

SMaT-HSI:Structure-aware Mamba-Transformer Hybrid Model for Hyperspectral Image Classification

by

Yaxuan Liu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2025

© Yaxuan Liu 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Hyperspectral image (HSI) classification is a crucial task in remote sensing, playing a fundamental role in environmental monitoring, precision agriculture, urban planning, and mineral exploration. By leveraging the rich spectral information across hundreds of contiguous bands, HSI classification enables precise identification of materials and land cover types, facilitating accurate mapping of vegetation, soil, water bodies, and built environments. Traditional convolutional neural network (CNN)-based methods effectively extract local spatial features, while transformer-based models excel in capturing global contextual dependencies. However, both approaches face challenges in fully leveraging the spectral and spatial dependencies inherent in hyperspectral data.

Recently, Mamba, a state-space model (SSM)-based architecture, has shown promise in sequence modeling by efficiently capturing long-range dependencies with linear computational complexity. A comprehensive comparison of CNN-based, transformer-based, and Mamba-based models for HSI classification reveals that Mamba-based models achieve performance comparable to transformer-based models, highlighting their potential in this domain. Current Mamba-based methods often convert images into one-dimensional sequences and use scanning strategies to capture local spatial and spectral dependencies. However, these approaches struggle to fully represent the intricate spectral-spatial structures in HSIs and introduce computational redundancy. To address this, a structure-aware state fusion mechanism is proposed to explicitly model the spatial and spectral relationships of neighboring features in the latent state space, enabling more efficient and accurate representation learning. To further improve the capture of global context and long-range spatial dependencies, a hybrid Mamba-transformer architecture is explored. Different integration strategies are investigated, including inserting transformer blocks in the earlier, middle, and final layers, as well as at regular intervals. Analysis indicates that incorporating a self-attention block in the final layer achieves the highest average overall accuracy of 97.58% across the five datasets.

The proposed approach is evaluated on five publicly available benchmark datasets—Indian Pines, Pavia University, Houston 2013, WHU-Hi-HanChuan, and WHU-Hi-HongHu, demonstrating an average overall accuracy improvement of 0.87% compared to the baseline model and competitive results with existing transformer-based and Mamba-based models. These findings underscore the potential of combining Mamba and transformer architectures for efficient and accurate hyperspectral image classification, offering new insights into advanced sequence modeling for remote sensing applications.

Acknowledgements

I would first and foremost like to thank my supervisors, Prof. Linlin Xu and Prof. David Clausi. Their expert advice, timely feedback, and continuous encouragement were crucial for the successful completion of this research. I am truly grateful for the time and energy they invested in my development as a researcher.

My sincere thanks also go to my fellow lab members, particularly Javier, Jayden, Lily and Muhammed. Our collaborative discussions and mutual support were incredibly valuable and made the lab a wonderful place to work.

I also wish to thank everyone else who has supported me throughout this project. Your encouragement has been invaluable.

Dedication

To my family, for your unconditional love, endless support, and the sacrifices you made that made this all possible.

Table of Contents

Author’s Declaration	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.2 Motivations	2
1.3 Related Works	3
1.4 Thesis Organization	8
2 Preliminaries	9
2.1 Hyperspectral Image Data	9
2.2 Feature Engineering-based Approaches	11
2.3 Deep Learning Approaches	13
2.3.1 Convolutional Neural Networks	13

2.3.2	Transformer Architecture	14
2.3.3	State-Space Models	16
3	Methodology	19
3.1	Architecture Overview	19
3.2	Spectral-Spatial Tokens Generation	20
3.3	Structure-Aware Spectral-Spatial Mamba Block	22
3.4	Hybrid Mamba-Transformer Architecture	25
4	Experiments and Discussion	27
4.1	Datasets	27
4.2	Comparison of Methods	36
4.2.1	Experimental setup	36
4.2.2	Evaluation Metrics	37
4.2.3	Classification Results	39
4.3	Discussion	40
4.3.1	Ablation Studies	40
4.3.2	Complexity Analysis	49
5	Conclusion	51
5.1	Summary	51
5.2	Limitations and future work	52
	References	53

List of Figures

3.1	Illustration of the proposed model’s architecture.	20
3.2	Illustration of the structure-aware state fusion mechanism in spatial and spectral block	23
3.3	Illustration of the Feature Enhancement Module	24
4.1	Indian Pines dataset. (a) False-color map. (b) Ground-truth map.	27
4.2	Pavia University dataset. (a) False-color map. (b) Ground-truth map.	29
4.3	Houston 2013 dataset. (a) False-color map. (b) Ground-truth map.	31
4.4	WHU-Hi-HanChuan dataset. (a) False-color map. (b) Ground-truth map.	32
4.5	WHU-Hi-HongHu dataset. (a) False-color map. (b) Ground-truth map.	34
4.6	Classification maps using different methods on the Indian Pines dataset.	41
4.7	Classification maps using different methods on the Pavia University dataset.	42
4.8	Classification maps using different methods on the Houston 2013 dataset.	43
4.9	Classification maps using different methods on the WHU-Hi-HanChuan dataset.	45
4.10	Classification maps using different methods on the WHU-Hi-HongHu dataset.	46

List of Tables

4.1	Land cover classes and numbers of samples in the Indian Pines dataset. . .	28
4.2	Land cover classes and numbers of samples in the Pavia University dataset.	30
4.3	Land cover classes and numbers of samples in the Houston 2013 dataset. .	31
4.4	Land cover classes and numbers of samples in the WHU-Hi-HanChuan dataset.	33
4.5	Land cover classes and numbers of samples in the WHU-Hi-HongHu dataset.	35
4.6	Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the Indian Pines dataset. The best is in bold.	40
4.7	Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the Pavia University dataset. The best is in bold.	40
4.8	Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the Houston 2013 dataset. The best is in bold.	41
4.9	Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the WHU-Hi-HanChuan dataset. The best is in bold.	44
4.10	Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the WHU-Hi-HongHu dataset. The best is in bold.	44
4.11	Ablation experiment with different component in model on five datasets . .	48
4.12	Comparison of different combinations of Transformer and Mamba blocks across five datasets.	49
4.13	Comparison of different models on Parameters and FLOPs.	50

Chapter 1

Introduction

1.1 Background

Hyperspectral imaging (HSI) is an advanced imaging modality that captures detailed spectral information across numerous contiguous, narrow wavelength bands, typically ranging from visible to near-infrared or short-wave infrared regions. Unlike conventional multispectral or RGB imaging, hyperspectral sensors generate three-dimensional datasets known as hypercubes, encompassing two spatial dimensions and one spectral dimension, thus allowing for precise spectral analysis at the pixel level [10].

The significance of hyperspectral imaging lies in its ability to discern subtle spectral differences among various materials, objects, or biological tissues, enabling more accurate detection, identification, and classification tasks. Due to its rich spectral resolution, hyperspectral imagery has been demonstrated to substantially improve the accuracy and robustness of classification tasks compared to traditional imaging techniques.

