_m^{(i,j)} \left[\frac{p^2-1}{2} \right] \right\}, \tag{4.16}$$

Through this operation, it aims to generate a center-oriented feature that effectively represents the cropped patch $\mathbf{X}^{(i,j)}$. It becomes crucial when the label of the patch is the same as the label of the central pixel, as it plays a significant role in the model training process. The generated center-focused representation enables the model to capture crucial information relevant to the HSI patch's characteristics by focusing on the central pixel. This approach enhances the model's ability to capture distinctive features and improve its performance in capturing the relevant characteristics of the HSI data.

4.6 Spectral-Spatial-Based Soft Masked Self-Attention (SMSA)

HSI classification aims to assign a meaningful land-cover category to the central pixel within a patch of HSI data. Randomly extracted from the original HSI data, these patches function as representative samples. However, interfering pixels carrying different labels within the same patch could potentially distort the spectral-spatial features intended for representation [65]. Hence, it becomes vital to devise an attention mask capable of highlighting helpful information from uniform pixels while simultaneously downplaying the impact of interfering pixels.

Most methodologies in the current literature emphasize the creation of a hard attention mask that applies binary values (0 and 1) to decide which features to keep or discard [4, 58, 81]. Never-

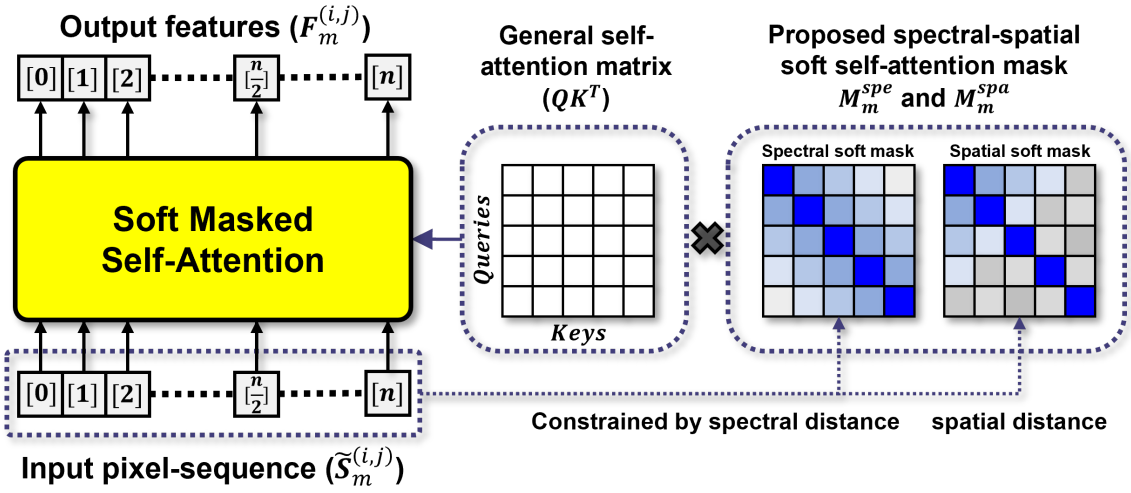

Output features ($F_m^{(i,j)}$)
[0] [1] [2] $[\frac{n}{2}]$ [n]
Soft Masked Self-Attention
General self-attention matrix (QK^T)
Queries
Keys
Proposed spectral-spatial soft self-attention mask M_m^{spe} and M_m^{spa}
Spectral soft mask
Spatial soft mask
Constrained by spectral distance
spatial distance
Input pixel-sequence ($\widetilde{S}_m^{(i,j)}$)

Figure 4.2 The SMSA module comprises the query-key attention matrix and soft spectral and spatial masks. The values designated for these two soft masks are constrained within the range [0,1].

theless, this approach harbors several shortcomings: 1) challenges in accurately setting the placement of 0s and 1s, 2) rigid binary values that could potentially exclude beneficial information or encompass adverse information, 3) limited regard for correlations between pixels, and 4) dependence on manual expertise. To circumvent these issues, we have fashioned a soft attention mask predicated on the similarity among pixels within a trimmed HSI patch, as depicted in Figure 4.2. This softer attention mask presents a more adaptable and nuanced method of capturing pertinent information, thus enhancing pixel differentiation and diminishing the effect of interfering pixels on the classification procedure.

Precisely, within a cropped patch ($\mathbf{X}^{(i,j)} \in \mathbb{R}^{p\times p\times C}$), the pair-wise distances between two pixels are calculated using the Euclidean distance, as shown in Equation (2.13):

$$\mathcal{D}^{spe}\left(\mathbf{x}^{(i_1,j_1)}, \mathbf{x}^{(i_2,j_2)}\right) = \left\|\mathbf{x}^{(i_1,j_1)} - \mathbf{x}^{(i_2,j_2)}\right\|^2, \tag{4.17}$$

where $i - \frac{p-1}{2} \leq i_1, i_2 \leq i + \frac{p-1}{2}$ and $j - \frac{p-1}{2} \leq j_1, j_2 \leq j + \frac{p-1}{2}$. The symbol $\|\cdot\|$ denotes the Euclidean norm. Based on these calculations, we can obtain a spectral distance matrix $\mathbf{D}^{spe} \in \mathbb{R}^{p^2\times p^2}$, which is a symmetric matrix consisting of all pair-wise distances $\mathcal{D}^{spe}(\mathbf{x}^{(i_1,j_1)}, \mathbf{x}^{(i_2,j_2)})$.

The values $\mathcal{W}^{spe}$ of the spectral-based soft mask are then determined using a Gaussian function:

$$\mathcal{W}^{spe}\left(\mathbf{x}^{(i_1,j_1)},\mathbf{x}^{(i_2,j_2)}\right)=\begin{cases}1, & \text{if } i_1=i_2 \text{ and } j_1=j_2\\ \exp\left(-\frac{\left(\mathcal{D}^{spe}\left(\mathbf{x}^{(i_1,j_1)},\mathbf{x}^{(i_2,j_2)}\right)\right)^2}{2\rho^2}\right), & \text{otherwise}\end{cases} \tag{4.18}$$

where ρ represents the average pair-distance in the matrix $\mathbf{D}^{spe}$. The weights assigned to longer spectral distances are small, and all weights fall within the range of 0 to 1. The resulting spectral soft mask, denoted as $\mathbf{M}^{spe} \in \mathbb{R}^{p^2\times p^2}$, is a feature matrix that contains all the corresponding pair-wise weights $\mathcal{W}^{spe} \in \mathbb{R}^1$.

Next, the spatial distance between two pixels can be calculated based on their spatial coordinates, employing a method such as spatial correlation [17]:

$$\mathcal{D}^{spa}\left(\mathbf{x}^{(i_1,j_1)},\mathbf{x}^{(i_2,j_2)}\right)=\left|i_1-i_2\right|+\left|j_1-j_2\right|, \tag{4.19}$$

where $|\cdot|$ represents the absolute value.

The maximum distance in the spatial domain is defined as follows:

$$\mathcal{D}^{spa}(max)=(p-1)\times 2. \tag{4.20}$$

Consequently, we establish the spatial distance matrix $\mathbf{D}^{spa} \in \mathbb{R}^{p^2\times p^2}$, which records the pair-distance $\mathcal{D}^{spa}(\mathbf{x}^{(i_1,j_1)},\mathbf{x}^{(i_2,j_2)})$. The values ($\mathcal{W}^{spa}$) of the spatial-based soft mask are calculated using the Subtract function as follows:

$$\mathcal{W}^{spa}\left(\mathbf{x}^{(i_1,j_1)},\mathbf{x}^{(i_2,j_2)}\right)=\frac{\mathcal{D}^{spa}\left(max\right)-\mathcal{D}^{spa}\left(\mathbf{x}^{(i_1,j_1)},\mathbf{x}^{(i_2,j_2)}\right)}{\mathcal{D}^{spa}\left(max\right)}. \tag{4.21}$$

The outcome is a soft spatial mask, symbolized as $\mathbf{M}^{spa} \in \mathbb{R}^{p^2\times p^2}$, which incorporates all the pair-wise weights $\mathcal{W}^{spa} \in \mathbb{R}^1$ that correspond to each pixel pair in the cropped patch.

In the final step, the soft spectral and spatial masks ($\mathbf{M}^{spe}$ and $\mathbf{M}^{spa}$) are amalgamated and incorporated into the self-attention layer, as articulated in Equation (4.1), facilitating the creation of soft features. The overall process of the $\mathcal{SMSA}()$ can be summarized as follows:

$$\mathbf{F}_m^{(i,j)} = \mathcal{SA}(\mathbf{S}_m^{(i,j)}) = Softmax\left(\frac{\mathbf{Q}_m \odot (\mathbf{K}_m)^T}{\sqrt{d_k}} \times \mathbf{M}_m^{spe} \times \mathbf{M}_m^{spa}\right) \odot \mathbf{V}_m, \tag{4.22}$$

where $\odot$ represents the element-wise multiplication (Hadamard product). $\mathbf{M}_m^{spe}$ and $\mathbf{M}_m^{spa}$ represent the spectral and spatial masks at the m-th scanning order, respectively.

These soft masks are generated leveraging the original input, which allows for precise detection and mitigation of disturbances caused by pixels.

Moreover, the SMSA can be expanded to a multi-head version where the input is transformed into features corresponding to multiple heads using different $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ matrices. The outputs from each head are then concatenated together to obtain the final representation:

$$\mathbf{F}_m^{(i,j)} = Concat\left(\mathbf{F}_{m,1}^{(i,j)}, \mathbf{F}_{m,2}^{(i,j)}, ..., \mathbf{F}_{m,heads}^{(i,j)}\right) \odot \mathbf{W}_c, \tag{4.23}$$

where *heads* denotes the number of multi-heads, and $\mathbf{W}_c$ represents the associated parameters.

4.7 Multiscanning Transformer (MT)

There are two distinct strategies to implement the entire network. The initial approach entails processing each scanned sequence independently, where the input sequences are fed into the network separately. In this approach, the network receives each sequence independently and performs computations accordingly.

The second approach, on the other hand, utilizes pairs of sequences, specifically the forward and backward scanning orders. Within the multiscanning strategy, four pairs of forward and backward scanning orders exist, including order-1 and order-3. Incorporating a Bi-RNN is advisable to capitalize on the larger receptive fields and improve the model's efficacy [35]. Using Bi-RNN, the

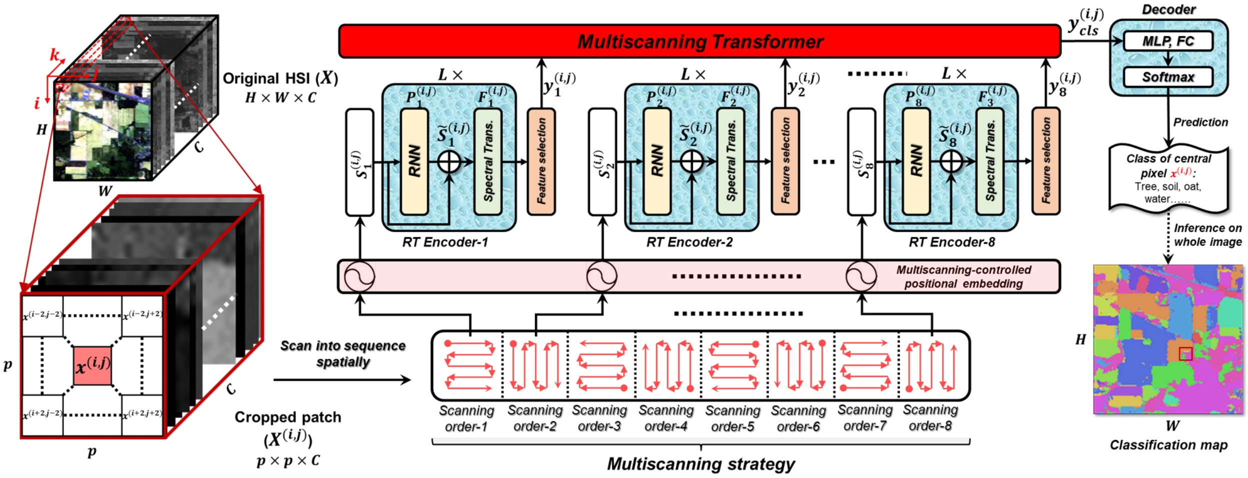


Figure 4.3 Our proposed approach entails the following phases: (1) Extracting an HSI patch with a size of $p = 5$, centring it around a focal pixel, denoted as $\mathbf{x}^{(i,j)}$ for instance. (2) Implementing a multiscanning strategy to scan the extracted patch into multiple pixel-sequences, labeled as $\mathbf{S}_m^{(i,j)}, m = 1, 2, ..., \mathcal{M}$. (3) Utilizing the RT encoder to each pixel-sequence to procure its feature representation, referred to as $\mathbf{F}_m^{(i,j)}$. (4) Deploying the feature selection layer to deduce a novel representation $\mathbf{y}_m^{(i,j)}$ from $\mathbf{F}_m^{(i,j)}$. (5) Merging all feature representations denoted as $\mathbf{y}^{(i,j)}m, m = 1, 2, ..., M$ through the use of the Multiscanning Transformer encoder. (6) Applying the decoder to determine the category of the central pixel ($\mathbf{x}^{(i,j)}$) based on the fused feature $\mathbf{y}_{cls}^{(i,j)}$. (7) Repeating these steps for each central pixel to generate the final classification map for the entire image.

network can capture information from both the forward and backward directions, enabling a more comprehensive analysis of the input data.

The overall structure of scheme-1 is depicted in Figure 4.3. It showcases the flow of information and the integration of the proposed approach within the network architecture.