Hyperspectral imaging has found extensive applications across numerous domains. In remote sensing, HSI facilitates precise land cover mapping, vegetation analysis, and environmental monitoring by differentiating spectral signatures associated with various land surface features [10]. Precision agriculture extensively utilizes hyperspectral technology to identify crop species, assess plant health, detect stress and diseases early, and optimize farming practices [1]. In medical fields, HSI has emerged as a non-invasive diagnostic tool for classifying healthy and pathological tissues, notably in oncology for tumor identification and surgical guidance [23]. Additionally, industrial sectors employ hyperspectral imaging for quality inspection and automated sorting based on spectral characterization of products and materials.

With recent advances in machine learning, particularly deep learning, hyperspectral image classification has gained further momentum, achieving unprecedented levels of accuracy and efficiency [21]. These developments underscore the growing importance and broadening applications of hyperspectral imaging in contemporary research and practical scenarios.

1.2 Motivations

The Promise of State Space Models for HSI Classification

In recent years, models based on convolutional neural networks (CNNs) and Transformers have achieved remarkable success in hyperspectral image (HSI) classification tasks. CNNs excel at capturing local spatial features due to their inherent inductive bias, but their limited receptive fields often make it not feasible to learn long-range spatial dependencies[18]. On the other hand, models represented by the Vision Transformer (ViT) effectively address the long-range dependency problem through their self-attention mechanism[9, 17]. However, their quadratic computational complexity presents a significant computational burden, especially when processing high-dimensional, high-resolution hyperspectral data.

To overcome these challenges, this thesis explores the application of modern state space models (SSMs), particularly the Mamba architecture[13]. Compared to Transformers, Mamba reduces computational complexity to a linear level while maintaining the ability to have a global receptive field. This efficiency is achieved through its selective scan mechanism, which builds upon foundational work in structured state spaces[14] and is exceptionally well-suited for handling long sequences, such as the spectral signatures of HSI pixels. When compared to both CNNs and Transformers, SSMs present a compelling alternative that promises a superior balance between classification accuracy and computational efficiency. Therefore, investigating SSM-based models for HSI classification is a promising direction for advancing the state-of-the-art[33].

Enhancing SSMs with a Structure-Aware State Fusion Mechanism

While Mamba demonstrates outstanding performance on sequential data, its original formulation is primarily designed for one-dimensional sequences. Applying it to 2D or 3D image data typically requires a "flattening" process, which disrupts the intrinsic spatial structure of the image. Furthermore, conventional scanning strategies, such as bidirectional scanning, can introduce computational redundancy without optimally leveraging the spatial and spectral dependencies inherent in HSI data cubes.

To address this issue, we propose a novel structure-aware state fusion mechanism within the SSM framework. This mechanism is designed to establish neighborhood connectivity directly in the spectral and spatial state space, enabling simultaneous information fusion across both domains. The goal of explicitly incorporating structural awareness into sequence models is a recognized challenge in the field, with other works exploring methods to achieve similar ends[31]. Our design allows the model to more naturally understand and leverage the cube structure of hyperspectral images, thereby enhancing its ability to extract joint spectral-spatial features within local regions and reducing the computational cost associated with redundant scanning.

A Hybrid Mamba-Transformer Architecture for Global Context Modeling

While our proposed structure-aware state fusion mechanism strengthens the model’s understanding of local structural information, the self-attention mechanism of the Transformer still holds a unique advantage in capturing global context and ultra-long-range spatial dependencies[30]. In certain complex global layout scenarios, the information aggregation capability of a standalone Mamba model may still have room for improvement.

Therefore, we further explore a hybrid Mamba-Transformer architecture. This architecture aims to combine Mamba’s efficient sequence modeling capabilities with the Transformer’s powerful global context-capturing abilities, a synergistic approach that is gaining traction in the wider machine learning community. Through this hybrid design, the model can not only leverage Mamba to efficiently process spectral sequences and local spatial dependencies but also utilize the Transformer’s self-attention mechanism to integrate feature information from the entire image at a higher level. This allows for a more comprehensive understanding of the spatial distribution patterns of land cover in complex scenes. We expect this hybrid architecture to further enhance the accuracy and robustness of hyperspectral image classification while maintaining computational efficiency.

1.3 Related Works

Over the past decades, the landscape of HSI classification has evolved significantly, progressing from traditional machine learning methods reliant on handcrafted features, through convolutional neural networks (CNNs) capable of automatic deep feature learning, to Transformers adept at capturing global dependencies, and most recently, to state space models (Mamba) that offer a compelling trade-off between performance and efficiency. This chapter provides a comprehensive review and critique of these principal methodological paradigms.

Methods Based on Traditional Machine Learning

Prior to the advent of deep learning, HSI classification was predominantly governed by traditional machine learning algorithms. These methods typically operate in two stages: feature extraction and classification. Among these, the support vector machine (SVM) stands out as a representative method[26]. SVM leverages the kernel trick (e.g., the radial basis function (RBF) kernel) to map non-linear spectral features into a high-dimensional space, wherein an optimal separating hyperplane is identified. Its robustness in small-sample scenarios established it as a benchmark model for HSI classification. Another widely adopted approach is ensemble learning, notably random forest (RF)[3], which constructs an ensemble of decision trees for classification, exhibiting strong adaptability to high-dimensional data and relative computational efficiency during training. Nevertheless, the efficacy of these conventional approaches is heavily contingent upon the design of handcrafted features. Furthermore, they primarily focus on pixel-wise spectral information, largely neglecting the spatial context, which is crucial for enhancing classification accuracy. This oversight constitutes their fundamental limitation.

Methods Based on Convolutional Neural Networks

The advent of Convolutional Neural Networks (CNNs) marked a paradigm shift in HSI classification. Their inherent ability to extract local features and share weights enables the automatic learning of joint spatial-spectral representations, leading to significant advancements in accuracy.

The architectural evolution of CNNs for HSI reflects a progressively sophisticated approach to data fusion. Early 1D-CNNs [18] analyzed pixel-wise spectral signatures but inherently ignored spatial context. To address this, 2D-CNNs were introduced to operate on spatial patches, though they often required a preliminary dimensionality reduction step like Principal Component Analysis (PCA), which risked the loss of critical spectral information. The breakthrough came with 3D-CNNs [4], which offered an end-to-end solution. By employing 3D kernels directly on HSI data cubes, these models simultaneously extract deep features from both spatial and spectral dimensions, establishing them as a dominant and highly effective methodology in the field.

A significant research trajectory has been the construction of deeper networks to learn more complex features. For instance, the Spectral-Spatial Residual Network (SSRN)[37] innovatively incorporated residual connections to construct separate spectral and spatial residual blocks. This design enables the construction of much deeper networks that effectively circumvent the vanishing gradient problem, thereby learning more discriminative

spatial-spectral features. Another prominent direction involves the integration of attention mechanisms. The Double-Branch Multi-Attention (DBMA) network[24], for example, employs parallel spectral and spatial branches, each embedded with attention modules. This allows the network to adaptively focus on critical spectral bands and spatial regions, thereby enhancing feature representation. Furthermore, researchers have explored advanced feature fusion strategies. The Cross-Level Optimal Learning Network (CLOLN)[32], for instance, leverages optimal transport theory to effectively integrate features from different network depths. This approach adaptively learns an optimal fusion scheme between shallow, detail-rich features and deep, semantic features, yielding a more comprehensive and discriminative feature representation.

While these sophisticated designs have elevated the performance of CNNs in HSI classification, they also highlight an inherent limitation: the difficulty in capturing long-range dependencies, a consequence of the local receptive field of the convolutional kernel.

Methods Based on Transformers

To overcome the limitations of CNNs in modelling global information, the Transformer model, originating from the field of natural language processing, was introduced to the task of HSI classification. The core of the Transformer lies in its self-attention mechanism, which evaluates the relevance between any two elements in a sequence by computing Query, Key, and Value vectors. This enables the model to directly capture global dependencies within the image, thereby breaking the local receptive field constraints inherent in convolutional operations.

The Vision Transformer (ViT) [9] was a pioneering work that successfully adapted this architecture for computer vision. Its standard procedure involves flattening input image patches into a one-dimensional sequence of tokens, embedding them into a high-dimensional space via a linear projection layer, and adding positional encodings to retain spatial information. This sequence is then fed into a standard Transformer encoder for global relationship modelling. However, applying ViT directly to HSI by indiscriminately flattening the entire spatial-spectral data cube can disrupt the fine-grained spectral structure. To address this issue, researchers have proposed several sophisticated variants tailored to the unique properties of HSI

SpectralFormer[17] posits that adjacent spectral bands are highly correlated and introduces a group-wise spectral embedding strategy. This approach partitions consecutive spectral bands into several groups, with each group treated as a single entity to generate a "spectral token." This method preserves local spectral continuity during the tokeniza-

tion stage. These spectral tokens are subsequently processed by a Transformer to learn long-range dependencies between different spectral regions.

SSFTT (Spectral-Spatial Feature Tokenization Transformer)[29] adopts a decoupled approach to spatial-spectral feature learning. It constructs a dual-branch architecture, where one branch is dedicated to extracting spectral features and the other to spatial features. This design allows the model to learn information from each dimension more effectively and subsequently fuse them using mechanisms like cross-attention, thus preventing interference between spatial and spectral information.

MorphFormer[27] introduces an innovative learnable morphological transformer. The model integrates morphological convolution operations with the attention mechanism through a learnable spectral and spatial morphological network. This design is intended to enhance the interaction of structural and shape information between the HSI tokens and the classification (CLS) token. In this way, the model can better capture the structural and shape information in hyperspectral images, achieving superior performance on multiple widely used HSI datasets.

GSCViT (Group-wise Spectral-Spatial Convolutional ViT)[35] is an efficient hybrid model that synergizes the efficiency of group-wise convolutions with the global modeling capacity of Transformers. Specifically, within its Transformer blocks, it replaces a portion of the fully connected layers with group-wise convolutions. This allows for the efficient aggregation of local spatial information before the self-attention computation, significantly reducing the computational cost while simultaneously enhancing the model’s local feature extraction capabilities.

Although these models have demonstrated the substantial advantages of Transformers in modeling global context, the quadratic computational complexity ($O(N^2)$) of the self-attention mechanism remains a formidable efficiency challenge when processing high-dimensional HSI data, which can contain hundreds or even thousands of spectral bands.

Methods Based on Mamba

To address the computational bottleneck of Transformer models, Mamba was proposed as a novel architecture integrating concepts from state space models (SSMs). The core of Mamba is its selective scan mechanism (S4) [14]. Unlike self-attention, which necessitates computing interactions for all pairs of elements, the S4 mechanism “scans” the sequence in a manner akin to a recurrent neural network (RNN). It dynamically updates a compressed “state” based on the input, with this state encapsulating all prior sequence information. By selectively allowing information to flow into or be forgotten from this state, Mamba can

model long-range dependencies with linear complexity, $O(N)$. This approach drastically reduces computational costs while demonstrating performance that is comparable, and in some cases superior, to that of Transformers. This efficient and powerful new paradigm was rapidly adopted in the HSI classification domain, leading to a series of innovative designs:

MambaHSI[22] introduces a novel framework for remote sensing HSI classification based on the Mamba model. Its core innovation lies not in simply flattening the data, but in meticulously designed spatial and spectral Mamba modules for separate processing and modeling. The model first employs a spatial Mamba block to capture long-range spatial dependencies across the entire image at the pixel level. Subsequently, a dedicated spectral Mamba block processes the spectral information by dividing the spectral vectors into groups to mine the interrelationships between different spectral groups, thereby extracting discriminative spectral features. Finally, a spatial-spectral fusion module adaptively integrates these two feature types. As the first image-level, Mamba-based HSI classification model, MambaHSI achieves comprehensive and adaptive modeling of both spatial and spectral information while maintaining the linear computational complexity of the Mamba architecture.

MiM (Mamba-in-Mamba)[38] proposes a more intricate, nested Mamba architecture. The outer Mamba treats HSI patches as tokens, responsible for modeling macroscopic spatial relationships between these patches. To process the internal information within each patch, an inner Mamba is invoked, which employs a centralized cross-scan pattern (scanning outwards from the center) to efficiently aggregate local spatial-spectral information. This strategy, which combines macroscopic and microscopic perspectives, enables the model to capture features comprehensively.

SSMamba (Spectral-Spatial Mamba)[19] adopts a parallel processing strategy for spatial-spectral fusion. The model first converts HSI cubes into separate spatial and spectral token sequences. Within its core Spatial-Spectral Mamba Block (SS-MB), the model uses two independent Mamba branches in parallel to process these respective tokens, facilitating the synchronous extraction of spatial and spectral features. Crucially, the block incorporates a feature enhancement mechanism that allows for information interaction and fusion between the spatial and spectral branches within each processing block, thereby constructing deeply-coupled spatial-spectral feature representations.

These advanced Mamba-based models have not only achieved new heights in performance but, more importantly, have provided a computationally feasible and efficient pathway for processing the ever-increasing spatial and spectral resolutions of future remote sensing data.

1.4 Thesis Organization

The remainder of this thesis is structured as follows:

Chapter 2 provides the necessary background and surveys related work. It begins by defining the fundamental structure of HSI data. The chapter then reviews traditional machine learning approaches. Subsequently, it chronicles the evolution of deep learning methodologies, starting with convolutional neural networks (CNNs), followed by the Transformer architecture, and culminating with the latest advancements in state-space models (SSMs).