4.7.1 Scheme-1: Separating Multiscanning Feature Fusion by MT

Generating a feature representation for an HSI patch begins by cropping it and applying the multiscanning strategy to scan it into multiple pixel-sequences. Each pixel-sequence is treated independently and passed through the RT encoder, resulting in a feature representation $\mathbf{y}_m^{(i,j)}$ for the m-th scanning order. These individual features are subsequently concatenated to capture scanning order-based attention using the MT module. By combining the features from multiple scanning orders, the model can effectively capture and utilize the complementary information obtained through different scanning directions, enhancing the overall representation of the HSI patch.

The feature representation $\mathbf{Y}^{(i,j)} = [\mathbf{y}_{cls}^{(i,j)}, \mathbf{y}_1^{(i,j)}, \ldots, \mathbf{y}_8^{(i,j)}]^\top \in \mathbb{R}^{(1+8)\times d}$ is created by concatenating the features generated by the RT encoder for each of the eight scanning orders, along with an additional learnable class token $\mathbf{y}_{cls}^{(i,j)}$. The resulting feature vector has a dimension of $(1+8)\times d$ and serves as the input for the MT encoder, which is designed based on the general $\mathcal{VIT}()$ encoder as described in [13]:

$$\tilde{\mathbf{Y}}^{(i,j)} = \mathcal{VIT}\Big(\mathbf{Y}^{(i,j)}\Big), \tag{4.24}$$

$$\mathbf{y}_{cls}^{(i,j)} = \sum_{m=1}^{8}\Big(\mathbf{E}^{(i,j)}[m] \odot \tilde{\mathbf{Y}}^{(i,j)}[m]\Big), \tag{4.25}$$

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

$$\mathbf{E}^{(i,j)}[m] = Softmax\Big\{\Big(\tilde{\mathbf{Y}}^{(i,j)}[m]\Big)^\top \odot \tilde{\mathbf{Y}}^{(i,j)}[0]\Big\}, \tag{4.26}$$

The resulting fused feature $\mathbf{y}_{cls}^{(i,j)}$ is subsequently passed through the subsequent decoder for classification. In Equation 4.24, each scanning order is treated independently, and positional information is not explicitly incorporated.

4.7.2 Scheme-2: Pairing Multiscanning Feature Fusion by MT

In Scheme-2, the multiscanning sequences are processed in pairs. The process is listed below:

The corresponding forward and backward pixel-sequences are generated for each scanning order pair using the multiscanning strategy.

Each forward and backward sequence pair is separately passed through the RT encoder to obtain the corresponding feature representations $\mathbf{y}_m^{(i,j)}$ and $\mathbf{y}_{m'}^{(i,j)}$, where m and m' represent the scanning orders.

The feature representations $\mathbf{y}_m^{(i,j)}$ and $\mathbf{y}_{m'}^{(i,j)}$ are concatenated to create a fused feature representation $\mathbf{Y}_{m,m'}^{(i,j)}$ for the pair of scanning orders.

The fused feature representation $\mathbf{Y}_{m,m'}^{(i,j)}$ is passed through the MT encoder, similar to Scheme-1, to obtain the final fused feature $\mathbf{y}_{cls}^{(i,j)}$.

The final fused feature $\mathbf{y}_{cls}^{(i,j)}$ is then used as input for the subsequent decoder for classification.

4.8 Experiments and Discussion

In this part, we first describe four public HSI datasets used in the experiments. Then, we will provide details about the implementation of the proposed method and the comparison methods employed for evaluation. Finally, we will present the results of extensive experiments, including an ablation analysis, to evaluate the performance of our proposed method quantitatively and qualitatively.

4.8.1 Description of Datasets

Three datasets are previously presented in Figure 2.11 with samples listed in Table 2.3. This section will focus on presenting the fourth dataset, Houston 2013 (HU), as depicted in Figure 4.4.

The 2013 IEEE Geoscience and Remote Sensing Society (GRSS) data fusion contest utilized the HU dataset. This dataset comprises 144 spectral bands with a 349×1905 pixels spatial dimension. It encompasses a total of 15 distinct land-cover classes. The dataset is valuable for evaluating and comparing different approaches in data fusion and land-cover classification.

For the IP, PU, and SA datasets, the labeled samples are divided into training and testing sets to assess the effectiveness and practicality of the proposed methods. In our experimental setup, 10% of the labeled pixels are selected as training samples, and the remained 90% labeled pixels are prepared for testing. Additionally, within the training set, 5% of the samples are further set aside

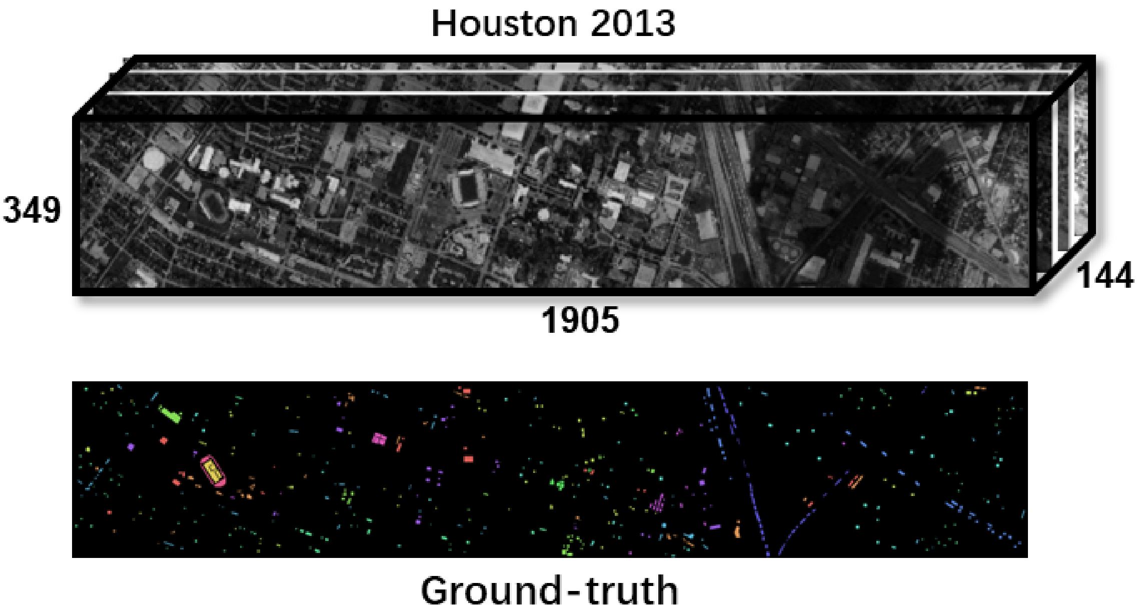


Figure 4.4 The depiction of HU dataset: original image, ground-truth.

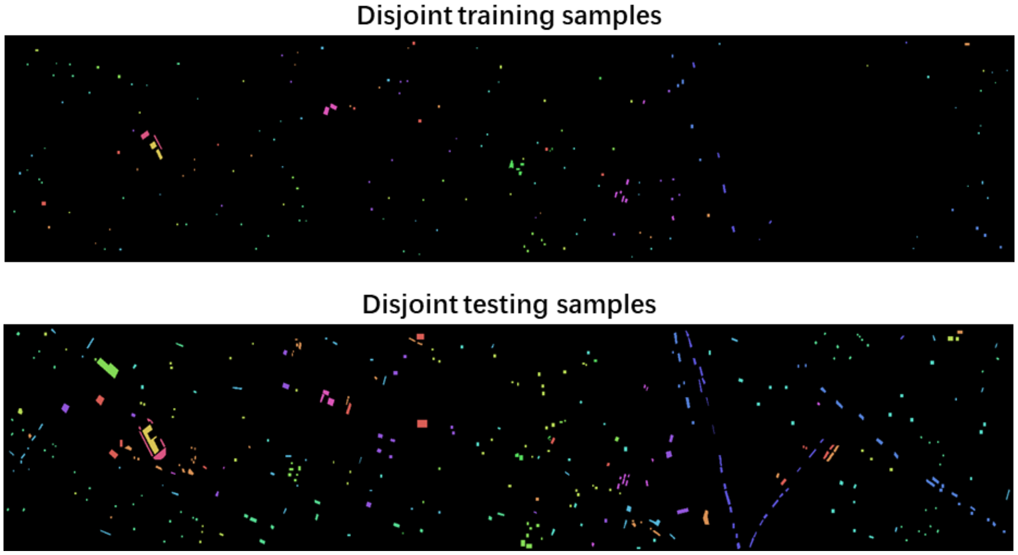


Figure 4.5 Illustration of disjoint training/testing samples for HU dataset.

as validation samples. These validation samples monitor the model's performance during training, ensuring it generalizes well and performs optimally on unseen data.

In the case of the HU dataset, we employ a disjoint sampling approach to simulate a more

Table 4.1 Types of land-cover and pixel count on the dataset of HU.

Class			Number of Samples	
Type Number	Color	Name	Training samples	Testing samples
1	🟥	Grass-health	198	1053
2	🟧	Grass-stress	190	1064
3	🟨	Grass-synthetic	192	505
4	🟩	Tree	188	1056
5	🟩	Soils	186	1056
6	🟩	Waters	182	143
7	🟩	Residential	196	1072
8	🟦	Commercials	191	1053
9	🟦	Roads	193	1059
10	🟦	High-ways	191	1036
11	🟦	Rail-ways	181	1054
12	🟪	Park-l-1	192	1041
13	🟪	Park-l-2	184	285
14	🟪	Tennis-ground	181	247
15	🟥	Running-track	187	473
Total			2832	12197

realistic scenario for practical use in the real world. This approach is depicted in Figure 4.5 with the number of samples selected listed in Tables 4.1.

4.8.2 Experimental Setting

General Setting

We employ a training epoch of 200 and a batch size of 100 for our proposed method. The Adam optimizer and the cross-entropy loss function are utilized during the training process. Our experiments replace all RNN units with LSTM units for improved performance. The LSTM layers are set to 3, with a hidden size of 64. The feature dimensions in the model, including LSTM, Transformer, and fully connected layers, are also set to 64. Additionally, we use three layers in the RT encoder. The initial patch size is 7×7, 9×9, 11×11, and 9×9 for the IP, PU, SA, and HU datasets.

Evaluation Indicators

The classification performance of the proposed method and other comparison methods is evaluated quantitatively using four key indicators: overall accuracy (OA), average accuracy (AA), Kappa coefficient ($Kappa$), and class-specific accuracy. These indicators provide a comprehensive assessment of the effectiveness of the methods in accurately classifying the HSI.

Compared Methods

The proposed method 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. Several state-of-the-art models were considered for this comparison, including General ViT [13], SpeFormer [32], 1D-CNN with Transformer (1DCT) [34], SST [30], SSFTT [67], SAT [56], 3D-ANAS [82], CasRNN [23], and Multi-LSTM [91].

All of the models mentioned above are designed as patch-wise classifiers, and the parameter settings used for each model are consistent with their respective references. It ensures a fair and standardized comparison among the models. Additionally, the comparison of results is based on the same conditions of training and testing samples to ensure a fair evaluation and eliminate any potential bias caused by imbalanced training data. By evaluating these standardized conditions, we can accurately assess the performance of each model and make reliable comparisons between them.

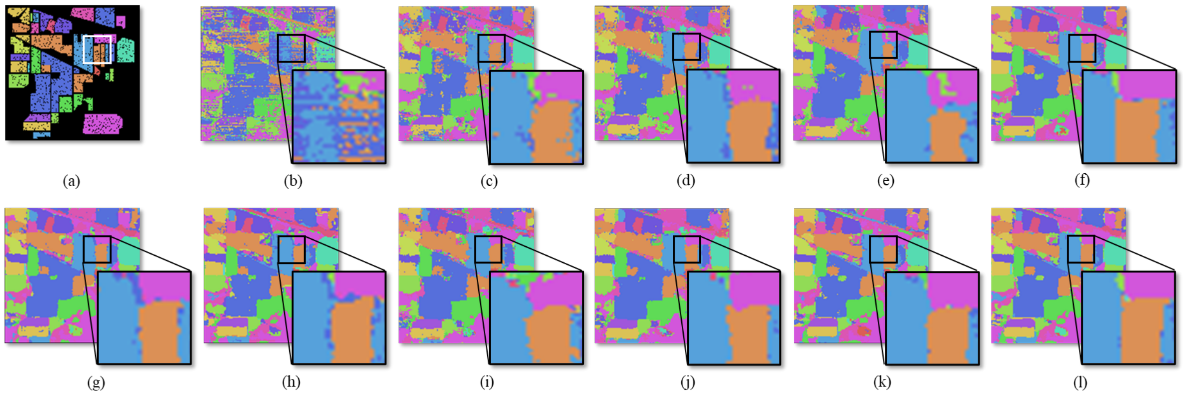


Figure 4.6 The classification maps for IP dataset: (a) Testing Area. (b) General ViT. (c) SpeFormer. (d) 1DCT. (e) SST. (f) SAT. (g) 3D-ANAS. (h) CasRNN. (i) Multi-LSTM. (j) Our First Scheme. (k) Our Second Scheme. The representative regions are magnified.

4.8.3 Quantitative Results with Classification Maps

Quantitative classification outcomes, which include OA, AA, Kappa, and class-specific accuracy for the IP, PU, SA, and HU datasets, are outlined in Tables 4.2, 4.3, 4.4, and 4.5 respectively. These tables offer an exhaustive assessment of the performance of the proposed methodology concerning other compared methods across diverse datasets.

Furthermore, the corresponding classification maps, depicting the training and testing samples, are displayed in Figure 4.6, Figure 4.7, Figure 4.8, and Figure 4.9. These maps visually illustrate the classification results and provide a qualitative understanding of the performance of the proposed method.

The results indicate that the methods relying solely on the Transformer for HSI classification, such as the General ViT model, do not achieve satisfactory performance. The overall classification maps generated by these methods show poor accuracy and significant misclassifications. However, there is a performance improvement when using the SpeFormer method compared to the General ViT model. The SpeFormer method captures more accurate land-cover classes in the classification maps. Nevertheless, some misclassifications are still present, resulting in noise-like patterns in

Table 4.2 Quantitative performance metrics, including OA, AA, and Kappa, along with the accuracies for each class, were evaluated for various classification methods for the IP dataset. Best results are highlighted.