Chapter 3 presents the detailed architecture of our proposed model. It elaborates on the proposed method for generating distinct spectral and spatial tokens from the HSI data. The core of the chapter details the spectral-spatial Mamba block, including its constituent spatial Mamba block, spectral Mamba block, and the feature fusion module that integrates their outputs. Finally, an alternative hybrid Mamba-Transformer architecture is proposed and described.

Chapter 4 is dedicated to the empirical validation of the proposed method. This chapter first introduces the five public datasets used for evaluation and specifies the experimental setup, including the baseline methods and evaluation metrics. It presents and analyzes the classification results in comparison to state-of-the-art methods. A series of comprehensive ablation studies are then conducted to validate the effectiveness of the model’s key components. The chapter concludes with a complexity analysis, evaluating the model’s computational efficiency in terms of parameters, and FLOPs, as well as a hyperparameter analysis to assess the impact of key settings.

Chapter 5 concludes the thesis. It summarizes the key findings and contributions of this research. Finally, it discusses the limitations of the current work and suggests potential directions for future research.

Chapter 2

Preliminaries

2.1 Hyperspectral Image Data

Hyperspectral imaging (HSI) data is mathematically described as a three-dimensional data cube, often represented by a tensor. This structure contains both spatial and spectral information, providing a detailed profile for the materials within the captured scene [20].

The HSI Data Cube

A hyperspectral data cube, denoted as I , can be mathematically represented as a three-dimensional array with dimensions $(M \times N \times L)$. M and N represent the spatial dimensions of the image, corresponding to the number of rows and columns of pixels. L represents the spectral dimension, which is the number of contiguous spectral bands [10].

The value of each element, $I(i, j, \lambda)$, in the cube denotes the radiance or intensity measured at the spatial position (i, j) for the spectral band indexed by λ . The formula for the data cube can be expressed as:

$$I \in \mathbb{R}^{M \times N \times L} \quad (2.1)$$

where $i = 1, 2, \dots, M$; $j = 1, 2, \dots, N$; $\lambda = 1, 2, \dots, L$

The Spectral Signature

Each pixel in the hyperspectral image contains a continuous spectrum, known as its spectral signature. This signature provides detailed information about the material at that

specific spatial location. Mathematically, the spectral signature for a single pixel at spatial coordinates (i, j) is a vector $p_{i,j}$ of length L . This vector contains the intensity values across all spectral bands for that pixel [28]. The formula for a pixel vector is:

$$p_{i,j} = [I(i, j, \lambda_1), I(i, j, \lambda_2), \dots, I(i, j, \lambda_L)]^T \in \mathbb{R}^L \quad (2.2)$$

Here, the superscript T denotes the transpose, indicating that it is a column vector. This vector is a fundamental unit for many hyperspectral data analysis techniques, including material identification and classification.

The Spatial Signature

Describes the spatial pattern of pixels within a single spectral band (represented as a matrix) or, more commonly, the spatial pattern of the neighbourhood surrounding a single pixel (represented as a vector or matrix). Here are the common mathematical methods for representing a spatial signature.

In many classification or feature extraction algorithms, the primary interest is in the spatial pattern within the neighborhood of a specific pixel. This local spatial signature is more effective at capturing information like texture, edges, and structure [2]. It is typically extracted using a "window" centred on the target pixel. Let's say we want to extract the local spatial signature for the pixel at coordinates (i, j) within the k -th spectral band. We can define a neighbourhood window of size $w \times w$ (where w is usually an odd integer, like 3, 5, or 7). The values of all pixels within this window constitute the local spatial signature of the central pixel. This can be represented as a matrix or a vector.

a) Matrix Representation:

The $w \times w$ window $W_{i,j,k}$ centered at pixel (i, j) is represented as:

$$W_{i,j,k} = \begin{pmatrix} I\left(i - \frac{w-1}{2}, j - \frac{w-1}{2}, \lambda_k\right) & \cdots & I\left(i - \frac{w-1}{2}, j + \frac{w-1}{2}, \lambda_k\right) \\ \vdots & I(i, j, \lambda_k) & \vdots \\ I\left(i + \frac{w-1}{2}, j - \frac{w-1}{2}, \lambda_k\right) & \cdots & I\left(i + \frac{w-1}{2}, j + \frac{w-1}{2}, \lambda_k\right) \end{pmatrix} \in \mathbb{R}^{w \times w} \quad (2.3)$$

b) Vector Representation (Flattening):

To make it easier for machine learning algorithms to process, this $w \times w$ matrix is often "flattened" into a one-dimensional vector. This vector, $v_{i,j,k}$, has a dimension of w^2 .

$$v_{i,j,k} = \text{vec}(W_{i,j,k}) \in \mathbb{R}^{w^2} \quad (2.4)$$

Here, $vec()$ is the vectorization operator that stacks the columns or rows of a matrix into a single column vector. For example, flattening row-by-row gives:

$$\mathbf{v}_{i,j,k} = \left[I \left(i - \frac{w-1}{2}, j - \frac{w-1}{2}, \lambda_k \right), \dots, I \left(i + \frac{w-1}{2}, j + \frac{w-1}{2}, \lambda_k \right) \right]^T \quad (2.5)$$

This vector $v_{i,j,k}$ becomes the local spatial feature signature for pixel (i, j) in band k .

The Spatio-Spectral Signature

More advanced algorithms combine both spatial and spectral information. This is achieved by extracting a 3D neighborhood patch around a pixel. For a pixel (i, j) , we extract not only its $w \times w$ spatial neighborhood but also retain the full spectral information for all pixels within that neighborhood [5, 10]. This 3D data patch, $P_{i,j}$, can be represented as:

$$P_{i,j} \in \mathbb{R}_{w \times w \times L} \quad (2.6)$$

where L is the total number of spectral bands. This 3D patch can also be flattened into a very long vector of dimension $w \times w \times L$ for use in subsequent classification or regression tasks.

2.2 Feature Engineering–based Approaches

In the realm of hyperspectral image (HSI) analysis, machine learning (ML) has become a cornerstone for classification tasks, enabling the differentiation of materials and land cover types with high precision. Among the plethora of ML algorithms, random forest (RF) and support vector machine (SVM) have emerged as two of the most effective and widely adopted methods [26, 15]. This section provides a brief introduction to these two powerful classifiers and their application in HSI.

Random Forest (RF)

Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time. For HSI classification, each pixel, represented by its spectral vector x , is fed through the forest of trees. Each tree in the ensemble, assuming there are B trees, produces its own prediction, $h_b(x)$ for the class label. The final prediction

of the RF model, $H(x)$, is determined by aggregating the predictions of all the individual trees, typically through a majority vote [3]:

$$H(\mathbf{x}) = \arg \max_c \sum_{b=1}^B I(h_b(\mathbf{x}) = c) \quad (2.7)$$

where $I(\cdot)$ is the indicator function, and c represents one of the possible classes. This dual randomness—training each tree on a random subset of the data and considering a random subset of spectral bands at each node—helps to mitigate overfitting, a common problem when dealing with high-dimensional data like HSI.

The inherent structure of RF makes it particularly well-suited for HSI classification. It can handle the high dimensionality of hyperspectral data, is robust to noise, and can effectively manage situations with more features (spectral bands) than samples [12]. Furthermore, RF provides valuable by-products, such as feature importance scores, which can be used to identify the most discriminative spectral bands for a given classification task.

Support Vector Machine (SVM)

The support vector machine is a powerful supervised learning model that is highly effective for classification. The fundamental principle of SVM is to find an optimal hyperplane that maximally separates the different classes in the feature space [7]. For a given input pixel vector x , the linear SVM decision function is formulated as:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \mathbf{b} \quad (2.8)$$

where w is the weight vector normal to the hyperplane and b is the bias term. The classification is then performed based on the sign of $f(x)$. To find the optimal hyperplane, SVM solves an optimization problem that maximizes the margin (the distance between the hyperplane and the nearest data points, or support vectors) between classes.

In the context of HSI classification, the relationships between spectral signatures are often complex and non-linear. SVMs address this by employing the "kernel trick," which implicitly maps the input data into a higher-dimensional feature space where a linear separation is more likely. The decision function then becomes:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \quad (2.9)$$

where $\{\mathbf{x}_i, y_i\}_{i=1}^N$ are the training samples and their labels, α_i are learned coefficients, and $K(\cdot, \cdot)$ is the kernel function (e.g., radial basis function, RBF). The ability to model

non-linear boundaries via kernels makes SVMs a robust and accurate choice for analyzing complex hyperspectral data, especially when training samples are limited [26].

2.3 Deep Learning Approaches

While traditional methods like RF and SVM are effective, deep learning models have become the new standard for HSI classification by automatically learning hierarchical features [21]. This paradigm shift has introduced several key architectures. Convolutional neural networks (CNNs) excel at simultaneously extracting local spatial and spectral features, treating the HSI data as an image cube [5]. More recently, the Vision Transformer (ViT) has been adapted for HSI, using its self-attention mechanism to model global, long-range dependencies between spectral bands [17]. Emerging at the forefront are state space models (SSMs), which offer an efficient and powerful new approach to process entire spectral and spatial sequences, presenting a compelling alternative to the computational demands of Transformers [33].

2.3.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a powerful class of deep learning models ideal for hyperspectral image (HSI) classification because they automatically learn discriminative features. The core operation is the convolution, where a set of learnable filters (or kernels, W) slide across the input HSI data patch (X) to produce feature maps. The value of each neuron in a feature map is calculated by:

$$v = g(\mathbf{W} * \mathbf{X} + \mathbf{b}) \quad (2.10)$$

Here, $*$ denotes the convolution operation, $\mathbf{b}$ is a learnable bias term, and g is a non-linear activation function (like ReLU) that allows the network to learn complex, non-linear patterns from the data.

The strength of CNNs in HSI lies in their ability to learn joint spatial-spectral features [25]. By processing a 3D data patch, the filters learn to recognize patterns that involve both spatial textures and spectral signatures simultaneously.

After several layers of convolutions that extract increasingly complex features, a final classifier (typically a fully connected layer with a softmax function) uses these high-level features to predict the class label of the central pixel. The output is a vector of probabilities,

where the final class $\hat{y}$ is the one with the highest probability:

$$\hat{y} = \arg \max_c \left(\frac{e^{z_c}}{\sum_j e^{z_j}} \right) \quad (2.11)$$

In this formula, z_c is the output score from the network for class c . This end-to-end process, from raw pixel data to final class probability, allows CNNs to achieve good performance in HSI classification.

2.3.2 Transformer Architecture

Adapting the Transformer architecture for hyperspectral image (HSI) classification hinges on effectively converting the 3D HSI data cube into a sequential format that the Transformer can process. This section details the step-by-step journey of a hyperspectral data patch through this architecture, from input preparation to final classification [17].

Input Stage: From HSI Data Cube to Embedded Vector Sequence

The input to a Transformer is a two-dimensional sequence of vectors. However, the original data from a hyperspectral image (HSI) is a three-dimensional cube ($\mathbb{R}^{H \times W \times B}$), where H and W represent the spatial dimensions, and B is the number of spectral bands. To reconcile this structural difference, the following preprocessing steps are necessary:

- 1. Patch Extraction:** First, for each central pixel of interest within the image, a surrounding neighborhood is extracted to form a three-dimensional data patch with dimensions $P \times P \times B$. For instance, a $9 \times 9 \times 100$ data patch represents a 9×9 spatial window and incorporates information from all 100 spectral bands. This patch, which contains both local spatial information and complete spectral information, serves as the fundamental unit for the model's analysis.

- 2. Tokenization:** The extracted data patch must be converted into a sequence of tokens. A predominant method is to partition the $P \times P$ spatial window into smaller sub-patches. For example, a 9×9 spatial window can be divided into nine 3×3 sub-patches. Each $3 \times 3 \times B$ sub-patch is then flattened into a one-dimensional vector. Consequently, the HSI data patch is transformed into a sequence containing 9 tokens.

- 3. Linear Embedding:** The dimensionality of the raw token vectors may be excessively high and thus not suitable for direct processing. Therefore, a learnable linear projection layer (i.e., a fully-connected layer) is employed to map each token to a fixed,

higher-dimensional embedding space, $\mathbb{R}^{d_{model}}$. At this point, we obtain a matrix with dimensions (N, d_{model}) , where N is the number of tokens (9 in this example) and d_{model} is the latent dimension of the model. This matrix constitutes the formal input for the Transformer encoder.

Positional Encoding

The self-attention mechanism is inherently permutation-invariant, meaning it cannot perceive the order of tokens within a sequence. In the context of HSI analysis, however, the spatial position of a sub-patch—whether it is at the center or a corner—is of critical importance. To address this limitation, it is essential to incorporate positional encoding.

Before the token embeddings are fed into the encoder, a positional encoding vector, which has the same dimension (d_{model}) as the embeddings, is added to each token’s embedding vector. This positional encoding is typically generated using fixed sine and cosine functions or is implemented as a learnable parameter, and hyperspectral-image (HSI) Transformers adopt the same two approaches [29, 17]. Its function is to inject information about each token’s absolute or relative position within the original $P \times P$ spatial window. By doing so, the model can differentiate between, for instance, spectral features from the top-left corner and those from the bottom-right corner, even if their corresponding tokens appear at different positions within the input sequence.