Class	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 maps. Upon closer inspection, it is observed that certain land-cover boundaries appear irregular, and some homogeneous regions lack smoothness in the classification maps. These issues are highlighted by the boxes in the classification maps, indicating areas where the methods struggle to classify the land-cover types accurately.

In contrast to the methods that solely rely on the Transformer, some approaches combine CNNs and Transformer for HSI classification. The 1DCT method, which incorporates a 1D-CNN, primarily focuses on capturing spectral features. However, this approach yields subpar classification results, indicating that more than solely relying on spectral information may be required for accurate classification. The SST method takes a different approach by utilizing the VGG-16 architecture on the HSI data before passing it to the Transformer encoder. It allows for integrating spectral and spatial information into the classification process. The SAT approach further improves upon this

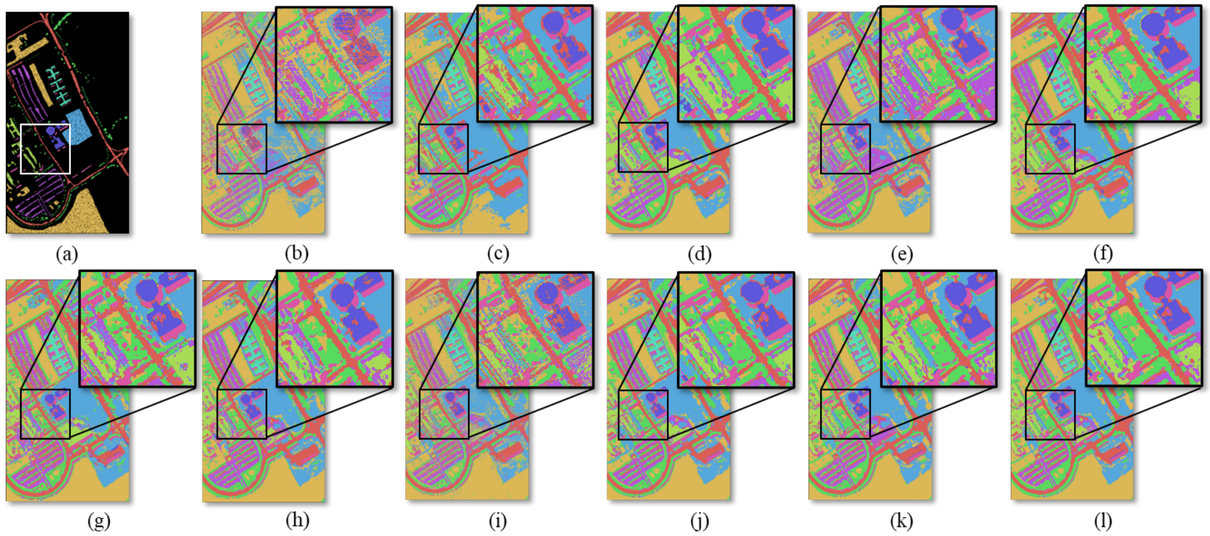


Figure 4.7 The classification maps for PU dataset: (a) Testing Area. (b) General ViT. (c) SpeFormer. (d) 1DCT. (e) SST. (f) SAT. (g) 3D-ANAS. (h) CasRNN. (i) Multi-LSTM. (j) Our First Scheme. (k) Our Second Scheme. The representative regions are magnified.

by incorporating the CBAM attention block for spectral attention prior to the spatial Transformer. These methods demonstrate improved performance compared to the Transformer-only approaches but still have limitations. More recently, the SSFTT and 3D-ANAS methods have emerged, combining 3D-CNN, 2D-CNN, and Transformer models. These hybrid models achieve comparable results, producing smoother regions and more precise boundaries in the classification maps. However, deep 3D-CNN and 2D-CNN models come with a higher computational burden and longer processing times, which may be a limitation in specific practical applications.

Among the RNN-based methodologies, CasRNN and Multi-LSTM offer an alternative approach to HSI classification. CasRNN combines CNNs and RNNs, with the RNN component specifically enhancing spectral features. It achieves this by dividing the HSI data into smaller sub-images along the spectral domain, allowing for a more focused analysis of spectral characteristics. This approach results in a refined feature representation that eliminates redundancies and incorporates non-adjacent information, ultimately improving the accuracy of the classification. However, the spatial distribution of the classification results obtained from CasRNN tends to be irregular.

Table 4.3 Quantitative performance metrics, including OA, AA, and Kappa, along with the accuracies for each class, were evaluated for various classification methods for the PU dataset. Best results are highlighted.

Class	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**

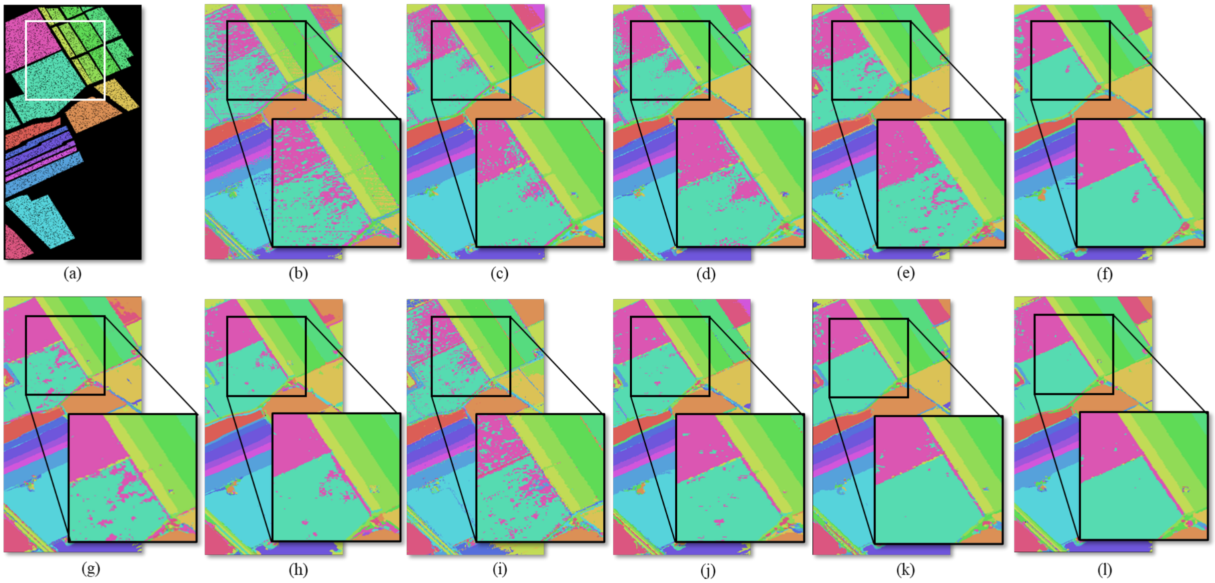


Figure 4.8 The classification maps for SA dataset: (a) Testing Area. (b) General ViT. (c) SpeFormer. (d) 1DCT. (e) SST. (f) SAT. (g) 3D-ANAS. (h) CasRNN. (i) Multi-LSTM. (j) Our First Scheme. (k) Our Second Scheme. The representative regions are magnified.

This irregularity is mainly due to the absence of shuffling in the spatial domain, a characteristic of CNN-based methods. The reliance on RNNs in CasRNN limits its ability to capture spatial dependencies effectively, resulting in less smooth and more irregular classification maps.

Table 4.4 Quantitative performance metrics, including OA, AA, and Kappa, along with the accuracies for each class, were evaluated for various classification methods for the SA dataset. Best results are highlighted.

Class	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**

In contrast, Multi-LSTM takes a different approach by adopting a multiscanning strategy to scan HSI patches into multiple complementary pixel-sequences. This approach recognizes the significance of diverse scanning orders in enhancing the discriminative power of RNNs. By incorporating multiple pixel-sequences into the classification process, Multi-LSTM surpasses the performance of CasRNN. The resulting classification maps demonstrate more precise land-cover boundaries and reduced noise-like misclassifications. It highlights the multiscanning approach's effectiveness in improving the classification results' accuracy and robustness.

This study introduces a novel approach for HSI classification called the multiscanning-based RNN-Transformer model. This model combines the advantages of both RNNs and Transformers to improve the accuracy and robustness of classification results. Our approach's key innovation is using a multiscanning strategy, which converts the HSI patch into multiple pixel-sequences.

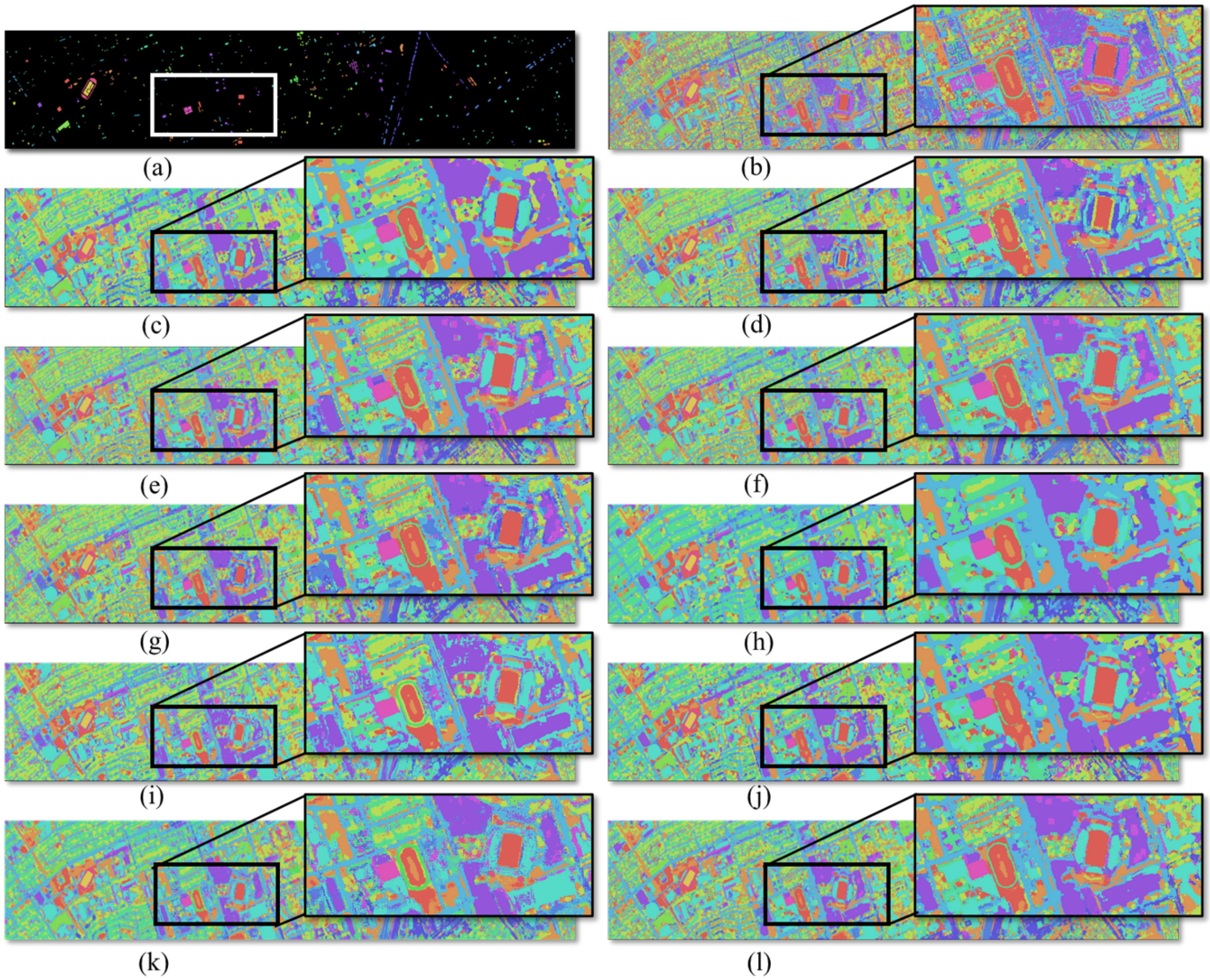


Figure 4.9 The classification maps for HU dataset: (a) Testing Area. (b) General ViT. (c) SpeFormer. (d) 1DCT. (e) SST. (f) SAT. (g) 3D-ANAS. (h) CasRNN. (i) Multi-LSTM. (j) Our First Scheme. (k) Our Second Scheme. The representative regions are magnified.

These sequences are then processed by an RT encoder, incorporating a spectral-spatial-based soft mask for feature generation and selection. The main objective of this approach is to overcome the challenges posed by interfering pixels and improve the accuracy of land-cover classifications, particularly at class boundaries.

Our evaluation indicates that our method achieves superior performance, particularly in scheme 2, which combines Bi-RNN with Transformer. The classification maps generated by our method

Table 4.5 Quantitative performance metrics, including OA, AA, and Kappa, along with the accuracies for each class, were evaluated for various classification methods for the HU dataset. Best results are highlighted.

Class	Transformer only		Transformer + CNNs					RNNs		Transformer + RNNs	
	general ViT	SpeFormer	1DCT	SST	SSFTT	SAT	3D-ANAS	CasRNN	Multi-LSTM	scheme 1	scheme 2
1	86.69	90.11	88.20	90.01	90.60	85.30	90.32	89.61	90.51	**95.17**	90.41
2	93.87	77.82	94.41	93.31	91.69	94.01	94.61	92.50	95.68	**96.52**	93.60
3	99.80	99.88	**100**	**100**	99.90	**100**	99.88	97.72	98.62	97.77	**100**
4	93.62	90.73	94.82	94.40	95.58	91.78	95.82	96.19	89.20	95.60	**96.21**
5	95.81	99.51	98.35	99.49	99.33	98.35	**99.59**	97.73	99.48	98.01	99.32
6	87.18	97.60	96.62	98.32	94.48	97.31	92.30	97.59	**100**	95.92	97.90
7	80.28	85.33	93.80	90.52	87.46	**92.88**	87.31	87.81	77.43	91.08	91.22
8	55.06	67.69	84.18	75.31	80.22	88.49	78.99	73.71	81.52	79.64	**89.18**
9	75.56	91.80	87.62	**93.79**	91.66	89.61	87.58	78.19	91.14	87.48	92.21
10	61.61	77.77	79.16	71.22	87.58	81.44	83.76	64.60	77.56	**92.56**	87.32
11	65.22	84.31	86.24	79.18	88.50	86.32	84.30	85.28	87.62	89.90	**90.37**
12	69.35	94.02	76.79	93.83	89.88	85.35	86.52	76.79	93.78	87.91	**94.09**
13	41.14	79.72	87.88	82.40	89.34	89.70	91.31	85.33	60.10	**91.58**	90.71
14	88.36	96.09	**100**	98.22	99.20	**100**	**100**	95.89	**100**	98.58	**100**
15	99.15	99.78	96.55	97.80	99.58	97.56	99.75	97.75	98.66	92.13	**100**
OA	79.257	87.235	89.497	89.145	91.104	90.358	89.998	84.299	88.538	91.834	**92.916**
AA	79.713	88.811	90.975	90.520	92.333	91.873	90.803	88.461	89.420	92.659	**94.162**
Kappa	0.7752	0.8615	0.8860	0.8822	0.9035	0.8954	0.8914	0.8295	0.8758	0.9114	**0.9231**

exhibit smoother regions and more explicit boundaries between different land-cover classes. These results demonstrate the effectiveness of our approach in addressing the challenges associated with HSI classification.

Our proposed method offers several advantages over SSFTT and 3D-ANAS. By incorporating multiple scanning orders and soft masks, we introduce diversity and enhance the model's ability to capture different perspectives within an HSI patch. It is a feature augmentation form that improves the model's discriminative ability and generalization. As a result, our approach outperforms SSFTT and 3D-ANAS, yielding superior classification results. Incorporating multiple scanning orders and soft masks allows our method to effectively handle complex spatial and spectral relationships in HSI data.

In addition to the advantages mentioned earlier, our proposed method also builds upon the previous work of Multi-LSTM by seamlessly integrating RNN and Transformer models. We have

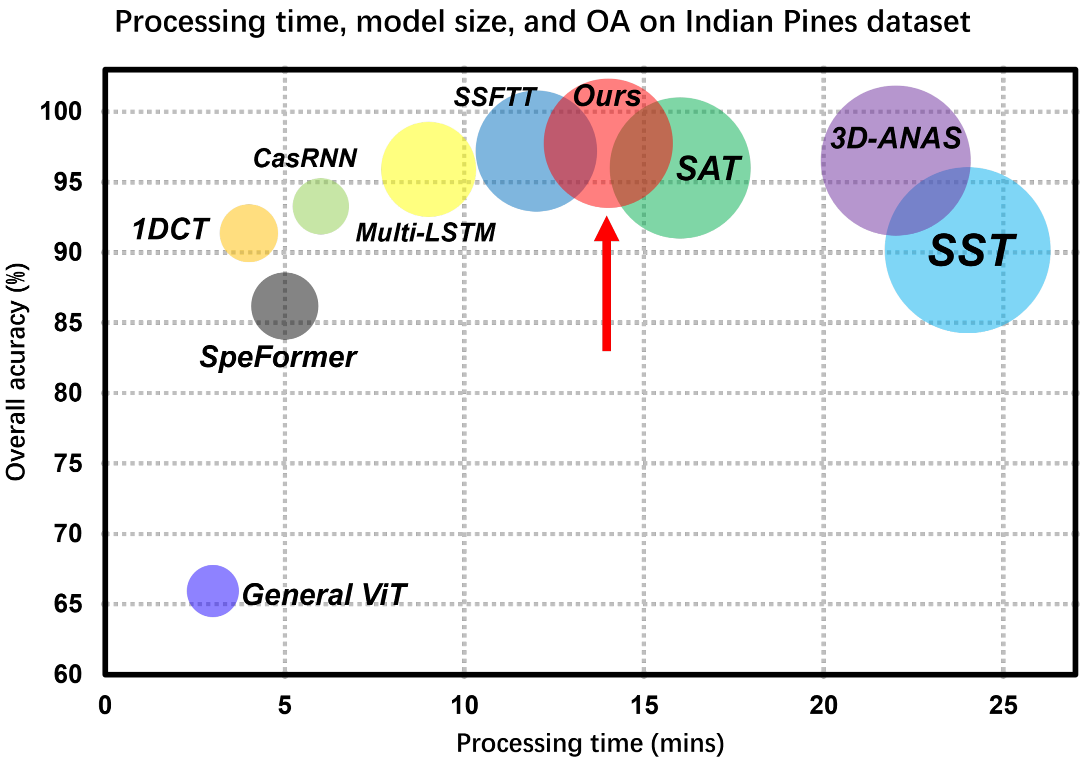


Figure 4.10 The comparison of processing time, model size, and OA on IP dataset.

addressed the limitations of Multi-LSTM, such as its performance with larger initial patch sizes and the feasibility of incorporating attention mechanisms. By incorporating these improvements, our method achieves enhanced classification performance compared to Multi-LSTM.

Moreover, our proposed scheme-2, which involves a bi-directional RNN-Transformer, offers a more compact and practical solution than scheme-1. Using Bi-RNN allows for the simultaneous processing of two inverse sequences, effectively expanding the receptive field and capturing more discriminative and informative features.

Additionally, the comparison of processing time, model size (number of parameters), and OA on the IP dataset is shown in Figure 4.10.

4.8.4 Other Analysis 1) to 6)

1) Balance Weight in Positional Embedding, γ and δ

In many studies, positional encoding is commonly achieved by directly adding positional information to the original input, assuming that $\gamma = \delta$, as demonstrated in Equation 4.3. However, summing these values without careful consideration may not be appropriate. Therefore, we investigated the optimal values of γ and δ using the multiscanning strategy to assess their impact on model performance, as shown in Table 4.6.

The experimental results demonstrate that the weight for positional features (δ_m) is generally higher than that for spectral features (γ_m) in the m-th scanning order. It implies that positional features derived from the RNN outputs play a dominant role in the input features for the Spectral Transformer. The RNN effectively encodes more discriminative features than the original inputs, enhancing the importance of positional information in the feature representation. This finding highlights the effectiveness of the RNN in capturing spatial dependencies and generating more informative positional features, which contribute significantly to the overall performance of the Spectral Transformer.

Our approach distinguishes itself from others by introducing dynamic weights γ and δ to enhance the informativeness of inputs for the Spectral Transformer. By assigning higher weights to positional features derived from the RNN outputs, we leverage the RNN's ability to capture spatial dependencies and generate more informative representations. This novel approach enhances the role of positional embedding in the feature representation process. It could contribute to advancing research on positional embedding techniques in various applications.

2) Attention Map Visualization

In the Spectral Transformer, the attention sequence $\mathbf{A}^{(i,j)} \in \mathbb{R}^{p^2 \times 1}$ is computed using the formula described in Eq 4.16. This sequence represents the dot-product values between each pixel and

Table 4.6 The impacts of parameters γ and δ in positional embedding with multiscanning strategy were examined.

	Datasets							
	IP		PU		SA		HU	
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

the central pixel. These dot-product values can be transformed into an attention map $\mathbf{A'}^{(i,j)}$ for visualization purposes, with a size of $\mathbb{R}^{p\times p}$. We provide representative attention maps in Figure 4.11.

The attention maps visually demonstrate the effectiveness and superiority of the designed Spectral Transformer module. They exhibit fine details like edges, object outlines, and textural structures. This visual evidence supports the claim that the Spectral Transformer module can capture important spectral-spatial features from the attended regions.

Moreover, attention maps emphasize the capability of the Spectral Transformer to differentiate between homogeneous and interfering pixels in HSI cubes. This ability enhances the discriminative power of the extracted features, resulting in improvements.

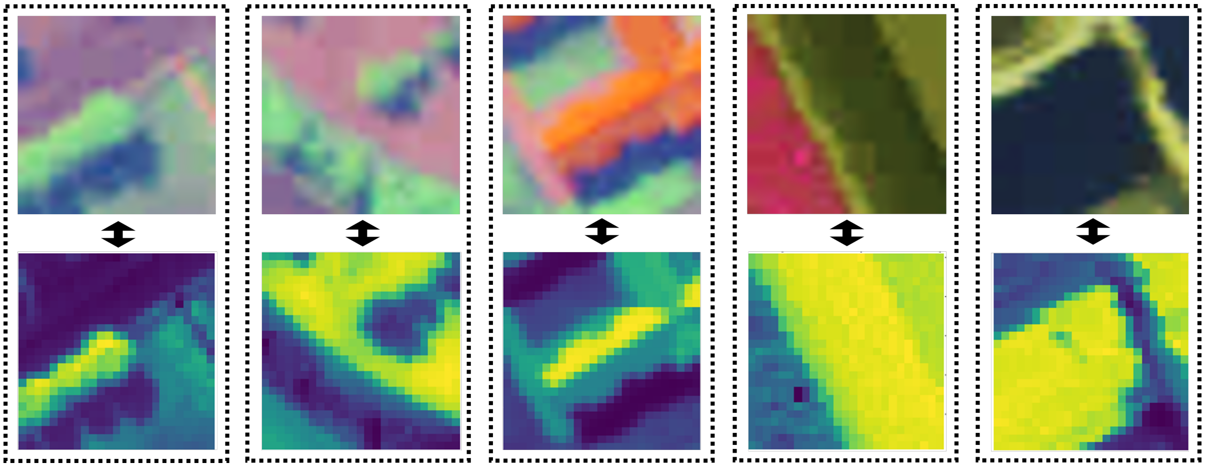


Figure 4.11 The examples of attention maps. The upper image displays the land-cover distribution of a cropped HSI patch, while the lower image showcases the obtained attention maps denoted as $\mathbf{A'}^{(i,j)}$.

3) Effects of Initial Patch Size for Performance

In this part, we look at how the starting patch size affects how well the classification works. We test an extensive range of patch sizes, from 3 to 23, and show the overall accuracy results in Table 4.7.

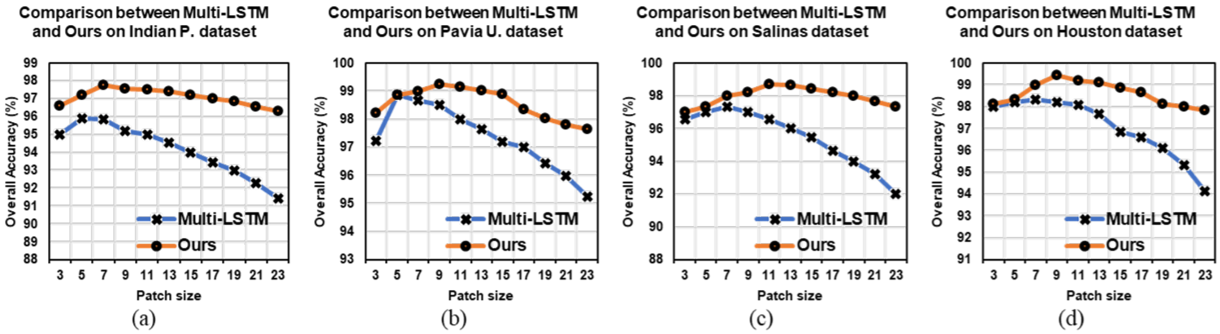
If the patch size is in the proper range, it can make the model work better because it uses more space information. However, if the patch size is too big, it may bring in extra interfering pixels that mess up the results. The IP dataset, which is small and has closely packed land-cover types, needs to work better with big patch sizes. However, the SA dataset, which is bigger and has evenly spread out ground objects, can handle bigger patch sizes. The PU and HU datasets, both big and with complex land-cover types, work best with patch sizes of 9×9. Based on our tests, the best results for the SA and IP datasets come from patch sizes of 11×11 and 7×7, respectively.

We also compare our model to a pure RNN-based approach, shown by Multi-LSTM, across different patch sizes. Figure 4.12 shows this comparison. Multi-LSTM's accuracy drops faster than its best patch size, while our model keeps its accuracy pretty well, only going down a little.

Also, the steady results with bigger patch sizes can be credited to using soft attention masks in

Table 4.7 The impact of different patch sizes on the overall accuracy (%) was evaluated across four datasets. The best performing patch size is highlighted.

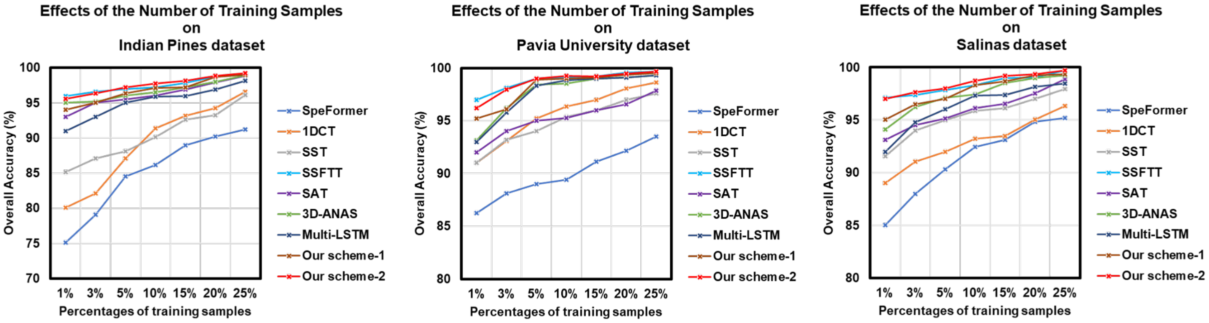
	Datasets			
Patch size	IP	PU	SA	HU
3×3	96.598	98.216	96.991	98.123
5×5	97.225	98.881	97.332	98.335
7×7	**97.751**	98.001	98.010	99.013
9×9	97.558	**99.253**	98.219	**99.441**
11×11	97.501	99.159	**98.723**	99.213
13×13	97.418	99.021	98.661	99.111
15×15	97.222	98.885	98.438	98.863
17×17	97.011	98.369	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



(a) (b) (c) (d)

Figure 4.12 A performance comparison was conducted between the Multi-LSTM method and our proposed method for various patch sizes. The evaluation was performed on the following datasets: (a) IP dataset. (b) PU dataset. (c) SA dataset. (d) HU dataset.

our method. These masks help handle the harmful effects of extra pixels by giving them weights, which helps keep the performance steady.