The Encoder Layer

The sequence is passed through a stack of L identical encoder layers. The core of each layer is the multi-head self-attention mechanism, as defined in the original Transformer paper [30].

First, the input sequence from the previous layer, X^{l_1} , is linearly transformed into three matrices: a query (Q), a key (K), and a value (V). In the multi-head scheme, these are projected separately for each of the h attention heads:

$$Q_j = \mathbf{X}^{(l-1)} \mathbf{W}_j^Q, \quad K_j = \mathbf{X}^{(l-1)} \mathbf{W}_j^K, \quad V_j = \mathbf{X}^{(l-1)} \mathbf{W}_j^V \quad (2.12)$$

where $j = 1, \dots, h$ is the head index, and $\mathbf{W}_j^Q$, $\mathbf{W}_j^K$, $\mathbf{W}_j^V$ are the corresponding learnable weight matrices for that head. Each head then independently performs scaled dot-product attention:

$$\text{head}_j = \text{Attention}(Q_j, K_j, V_j) = \text{softmax} \left(\frac{Q_j K_j^T}{\sqrt{d_k}} \right) V_j \quad (2.13)$$

The scaling factor $\sqrt{d_k}$, where d_k is the dimension of the key vectors (typically d_{model}/h), is used to maintain stable gradients. This function computes a weighted aggregation of values for every HSI sub-patch token, where the weights are determined by the similarity between tokens. Finally, the outputs of all heads are concatenated and projected back to the original dimension:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.14)$$

where W^O is a learnable output projection matrix.

Following the aggregation of global contextual information by the self-attention layer, the representation vector for each token is processed independently by a fully-connected feed-forward network. This network applies a non-linear transformation to its inputs. This step can be interpreted as a further refinement and deepening of the context-aware features, thereby enhancing the expressive power of the model.

Each sub-layer is followed by a residual connection and layer normalization. This architecture ensures stable information flow throughout the model and contributes to efficient training. As the data patch passes through the entire encoder stack (e.g., six encoder layers), the output is a set of token vectors of the same quantity as the input, but with a significantly richer representation that has fully integrated the global spatio-spectral context.

Through this process, the Transformer architecture successfully deconstructs a three-dimensional hyperspectral data patch into a sequence. By leveraging the self-attention mechanism, it deeply mines the complex internal spatio-spectral correlations, ultimately achieving high-precision classification.

2.3.3 State-Space Models

State space models (SSMs) originate from classical control theory and describe the evolution of a system over time through an implicit latent state $h(t) \in \mathbb{R}^N$. In the context of deep learning, they have been developed as a highly efficient architecture for sequence modeling [14]. A continuous-time linear time-invariant (LTI) SSM system can be defined by the following differential equations:

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t) \\ y(t) &= \mathbf{C}h(t) + \mathbf{D}x(t) \end{aligned} \quad (2.15)$$

where $x(t) \in \mathbb{R}^1$ is the input signal, $y(t) \in \mathbb{R}^1$ is the output signal, $A \in \mathbb{R}^{N \times N}$ is the state transition matrix, and $B \in \mathbb{R}^{N \times 1}$ and $C \in \mathbb{R}^{1 \times N}$ are the input and output matrices, respectively.

To be applied to discrete sequence data (such as text or time series), the continuous system must be discretized. Using a timestep Δ , the continuous parameters (A, B) can be converted to discrete parameters $(\bar{A}, \bar{B})$ via methods like the zero-order hold (ZOH) [14]:

$$\begin{aligned}\bar{\mathbf{A}} &= \exp(\Delta \mathbf{A}) \\ \bar{\mathbf{B}} &= (\exp(\Delta \mathbf{A}) - \mathbf{I})(\Delta \mathbf{A})^{-1} - \Delta \mathbf{B}\end{aligned}\tag{2.16}$$

And the discretization SSM can then be calculated in recurrence denoted by the following equations:

$$\begin{aligned}h_t &= \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t \\ y_t &= \mathbf{C}h_t + \mathbf{D}x_t\end{aligned}\tag{2.17}$$

where x_t is the t -th element of the input sequence. Traditional SSMs such as S4 [14], due to the time-invariant nature of their parameters $(\bar{A}, \bar{B})$, can be unrolled into an equivalent convolutional kernel for highly efficient parallel computation. However, this same time-invariance limits the model’s ability to dynamically adjust its behavior based on input content, restricting its performance on complex, content-aware tasks.

For fast and efficient parallel training, the above process in recurrence can also be reformulated using convolution:

$$\mathbf{y} = \mathbf{x} * \bar{\mathbf{K}}\tag{2.18}$$

where $\bar{\mathbf{K}}$ denotes the structured convolutional kernels. It is obtained by

$$\bar{\mathbf{K}} = (\mathbf{C}\bar{\mathbf{B}}, \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}}, \dots, \mathbf{C}\bar{\mathbf{A}}^{N-1}\bar{\mathbf{B}})\tag{2.19}$$

where N denoted the input sequence’s length. However, this same time-invariance limits the model’s ability to dynamically adjust its behavior based on input content, restricting its performance on complex, content-aware tasks [13].

Mamba’s core breakthrough is the introduction of a selection mechanism, which transforms the linear time-invariant-Invariant (LTI) system to a time-varying one, thereby endowing the model with content-awareness [13]. This is achieved by making the key SSM parameters a function of the input data. Specifically, Mamba allows the discretization timestep Δ , the input matrix B , and the output matrix C to be dynamically generated from the current input x_k :

$$\begin{aligned}\Delta_t &= \text{softplus}(\text{Linear}_\Delta(x_t)) \\ \mathbf{B}_t &= \text{Linear}_B(x_t) \\ \mathbf{C}_t &= \text{Linear}_C(x_t)\end{aligned}\tag{2.20}$$

where *Linear* denotes a linear projection layer and the *softplus* function ensures the timestep Δ_t is positive. The state transition matrix A remains fixed, but the discretized parameters $\bar{\mathbf{A}}_t$ and $\bar{\mathbf{B}}_t$ now depend on the per-timestep Δ_t and thus also become time-varying.

Chapter 3

Methodology

3.1 Architecture Overview

The architecture of our proposed model is systematically structured into three primary stages, as illustrated in the Figure 3.1: (1) Spectral-Spatial Token Generation, (2) Spectral-Spatial Feature Extraction, and (3) Classification. This design facilitates the concurrent processing and fusion of spectral and spatial information from hyperspectral imagery (HSI).

In the initial stage, the input HSI data cube is partitioned into several patches. The objective of the model is to predict the category of the central pixel in each patch. For each patch, we employ a bidirectional sweeping scan method to generate two distinct sequences of embeddings: spatial tokens and spectral tokens. This tokenization process effectively linearizes the 2D spatial context and the 1D spectral signatures, preparing them for sequential modeling. The core of our model is the spectral-spatial feature extraction stage, which is composed of a series of stacked spectral-spatial Mamba blocks. Each block features a dual-branch design to handle the decoupled tokens.