(a) (b) (c)

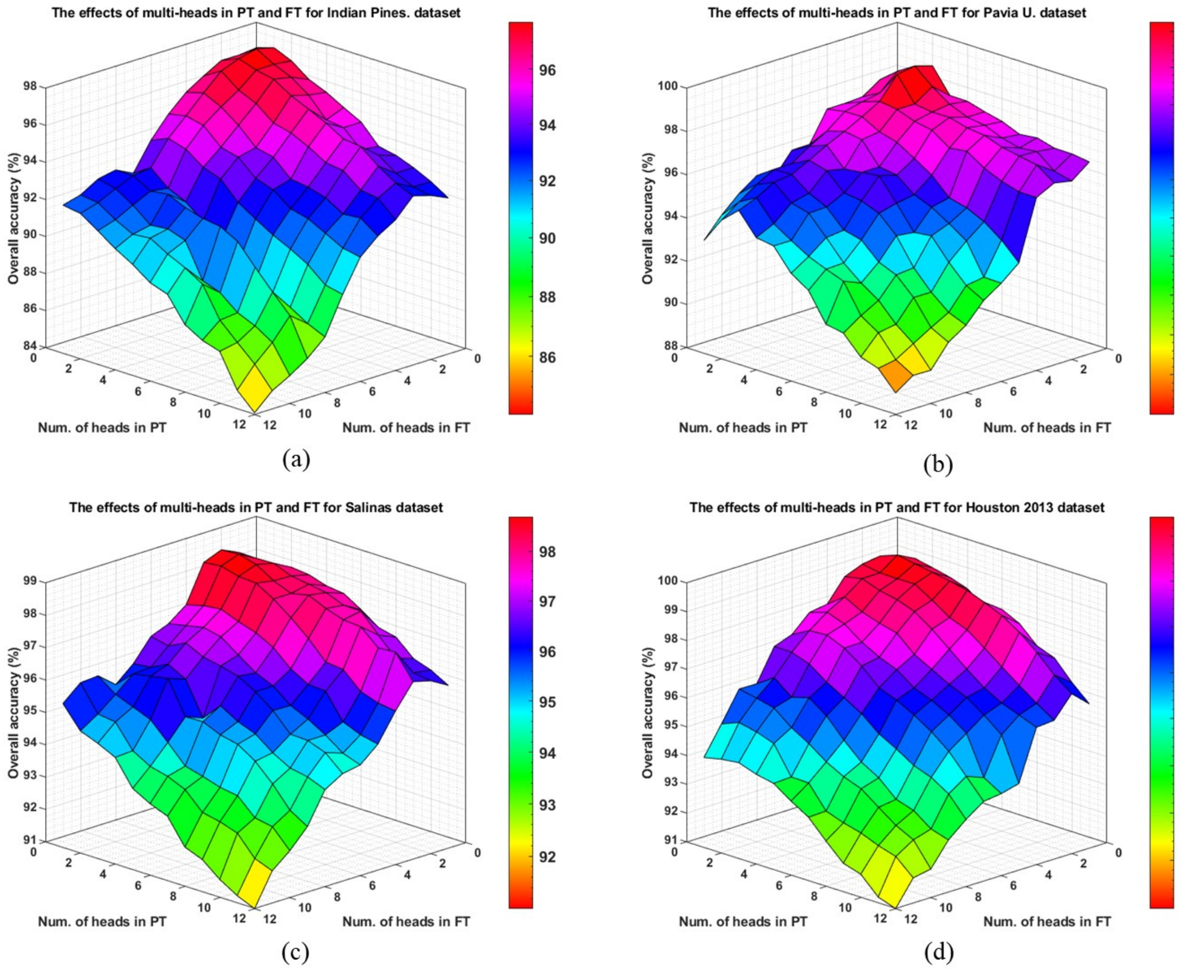
Figure 4.13 The overall accuracies of different models was assessed by using different percentages of training samples on the following datasets: (a) IP dataset. (b) PU dataset. (c) SA dataset.

4) Effects of the Number of Training Samples

In this test, we look at how changing the number of training samples affects how well each method works. The only thing we changed in this test is the number of chosen training samples. We keep everything else the same. We randomly pick the training sets from each group, with sizes from 1% to 25%.

Because the HU dataset has set samples, we show the overall accuracy (OA) results for different training set sizes for IP, PU, and SA datasets in Figure 4.13. The bottom line of the graph shows the percentage of chosen training samples per group, and the sideline shows the OA.

Looking at Figure 4.13, we can see that the OA for each method goes up when we increase the number of training samples. It tops out at almost 100% for some methods. Our method works well even with a small number of samples. However, it is essential to remember that when the sample size is 10%, other methods do not work as well. The SSFTT method, though, works as well or better than our method. Plus, we can see that if we increase the training sample size beyond 5%, it does not change how well the methods work.



(a) (b) (c) (d)

Figure 4.14 The overall accuracy influenced by the number of heads in multi-head self-attention within the Transformer settings (ST and MT) was examined on the following datasets: (a) IP dataset. (b) PU dataset. (c) SA dataset. (d) HU dataset. The x-axis represents the number of heads in ST, while the y-axis represents the number of heads in MT. The z-axis represents the corresponding overall accuracy.

5) Analysis on Multi-Heads in ST and MT

We conducted supplementary trials to scrutinize the influence of the headcount on our multi-head self-attention efficacy. The headcount was altered as a vital hyperparameter in the ST and MT scenarios. Figure 4.14 depicts the findings of these trials, focusing on overall precision. It is essential to acknowledge that other adjustable hyperparameters exist within the Transformer setup, such as the feature size and layer depth. Nevertheless, we maintained a standard setup: a single layer depth, all dimensions set to 64, and a dropout rate of 0.5 for consistency.

As revealed in Figure 4.14, there is a slight increase in overall accuracy corresponding with the growth in the number of heads. The ideal number of heads varies across datasets; the most efficacious results were observed for IP at ST heads = 2 and MT heads = 2, PU at ST heads = 2 and MT heads = 3, SA at ST heads = 3 and MT heads = 2, and HU at ST heads = 2 and MT heads = 2. Notably, the model exhibits increased sensitivity towards the headcount in the ST. As a result, choosing the optimal number of heads for the multi-head self-attention mechanism becomes critical for securing the best possible results.

6) Analysis on Layer Depth in MT

To explore how the depth of the layers affects how well the Transformer-based model works, we ran more tests with everything else staying the same (all dimensions set to 64, dropout set to 0.5, and the number of multi-heads decided based on what we found in earlier tests).

We conducted experiments to investigate layer depth's impact on our model's MT component. The results of these experiments can be found in Table 4.8. Our results suggest that the ideal MT depths for the four respective datasets are 3, 2, 2, and 3.

The results show that setting the MT depth to 1 leads to poor model fit across all datasets. Conversely, the model may overfit the data when the layer depth is excessively high, resulting in less-than-ideal performance. Although deep layers are frequently employed in model training for many tasks, selecting suitable hyperparameters for the Transformer model, such as the number of layers and multi-heads, is imperative to ensure optimal performance.

Table 4.8 The effects of layer depth were investigated in the proposed MT. Best results are highlighted.

	Datasets			
Layer Depth	IP	PU	SA	HU
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

4.8.5 Ablation Study

Exploring the Capacity with Multiscanning

Given the better results we have seen using a multiscanning strategy, it makes sense to ask two key questions: 1) how much can the multiscanning process do? and 2) which scan is the most important? To answer these questions, we run tests focusing on three main areas:

- The influence of individual scanning on the overall accuracy.
- The saturation effect that arises from using multiple scanning procedures.
- The comparative significance of each scanning order while understanding an HSI patch.

In addressing the first point, we conducted tests employing a single scan for training. The results, shown in Figure 4.15, suggest that each single scan leads to a similar total accuracy. It is probably because just one scan does not give much room for generation. Also, the less-than-great validation accuracy we saw in these tests shows that the model does not fit well enough when we use just one scan.

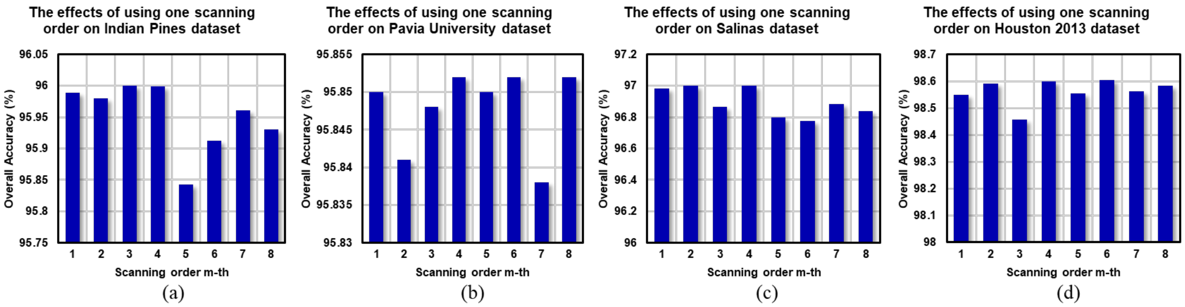


Figure 4.15 The effects of using a single scanning order for classification performance were examined on the following datasets: (a) IP dataset. (b) PU dataset. (c) SA dataset. (d) HU dataset.

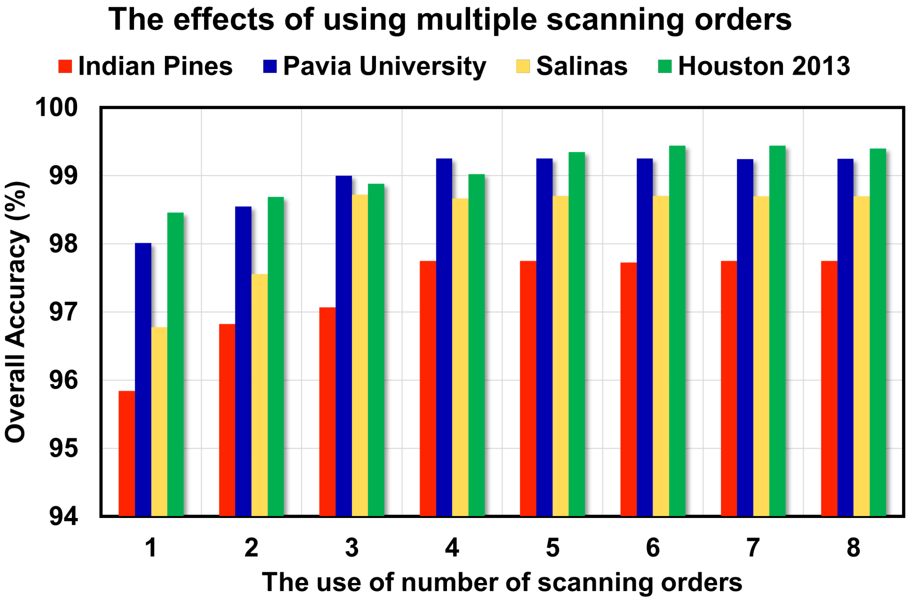


Figure 4.16 The effects of using multiple scanning orders on performance were investigated.

For the second point, we ran tests using different numbers of scans from 1 to 8. It is essential to say that when the number of scans is even (like 2, 4, 6, and 8), we use the Bi-RNN architecture described in Sec. 4.7. The results of the tests are shown in Figure 4.16. The classification accuracy improves as we increase the number of scan orders, which shows that the multiscan strategy works well. However, there is a point where adding more scans does not make the performance any better,

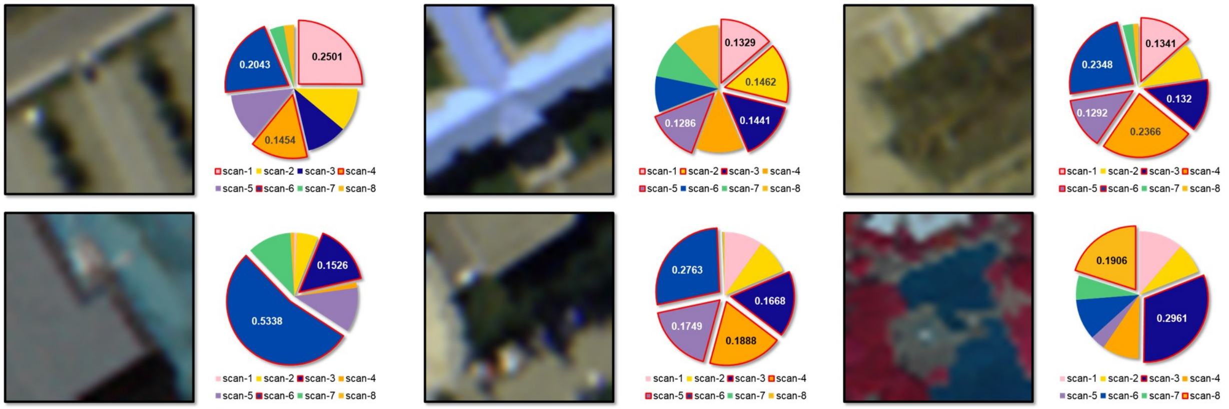


Figure 4.17 An illustration showing the distribution of scanning-based weights for a single HSI patch, highlighting higher weights (≥ 0.125).

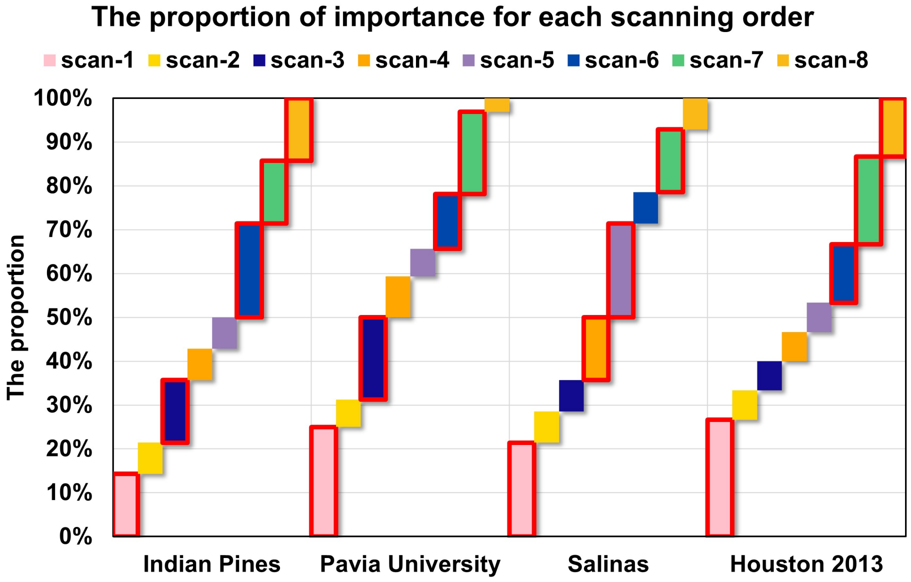


Figure 4.18 The proportion of each scanning's importance was determined by counting the frequency of scanning-based weights greater than 0.125 across four datasets.

which tells us that using all the scans is too much. The best number of scans for the IP and PU datasets turned out to be 4, while for the SA dataset, it was 3. The performance stopped improving at six scans for the HU dataset, which is complex and has an uneven spatial distribution and few samples.

For the third point, to assess the importance of each scanning order in analyzing an HSI patch, we conducted tests and analyzed the scan-based weights assigned to each scanning order. These weights indicate the attention given to each scan order by the model for a specific HSI patch. Figure 4.17 provides examples of the scan-based weights assigned to different scanning orders. These weights demonstrate the spatial distribution and highlight the varying importance of different scan orders for capturing spatial information. We conducted additional tests across multiple datasets to further investigate the importance of each scan order. We counted the number of times each scan-based weight was more significant than the average weight, as shown in Figure 4.18. This analysis allowed us to identify the scan orders that had a positive impact on the model training and those that had a negative impact.

Interestingly, our results indicate that the importance of each scan order varies across datasets and is strongly influenced by the dataset's spatial patterns. The starting patch size also plays a role in determining the importance of scan orders. However, scan-1 and scan-7 consistently emerge across all datasets as more critical scan orders than others. These findings highlight the adaptive nature of our model in assigning different weights to each scan order based on their effectiveness in capturing spatial information. By considering the specific characteristics of the dataset and the spatial patterns, our model intelligently adjusts the attention given to each scan order, ultimately enhancing the model's performance in HSI classification.

Comparison Between RNN and 1D-CNN for Integration with Transformer

To compare the performance of RNN-Transformer and 1D-CNN-Transformer models in handling pixel-sequences, we conducted tests with carefully selected parameters to ensure a fair comparison. The 1D-CNN model deployed a single feature extraction layer with 64 kernels. Meanwhile, the kernel size is 3, and the stride is 1.

The results, as shown in Figure 4.19, demonstrate that the RNN-Transformer outperforms the 1D-CNN-Transformer during validation accuracy across all datasets. It indicates that RNNs,

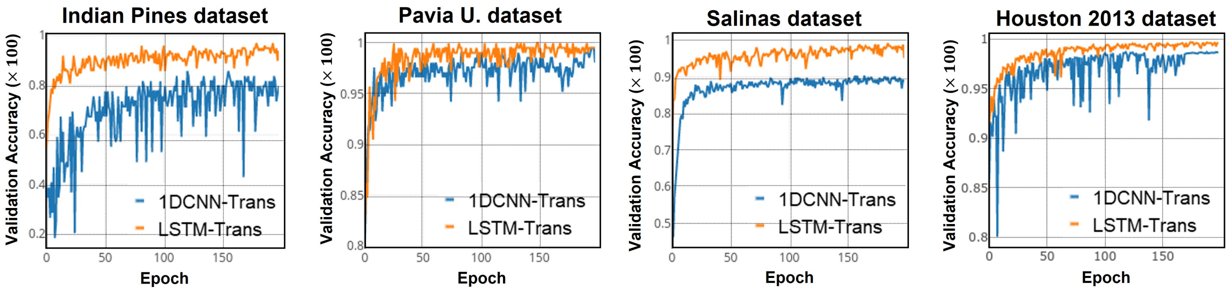


Figure 4.19 The validation accuracies of both RNN-Transformer and 1D-CNN-Transformer were compared across four datasets.

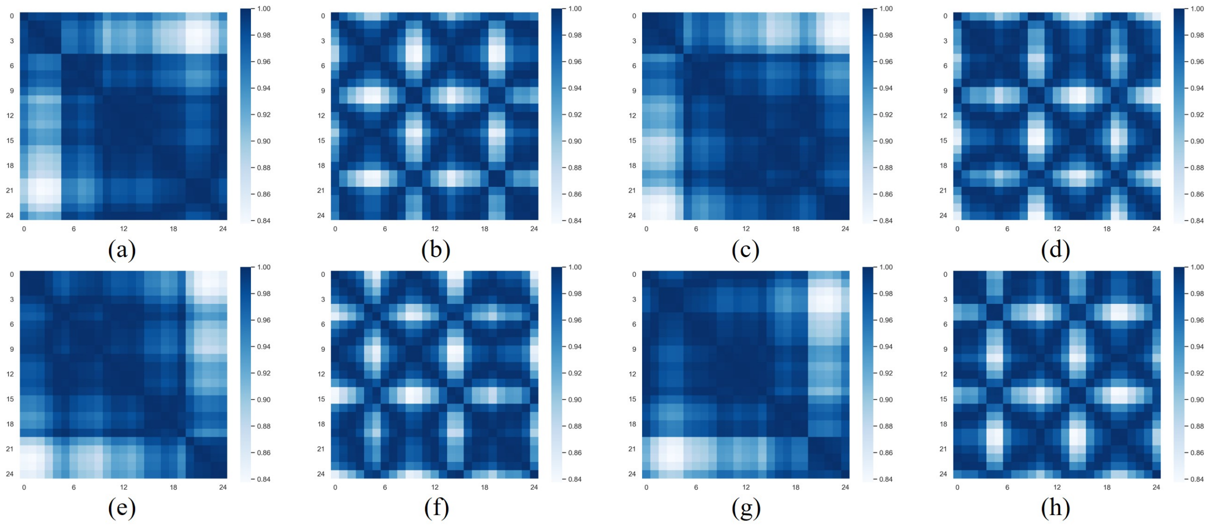
specifically LSTMs, are more effective in handling sequential data and capturing order dependencies within the pixel-sequences. The RNN-Transformer model also exhibits greater stability in its performance.

These findings suggest that the inherent nature of RNNs in modeling sequential data and capturing order bias makes them a more suitable choice for processing pixel-sequences in the context of HSI classification. The RNN-Transformer approach offers better performance and stability than the 1D-CNN-Transformer, highlighting the importance of leveraging the strengths of RNNs in modeling sequential information.

Effects of SMSA

Using the proposed soft mask that considers both spectral and spatial features in the RT encoder, our network can better decide which parts are important and which are not instead of treating everything equally. It leads to better outputs and more accurate classifications, as shown in Table 4.9. The improvements are especially noticeable for the IP and HU datasets. These two datasets have more pixels that interfere when we use a larger patch size.

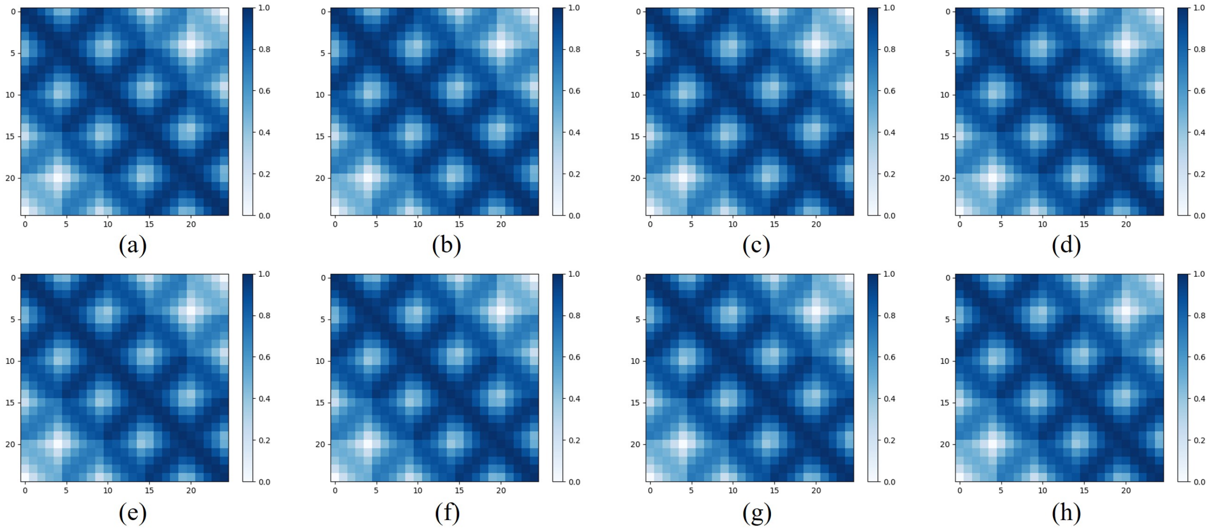
We also notice that the soft mask with spectral features works better than the one with spatial features. It happens because the spectral-based soft mask is more precise in figuring out how similar pixels are. So, it acts as the main component, while the spatial-based mask is more like a



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.20 An example of a spectral-based soft mask ($\mathbf{M}_m^{spe}$) with a size of 25×25 (patch size = 5), where the lighter parts represent the lower soft weight.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.21 An example of a spatial-based soft mask ($\mathbf{M}_m^{spa}$) with a size of 25×25. It is observed that the patterns of the spatial soft mask remain the same for different scanning orders. The lighter parts in the mask represent lower soft weight.

helper.

On top of that, we found that the pattern of the mask depends on the positional embedding.

Table 4.9 The effects of using the proposed soft mask on the overall accuracy were examined. Best results are highlighted.

	Datasets			
	IP	PU	SA	HU
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**

This results in distinct soft spectral-based masks and identical soft spatial-based masks when we employ the multiscanning strategy, as shown in Figure 4.20 and Figure 4.21. It lets the network give the proper attention weights to each pixel. This idea is supported by the better classifications on land-cover boundaries, where interfering pixels get into an HSI patch across different regions.

Comparison on Positional Embedding Methods

The results of the experiments comparing different positional embedding methods in our model are shown in Table 4.10. The RNN-based positional embedding method outperforms the other methods regarding classification accuracy.

The absolute positional embedding method, which combines an index number with the sequence length, often leads to significant differences between sequence values. It can make it challenging to manage long sequence lengths and result in unstable outcomes.

The learned positional embedding method, commonly used in Transformer models, is noisy, unstructured, independent, and lacks continuity. It assigns a position to each value independently, without considering the sequential steps, making it challenging to accurately capture the positional information among the values.

The sinusoidal/cosine positional embedding method is manually designed with fixed numbers and captures the relationship among steps to some extent. However, it may need to provide more

Table 4.10 The effects of different positional encoding strategies were investigated. Best results are highlighted.

	Datasets			
	IP	PU	SA	HU
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**

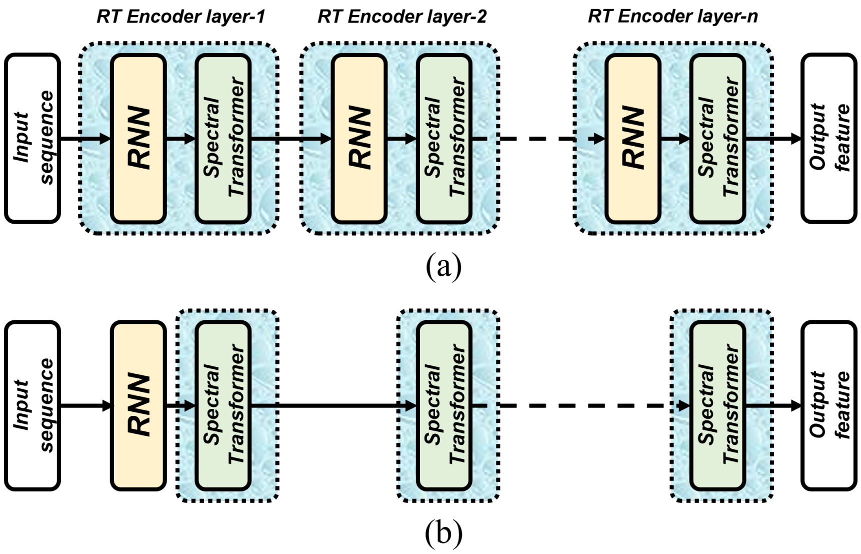
flexibility and adaptability for different situations.

In contrast, our proposed RNN-based positioning algorithm combines positional embedding with feature embedding, using dynamic and learnable parameters. It allows the model to adapt to different situations and capture the evolving positional information within the sequence. The RNN-based approach outperforms other methods by effectively modeling the order and dependencies in the sequence, leading to improved classification performance.