The Spatial Feature Extraction Branch processes the sequence of spatial tokens. It leverages a state space model (SSM) to compute state variables based on state transition equations. A key innovation in this branch is the structure-aware state fusion kernel. After the initial SSM computation, the state variables are reshaped from a sequence back into a 2D spatial format. The fusion kernel then applies dilated convolutions to re-weight and merge neighboring state variables, effectively expanding the receptive field and capturing contextual structural information. These refined, structure-aware state variables are subsequently passed to the observation equation to produce the final output sequence for the

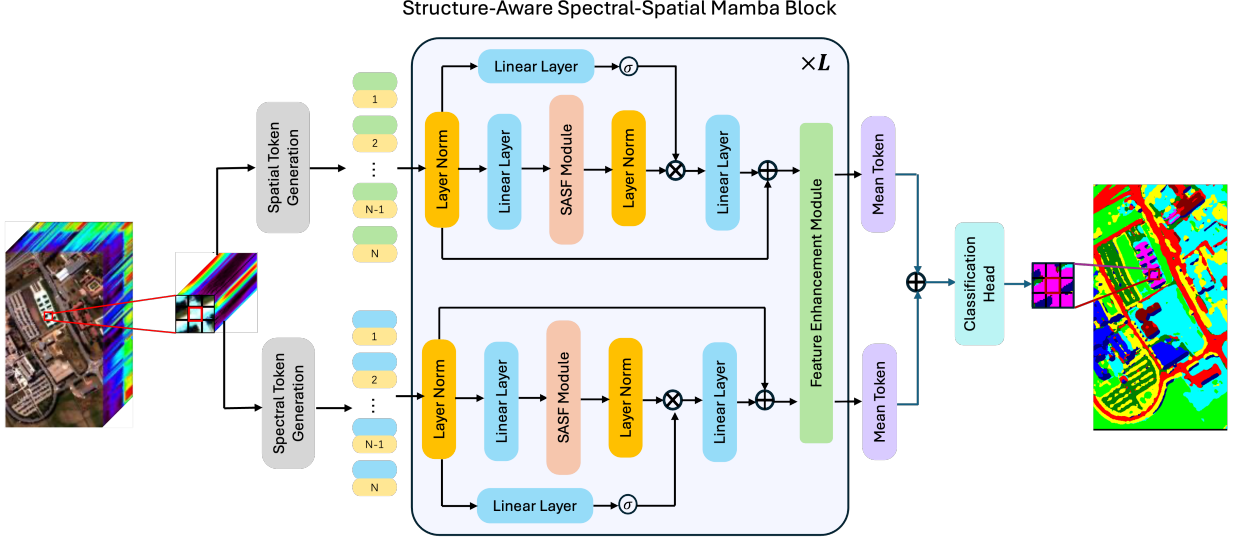


Figure 3.1: Illustration of the proposed model’s architecture.

spatial branch. The spectral feature extraction branch largely mirrors the architecture of the spatial branch. However, it is adapted for 1D spectral data. Since spectral signatures lack the inherent 2D structure of spatial data, the structure-aware state fusion kernel is replaced with a standard 1D convolution. This allows for effective feature fusion along the spectral dimension without imposing inappropriate spatial priors.

Following the parallel processing, the outputs from both branches are fed into a Feature Enhancement Module, which integrates the learned spatial and spectral representations. After passing through the stacked blocks, the resulting feature map is aggregated, for instance by a mean pooling operation on the tokens. Finally, this consolidated feature vector is projected by a classification head, typically implemented as a fully connected layer, to yield the final pixel-wise classification map.

3.2 Spectral-Spatial Tokens Generation

The successful application of sequence-based architectures like Mamba to hyperspectral imagery (HSI) necessitates a preliminary tokenization stage. This process transforms the HSI data cube into a format amenable to sequence modeling, i.e., a series of input tokens. A critical characteristic of HSI data is the profound intertwining of information across both spatial and spectral dimensions. A naive flattening of the data would disrupt these

fundamental structures. Therefore, to preserve and explicitly model these distinct yet complementary characteristics, our methodology employs a dual-stream approach to generate separate sequences of spatial and spectral tokens.

Spatial Token Generation

The primary objective of spatial tokenization is to capture the contextual information and geometric structures within the image. The process begins with the full HSI sample, denoted as $\mathbf{x} \in \mathbb{R}^{H \times W \times B}$, where H and W are the spatial dimensions (height and width), and B is the number of spectral bands. This input cube is first processed by a "Stem" module. It is composed of a sequence of 2D convolutional layers (*Conv2d*), each typically followed by a normalization layer (e.g., *BatchNorm*) and a non-linear activation function (e.g., *GELU*). This initial feature extraction maps the spectral dimension into a high-dimensional feature space, D . The transformation is expressed as:

$$\mathbf{x}_{spa} = Stem(\mathbf{x}) \quad (3.1)$$

where the output feature map $x_{spa} \in \mathbb{R}^{H/S_{spa} \times W/S_{spa} \times D}$. Here, S_{spa} represents the overall spatial downsampling factor of the Stem module. To create the final sequence, this 2D feature map is flattened along its spatial dimensions, yielding the spatial token sequence $X_{spa}^0 \in \mathbb{R}^{N \times D}$, where the sequence length $N = HW/S_{spa}^2$.

Spectral Token Generation

Complementary to spatial tokens, spectral tokens are generated to explicitly model the spectral signatures, which are paramount for material identification. This process operates on a small patch, $x' \in \mathbb{R}^{P \times P \times B}$, centered on the same pixel as the corresponding spatial analysis region. To focus on the spectral profile, the spatial information within this patch is first aggregated. The patch x' is reshaped into a tensor of dimensions $B \times P^2$, effectively treating the flattened $P \times P$ spatial neighborhood as a feature vector for each spectral band. A 1D convolutional layer is then applied along the spectral dimension (B) to map these spatial features into the unified embedding dimension D . This step simultaneously reduces the spectral dimensionality by a factor of S_{spe} , the spectral downsampling rate, which helps in mitigating redundancy between adjacent, often highly correlated, spectral bands. The resulting sequence of spectral tokens is denoted as $X_{spe}^0 \in \mathbb{R}^{M \times D}$, where the sequence length $M = B/S_{spe}$.

To incorporate local context, positional embeddings are integrated into the generated tokens, a technique adapted from the Vision Transformer (ViT) model. Specifically, the

spatial tokens are augmented with 2D sinusoidal positional embeddings to retain spatial locality. In contrast, the spectral tokens are enhanced with 1D positional embeddings to preserve spectral relationships.