The Design of RT Encoder

Earlier works usually apply positional embedding before multiple Transformer layers. However, our proposed method combines RNNs with the Transformers, where the RNN is used for positional embedding. To find the best combination design, we tried two strategies as shown in Figure 4.22. We also ran additional experiments to see how these two designs affect performance. The results are given in Table 4.11.

The results from our experiments show that Design-1, which merges the RNN and Transformer into a single RT encoder, performs better than Design-2 in terms of classification performance. We observed that the OA also improves as the number of layers initially increases. The IP and HU datasets yield the best results when three layers of RT encoder are employed, while the optimal number of layers for the PU and SA datasets is two. However, it is worth noting that the classifi-



(a)

(b)

Figure 4.22 Two architectural designs for integrating RNN with Transformer are a) Design-1: This design amalgamates RNN and Transformer into a singular RT encoder. b) Design-2: This design introduces RNN once as the positional embedding before several Transformer layers.

Table 4.11 The effects of different designs for the RT encoder. Best results are highlighted.

		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

cation performance may worsen if the number of layers increases too much. Therefore, choosing the correct number of RT layers is vital for creating distinguishing features.

4.9 Conclusions with Further discussion

4.9.1 Conclusions

This study introduces a new way of classifying HSI using the multiscanning strategy, RNNs, and Transformers. The suggested method takes advantage of RNNs and Transformers' strengths while reducing shortcomings. RNNs give an ordering bias to the Transformer, which helps capture the continuous relationship between the input data. Furthermore, using multiple scanning orders for positional embedding allows us better to understand the connections between pixels with different orders. It acts like an augmentation technique for sequence models, helping to train a more robust model.

We also introduced a spectral-spatial-based soft self-attention mask, which automatically enhances features based on the data. Finally, we performed the classification using the features enhanced by the multiple scanning processes in our overall system. Compared to other methods, our approach provides competitive results with fewer parameters without using any CNNs units on four public datasets. In particular, it improves the classification of land-cover boundaries. For future work, we plan to explore how to use RNNs and Transformers for HSI classification tasks.

The detailed contributions are summarized below:

1) In this chapter, our goal is to build upon the research presented in Chapter 3 by introducing approaches such as the 'Spectral Transformer' and 'Multiscanning Transformer.' The experimental results validate their effectiveness. When compared to the findings in Chapter 3, we observe significant improvements, ranging from 2% to 4%..

2) Diverging from the conventional datasets of IP, PU, and SA, we introduce a new urban scene dataset from the GRSS community, the HU dataset. Across these four datasets, we achieve overall accuracy improvements of 6% to 11% compared to baseline methods, specifically Transformer-based approaches. Against state-of-the-art methods, our improvements range

from 2% to 5%. These enhancements are especially pronounced in the urban scene datasets, PU and HU, which feature complex geometric designs such as buildings, roads, and commercial areas. Additionally, we note significant improvements of 1% to 6% in agricultural scenes, specifically in the IP and SA datasets, surpassing the results obtained in Chapter 3.

3) Compared to other methods, our approach can save approximately 50% in processing time and reduce the model size by 40%, demonstrating the significant efficiency of our method.

4) In this chapter, we introduce several innovative proposals, including the 'Spectral Transformer' and 'Multiscanning Transformer,' among others, all of which incorporate a multiscanning strategy. We also explore and verify their potential to enhance HSI classification results.

5) In this study, we reassess the effectiveness of employing the RNNs and Transformers (i.e., pure sequential models) for HSI classification tasks. While their integration has not been deeply explored in this community, we aim to contribute to advancing related studies with our research. By investigating and demonstrating the viability of this approach, we hope to inspire future research and foster further exploration of the combined use of RNN and Transformer models for HSI classification applications.

4.9.2 Further Discussion

Even though our proposed method significantly improves upon prior work and further develops the use of sequential neural networks in HSI classification, there are still a few issues that require further discussion:

1) We applied the multiscanning strategy in the spatial domain to understand spatial dependencies. However, continuity exists clearly in the spectral domain as well. Developing a scanning method in the spectral domain might yield better results.

2) Current spectral-spatial-based HSI classifiers handle spectral and spatial features either successively (in a cascaded manner) or parallelly (in a parallel manner). While these methods incorporate both features, they often ignore the inherent relationship between these two domains. Figuring out how to unify them simultaneously is a challenge.

3) The dense design of Transformer models can present difficulties when addressing global dependencies using their self-attention mechanisms, especially with long input sequences. Introducing sparsity into the Transformer model could refine the generation of features and reduce the computational burden.

Chapter 5

General Conclusions and Future Works

5.1 Conclusions

This dissertation studies sequential neural models, including Recurrent Neural Networks (RNNs) and Transformers, with a novel multiscanning strategy for Hyperspectral Image Classification (HSI classification) tasks. These tasks involve the limitations inherent in Convolutional Neural Networks (CNNs) and RNNs, such as their inability to generate multi-directional features. This limitation can hinder the generative ability during model training and result in poorer classification results on unseen testing samples.

Chapter 2 presents an in-depth exploration of our proposed multiscanning strategy. It includes a detailed discussion of the necessary observations, the investigation of scanning selection, and the combination of scans. A broad range of potential scanning manners from the research community is examined to the greatest extent possible.

Chapter 3 outlines our further research on integrating a multiscanning strategy with RNNs as well as attention mechanism for HSI classification. We provide a comprehensive procedure for the designed integrated networks and conduct extensive experiments to validate the feasibility of our approach. A subsequent analysis reinforces our ideas and observations.

Chapter 4 further investigates integrating the multiscanning strategy with Transformers to address the inherent limitations of RNNs. Furthermore, to improve our previous work, we propose several novel modules, such as the RT encoder and SMSA, to incorporate with Transformers. Our experiments provide strong validation for our approach.

Compared with other state-of-the-art (SOTA) methods, our approach delivers superior results on disjoint sampled datasets, thereby proving the heightened generative ability of the model. The outcomes of our research will bring significant advantages to the practical applications of HSI classification, facilitating more efficient, accurate, and expedited interpretation processes.

The importance of this study can be summarized as follows: The introduction of a multiscanning strategy, augmented with RNNs, Transformers, and attention mechanisms, represents a promising alternative to the widely used CNN approaches. This work not only presents a more efficient and lightweight method for image processing but also demonstrates greater practicality in real-world applications. Crucially, this approach excels in scenarios with limited data availability, enhancing the interpretation and utilization of spatial features. These advancements pave the way for new applications in various tasks, marking a versatile progression in the field of image analysis.

To sum up the whole research in this thesis and item-to-item correspondent solutions, we prepare a Table 5.1 and a Figure 5.1 for better understanding.

5.2 Future Works

As outlined in the final section of Chapter 4, we highlight several limitations of our proposed multiscanning-based RNN-Transformer and suggest potential areas of further research in HSI classification. For instance, refining spectral feature localization and globalization could improve spectral feature extraction and reduce the computational burden.

Moreover, exploring a 3D Transformer might delve deeper into the internal correlation within the HSI, enhancing our understanding of the HSI structure for potential applications in HSI reconstruction or super-resolution tasks.

Lastly, introducing sparsity via hypergraph into the Transformer architecture presents an intriguing topic, given the current dominance of the Transformer model in various fields of deep learning. This approach could enhance the efficiency and effectiveness of the Transformer by reducing redundancy in the self-attention mechanism and focusing on more relevant feature relationships.

These potential research directions could significantly contribute to further advancing the field and developing deep learning architectures. Our current work is a stepping stone towards these future explorations as we strive for more efficient and effective methods in HSI classification and beyond.

Table 5.1 Summary list of this research project.

Research Topic	1. HSI 2. HSI classification 3. Deep Learning 4. Previous works: CNNs, RNNs, Transformers, etc.
Goal	To build up an accurate and efficient HSI classification network.
Problem Setting	Initial problems set in Chapter 1: 1. Existing CNNs can not well classify small objects and boundaries. 2. Existing RNNs can not well handle noise and classify boundaries. Limitations in Chapter 2: 1. No attention for multiscanning orders. 2. Ineffective multiscanning feature summation. Limitations in Chapter 3: 1. RNN biases feature learning 2. General Transformer can not well handle boundaries. 3. RNN is not enough for multiscanning feature fusion.
Proposal	Proposals in Chapter 2: 1. Multiscanning-based pixel-level RNN. 2. Multiscanning feature summation. Proposals in Chapter 3: 1. Scanning order-based attention. 2. RNN-based multiscanning feature fusion. Proposals in Chapter 4: 1. RNN-Transformer (Spectral Transformer). 2. Soft masked self-attention. 3. Multiscanning fusion Transformer.
Implementation	Chapter 2: Multiscanning-based RNN for HSI classification. Chapter 3: HSI Classification Using Multiscanning-Based RNN with Attention Chapter 4: HSI Classification Using Multiscanning-Based RNN-Transformer

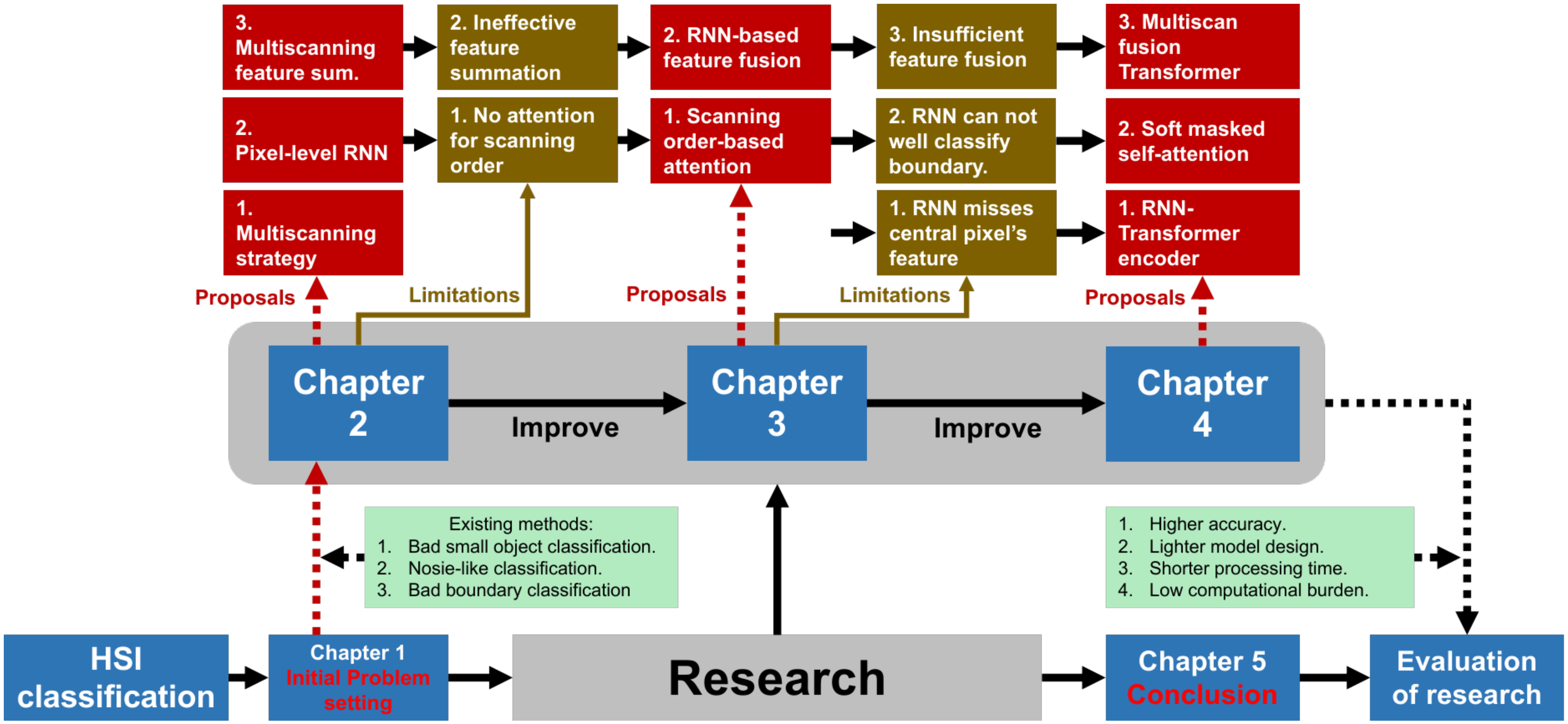

3. Multiscanning feature sum.
2. Ineffective feature summation
2. RNN-based feature fusion
3. Insufficient feature fusion
3. Multiscan fusion Transformer
2. Pixel-level RNN
1. No attention for scanning order
1. Scanning order-based attention
2. RNN can not well classify boundary.
2. Soft masked self-attention
1. Multiscanning strategy
1. RNN misses central pixel's feature
1. RNN-Transformer encoder
Proposals
Limitations
Proposals
Limitations
Proposals
Chapter 2
Improve
Chapter 3
Improve
Chapter 4
Existing methods:
1. Bad small object classification.
2. Nosie-like classification.
3. Bad boundary classification
1. Higher accuracy.
2. Lighter model design.
3. Shorter processing time.
4. Low computational burden.
HSI classification
Chapter 1
Initial Problem setting
Research
Chapter 5
Conclusion
Evaluation of research

Figure 5.1 The summarization of this research project with item-to-item limitations and solutions.

Bibliography

[1] N. A and A. A. Current advances in hyperspectral remote sensing in urban planning. In *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, pages 94–98, 2022.

[2] M. Ahmad, S. Shabbir, S. K. Roy, D. Hong, X. Wu, J. Yao, A. M. Khan, M. Mazzara, S. Distefano, and J. Chanussot. Hyperspectral image classification—traditional to deep models: A survey for future prospects. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:968–999, 2022.

[3] T. V. Bandos, L. Bruzzone, and G. Camps-Valls. Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3):862–873, 2009.

[4] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

[5] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar. 3-d deep learning approach for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(8):4420–4434, 2018.

[6] B. Chandra and R. K. Sharma. On improving recurrent neural network for image classifica-

tion. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1904–1907, 2017.

[7] K. Chen, R. Wang, M. Utiyama, and E. Sumita. Recurrent positional embedding for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1361–1367, 2019.

[8] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen, et al. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*, 2018.

[9] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.

[10] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2094–2107, 2014.

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

[12] R. Dey and F. M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1597–1600, 2017.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[14] A. Fahim, Q. Tan, M. Mazzi, M. Sahabuddin, B. Naz, and S. Ullah Bazai. Hybrid lstm self-attention mechanism model for forecasting the reform of scientific research in morocco. *Computational Intelligence and Neuroscience*, 2021:1–14, 2021.

[15] A. Fong, G. Shu, and B. McDonogh. Farm to table: Applications for new hyperspectral imaging technologies in precision agriculture, food quality and safety. In *2020 Conference on Lasers and Electro-Optics (CLEO)*, pages 1–2, 2020.

[16] K. Gao, B. Liu, Z. Xue, X. Zuo, Y. Sun, M. Dai, et al. Deep transformer network for hyperspectral image classification. *Academic Journal of Computing & Information Science*, 4(7):11–17, 2021.

[17] Y. Gao and T.-S. Chua. Hyperspectral image classification by using pixel spatial correlation. In *Advances in Multimedia Modeling*, pages 141–151, Berlin, Heidelberg, 2013.

[18] S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi, N. Zhao, and A. Tariq. Hyperspectral image classification using a hybrid 3d-2d convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:7570–7588, 2021.

[19] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):37–78, 2017.

[20] A. Graves, S. Fernández, and J. Schmidhuber. Multi-dimensional recurrent neural networks. In *International Conference on Artificial Neural Networks*, pages 549–558. Springer, 2007.

[21] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1462–1471, Lille, France, 2015.

[22] R. Hang, Z. Li, Q. Liu, P. Ghamisi, and S. S. Bhattacharyya. Hyperspectral image classification with attention-aided cnns. *IEEE Transactions on Geoscience and Remote Sensing*, 59(3):2281–2293, 2021.

[23] R. Hang, Q. Liu, D. Hong, and P. Ghamisi. Cascaded recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5384–5394, 2019.

[24] J. Hao, X. Wang, B. Yang, L. Wang, J. Zhang, and Z. Tu. Modeling recurrence for transformer. *arXiv preprint arXiv:1904.03092*, 2019.

[25] S. Hao, W. Wang, and M. Salzmann. Geometry-aware deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(3):2448–2460, 2021.

[26] J. He, L. Zhao, H. Yang, M. Zhang, and W. Li. Hsi-bert: Hyperspectral image classification using the bidirectional encoder representation from transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 58(1):165–178, 2019.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[28] L. He, J. Li, C. Liu, and S. Li. Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines. *IEEE Transactions on Geoscience and Remote Sensing*, 56(3):1579–1597, 2017.

[29] M. He, B. Li, and H. Chen. Multi-scale 3d deep convolutional neural network for hyperspectral image classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3904–3908, 2017.

[30] X. He, Y. Chen, and Z. Lin. Spatial-spectral transformer for hyperspectral image classification. *Remote Sensing*, 13(3), 2021.

[31] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[32] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot. Spectralformer: Rethinking hyperspectral image classification with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2021.

[33] W. Hu, Y. Huang, W. Li, F. Zhang, and H. Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015:1–12, 2015.

[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*, 21(5), 2021.

[35] Z. Huang, P. Xu, D. Liang, A. Mishra, and B. Xiang. Trans-blstm: Transformer with bidirectional lstm for language understanding. *arXiv preprint arXiv:2003.07000*, 2020.

[36] D. Ibanez, R. Fernandez-Beltran, F. Pla, and N. Yokoya. Masked auto-encoding spectral–spatial transformer for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.

[37] R. Jing. A self-attention based lstm network for text classification. *Journal of Physics: Conference Series*, 1207:012008, 2019.

[38] B.-C. Kuo, H.-H. Ho, C.-H. Li, C.-C. Hung, and J.-S. Taur. A kernel-based feature selection method for svm with rbf kernel for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(1):317–326, 2013.

[39] H. Lee and H. Kwon. Going deeper with contextual cnn for hyperspectral image classification. *IEEE Transactions on Image Processing*, 26(10):4843–4855, 2017.

[40] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi, and J. Hu. Classification of hyperspectral imagery using a new fully convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 15(2):292–296, 2018.

[41] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6690–6709, 2019.

[42] X. Li, C. Li, and L. Jiang. A multi-view-based noise correction algorithm for crowdsourcing learning. *Information Fusion*, 91:529–541, 2023.

[43] Y. Li, H. Zhang, and Q. Shen. Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network. *Remote Sensing*, 9(1), 2017.

[44] Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[45] X. Liu, H.-F. Yu, I. Dhillon, and C.-J. Hsieh. Learning to encode position for transformer with continuous dynamical model. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 6327–6335, 2020.

[46] Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, and G. Bai. Hsi-cnn: A novel convolution neural network for hyperspectral image. In *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 464–469, 2018.

[47] W. Lv and X. Wang. Overview of hyperspectral image classification. *Journal of Sensors*, 2020:1–13, 2020.

[48] L. Mou, P. Ghamisi, and X. X. Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655, 2017.

[49] L. Mou, P. Ghamisi, and X. X. Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655, 2017.

[50] 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 *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4, 2014.

[51] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza. Scalable recurrent neural network for hyperspectral image classification. *The Journal of Supercomputing*, 76:8866–8882, 2020.

[52] S. Pirhosseinloo and J. S. Brumberg. Dilated convolutional recurrent neural network for monaural speech enhancement. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 158–162, 2019.

[53] K. Pooja, R. R. Nidamanuri, and D. Mishra. Multi-scale dilated residual convolutional neural network for hyperspectral image classification. In *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5, 2019.

[54] S. Prasad and L. M. Bruce. Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geoscience and Remote Sensing Letters*, 5(4):625–629, 2008.

[55] W. Qi and X. Zhang. A multi-scale 3d convolution neural network for spectral-spatial classi-

fication of hyperspectral imagery. *IOP Conference Series: Earth and Environmental Science*, 502(1):012015, 2020.

[56] Y. Qing, W. Liu, L. Feng, and W. Gao. Improved transformer net for hyperspectral image classification. *Remote Sensing*, 13(11):2216, 2021.

[57] F. Racek, T. Baláž, and P. Melša. Hyperspectral data conversion in the case of military surveillance. *Advances in Military Technology*, 10(1):5–13, 2015.

[58] J. W. Rae and A. Razavi. Do transformers need deep long-range memory. *arXiv preprint arXiv:2007.03356*, 2020.

[59] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015.

[60] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*, 2017.

[61] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[62] C. Shi and C.-M. Pun. Multi-scale hierarchical recurrent neural networks for hyperspectral image classification. *Neurocomputing*, 294:82–93, 2018.

[63] B. Shuai, Z. Zuo, and G. Wang. Quaddirectional 2d-recurrent neural networks for image labeling. *IEEE Signal Processing Letters*, 22(11):1990–1994, 2015.

[64] H. Sun, X. Zheng, and X. Lu. A supervised segmentation network for hyperspectral image classification. *IEEE Transactions on Image Processing*, 30:2810–2825, 2021.

[65] H. Sun, X. Zheng, X. Lu, and S. Wu. Spectral–spatial attention network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3232–3245, 2020.

[66] H. Sun, X. Zheng, X. Lu, and S. Wu. Spectral–spatial attention network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3232–3245, 2020.

[67] L. Sun, G. Zhao, Y. Zheng, and Z. Wu. Spectral–spatial feature tokenization transformer for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.

[68] 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 *2013 5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4, 2013.

[69] K. Tran and A. Bisazza. The importance of being recurrent for modeling hierarchical structure. *arXiv preprint arXiv:1803.03585*, 2018.

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

[71] L. Van Der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[72] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[73] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and

I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA, 2017.

[74] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten. Hyperspectral image classification with independent component discriminant analysis. *IEEE transactions on Geoscience and remote sensing*, 49(12):4865–4876, 2011.

[75] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv preprint arXiv:1505.00393*, 2015.

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

[77] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[78] X. Xue, J. Feng, Y. Gao, M. Liu, W. Zhang, X. Sun, A. Zhao, and S. Guo. Convolutional recurrent neural networks with a self-attention mechanism for personnel performance prediction. *Entropy*, 21(12):1227, 2019.

[79] X. Yan, S. Hu, Y. Mao, Y. Ye, and H. Yu. Deep multi-view learning methods: A review. *Neurocomputing*, 448:106–129, 2021.

[80] X. Yang, W. Cao, Y. Lu, and Y. Zhou. Hyperspectral image transformer classification networks. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022.

[81] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

[82] 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 Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022.

[83] H. Zhang, Y. L. Li, Y. Zhang, and Q. Shen. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sensing Letters*, 8(5):438–447, 2017.

[84] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, and H. Zhou. Spatial sequential recurrent neural network for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(11):4141–4155, 2018.

[85] J. Zhao, X. Xie, X. Xu, and S. Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54, 2017.

[86] Q. Zheng, J. Zhu, Z. Li, Z. Tian, and C. Li. Comprehensive multi-view representation learning. *Information Fusion*, 89:198–209, 2023.

[87] 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 Transactions on Geoscience and Remote Sensing*, 58(8):5612–5626, 2020.

[88] 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 Transactions on Geoscience and Remote Sensing*, 58(8):5612–5626, 2020.

[89] F. Zhou, R. Hang, Q. Liu, and X. Yuan. Hyperspectral image classification using spectral-spatial lstms. *Neurocomputing*, 328:39–47, 2019.

[90] W. Zhou, S.-i. Kamata, and Z. Luo. Sub-band grouping spectral feature-attention block for

hyperspectral image classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1820–1824, 2021.

[91] W. Zhou, S.-i. Kamata, Z. Luo, and H. Wang. Multiscanning strategy-based recurrent neural network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18, 2021.

[92] W. Zhou and Seiichiro-Kamata. Multi-scanning based recurrent neural network for hyperspectral image classification. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4743–4750, 2021.

[93] Y. Zhou and Y. Wei. Learning hierarchical spectral–spatial features for hyperspectral image classification. *IEEE Transactions on Cybernetics*, 46(7):1667–1678, 2015.

[94] Z. Zuo, B. Shuai, G. Wang, X. Liu, and X. Wang. Learning contextual dependence with convolutional hierarchical recurrent neural networks. *IEEE Transactions on Image Processing*, 25(7):2983–2996, 2016.

[95] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen. Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 18–26, 2015.

[96] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen. Learning contextual dependence with convolutional hierarchical recurrent neural networks. *IEEE Transactions on Image Processing*, 25(7):2983–2996, 2016.

Related Publications by the Author

Articles in Journal Papers

- W. Zhou, S. -i. Kamata, H. Wang and X. Xue, "Multiscanning-Based RNN–Transformer for Hyperspectral Image Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1-19, 2023, Art no. 5512319, doi: 10.1109/TGRS.2023.3277014.

- W. Zhou, S. -i. Kamata, Z. Luo and H. Wang, "Multiscanning Strategy-Based Recurrent Neural Network for Hyperspectral Image Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-18, 2022, Art no. 5521018, doi: 10.1109/TGRS.2021.3138742.

- Z. Luo, Z. Sun, W. Zhou, Z. Wu, and S. -i. Kamata. "Rethinking ResNets: improved stacking strategies with high-order schemes for image classification." Complex & Intelligent Systems 8, no. 4 (2022): 3395-3407, doi: 10.1007/s40747-022-00671-3.

- Z. Luo, Z. Sun, W. Zhou, Z. Wu, and S. -i. Kamata. "Constructing infinite deep neural networks with flexible expressiveness while training." Neurocomputing 487 (2022): 257-268, doi: 10.1016/j.neucom.2021.11.010.

Peer-reviewed Articles in International Conferences

- W. Zhou, S. -i. Kamata, Z. Luo and X. Xue, "Rethinking Unified Spectral-Spatial-Based Hyperspectral Image Classification Under 3D Configuration of Vision Transformer," 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 2022, pp. 711-715, doi: 10.1109/ICIP46576.2022.9897603.

- Z. Luo, W. Zhou, S. -i. Kamata and X. Hu, "Deep Residual Networks with Common Linear Multi-Step and Advanced Numerical Schemes," 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 2022, pp. 3286-3290, doi: 10.1109/ICIP46576.2022.9897531.

- W. Zhou, S. -i. Kamata, Z. Luo and X. Chen, "Hierarchical Unified Spectral-Spatial Aggregated Transformer for Hyperspectral Image Classification," 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 2022, pp. 3041-3047, doi: 10.1109/ICPR56361.2022.9956396.

- W. Zhou, S. -i. Kamata and Z. Luo, "Sub-Band Grouping Spectral Feature-Attention Block for Hyperspectral Image Classification," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 1820-1824, doi: 10.1109/ICASSP39728.2021.9414678.

- Z. Luo, S. -i. Kamata, Z. Sun and W. Zhou, "Deep Neural Networks with Flexible Complexity While Training Based on Neural Ordinary Differential Equations," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 1690-1694, doi: 10.1109/ICASSP39728.2021.9413916.

- W. Zhou and S. -i. Kamata, "Multi-Scanning Based Recurrent Neural Network for Hyperspectral Image Classification," 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021, pp. 4743-4750, doi: 10.1109/ICPR48806.2021.9413071.

- X. Chen, S. -i. Kamata and W. Zhou, "Hyperspectral Image Classification Based on Multi-stage Vision Transformer with Stacked Samples," TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON), Auckland, New Zealand, 2021, pp. 441-446, doi: 10.1109/TENCON54134.2021.9707289.