3.3 Structure-Aware Spectral-Spatial Mamba Block

The core component of our model is the spectral-spatial Mamba block, which is designed for concurrent feature extraction and fusion. As illustrated in Figure 3.1, this block employs a parallel architecture consisting of two specialized Mamba blocks—one for spatial features and one for spectral features. The outputs from these two streams are then integrated and enhanced by a feature fusion module.

A key innovation in our work is the modification of the standard Mamba design [13]. We replace the original Selective Scan (S6) module with a novel Structure-Aware State Fusion (SASF) module. This substitution enables the state-space model to achieve local neighborhood connectivity, thereby capturing the data’s underlying topology with linear computational complexity [31].

The Structure-Aware State Fusion Module

The process of the Structure-Aware State Fusion (SASF) module can be described by three equations: the state transition equation, the state fusion equation and the observation equation [31], which are formulated for a given token position t as:

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t x_t, \quad (3.2a)$$

$$h'_t = \sum_{k \in \Omega} \alpha_k h_{\rho_k(t)}, \quad (3.2b)$$

$$y_t = C_t h'_t + D x_t, \quad (3.2c)$$

Here, Equation 3.2a is the standard Mamba state transition, where the latent state h_t is updated based on its immediately preceding state h_{t-1} and the current input token x_t . The innovation lies in Equation 3.2b, where the state h_t is not used directly as in the original Selective Scan (S6) module. Instead, a new structure-aware state h'_t is computed. This new state is a weighted sum of the latent states of its neighbors, defined by the neighbor set Ω . The function $\rho_k(t)$ provides the index of the k -th neighbor of position t . This crucial step allows information to flow between spatially or spectrally adjacent tokens, rather than being restricted to the 1D scan direction. Finally, the output y_t is produced by the observation equation (Equation 3.2c) using this enriched state h'_t .

Spatial Mamba Block

The spatial Mamba block is specifically designed to model the 2D spatial context within the HSI patch. Given an input spatial token sequence X_{spa}^{l-1} for block l , the latent state h_{spa}^t for each token is first computed using the state transition equation:

$$h_{spa}^t = \bar{A}_t h_{spa}^{t-1} + \bar{B}_t x_{spa}^t \quad (3.3)$$

Conceptually, the computed states $\{h_{spa}^t\}_{t=1}^N$ are then rearranged from their 1D sequence form back into a 2D grid that reflects their original spatial positions. To perform the state fusion operation on this 2D grid, it employs multi-scale dilated convolutions, as shown in Figure 3.2a. This choice is motivated by their ability to capture contextual information from varying receptive fields efficiently. Specifically, the module utilizes three parallel 3×3 depth-wise convolutional filters with dilation factors $d = 1, 3, 5$. This constructs a multi-scale neighbor set $\Omega_d = \{(i, j) | i, j \in \{d, 0, d\}\}$. The state fusion equation for the spatial domain is thus instantiated as:

$$h_{spa}'^t = \sum_{d=1,3,5} \sum_{i,j \in \Omega_d} k_{ij}^d \cdot h_{spa}^{t+iw+j}, \quad (3.4)$$

where k_{ij}^d is the learnable filter weight for dilation factor d at relative position (i, j) , W is the width of the feature map, and h_{spa}^{t+iw+j} denotes the neighboring state of h_{spa}^t . The resulting fused state $h_{spa}'^t$, which now contains multi-scale spatial context, is used to generate the final output sequence Y_{spa}^l via the observation equation.

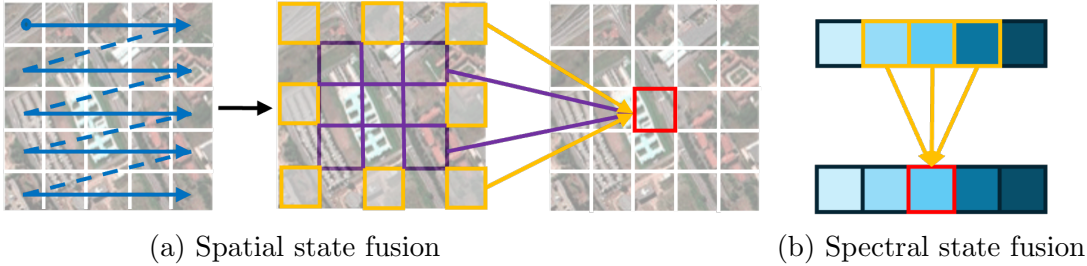


Figure 3.2: Illustration of the structure-aware state fusion mechanism in spatial and spectral block

Spectral Mamba Block

The spectral Mamba block processes the spectral token sequence X_{spe}^{l-1} to capture relationships along the spectral space. The initial state update follows the same transition

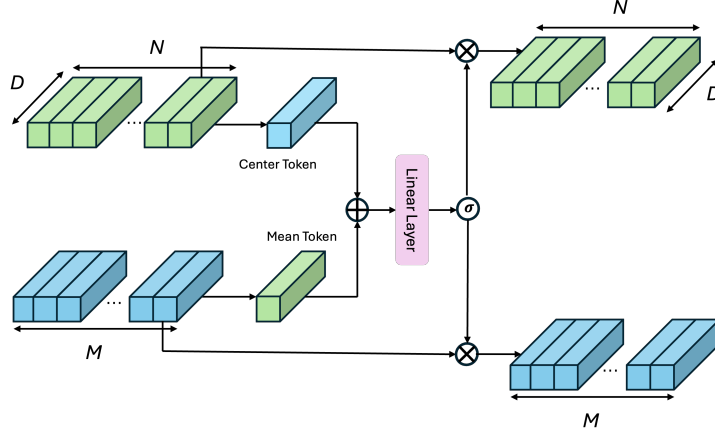


Figure 3.3: Illustration of the Feature Enhancement Module

rule:

$$h_{spe}^t = \bar{A}_t h_{spe}^{t-1} + \bar{B}_t x_{spe}^t \quad (3.5)$$

However, the state fusion mechanism is adapted to the inherent 1D nature of spectral data. Unlike spatial information, which exists on a 2D plane, spectral bands form an ordered 1D sequence. Therefore, I employ a standard 1D convolution for the state fusion step, as shown in Figure 3.2b. The neighbor set is defined along a single axis, $\Omega = \{i | i \in \{k, 0, k\}\}$, where k is half the kernel size. The spectral state fusion is given by:

$$h_{spe}'^t = \sum_{i \in \Omega} \alpha_i \cdot h_{spe}^{t+i}, \quad (3.6)$$

where α_i represents the filter weight in 1D convolution at position i , and h_{spe}^{t+i} denotes the neighbor of the state h_{spe}^t located at the position i . This operation allows each spectral state to integrate information from adjacent spectral bands, effectively modeling local spectral patterns. The output sequence Y_{spe}^l is then generated from this fused state.

Cross-Modal Feature Enhancement Module

After features are extracted by the parallel Mamba blocks, the resulting spatial tokens X_{spa}^l and spectral tokens X_{spe}^l are passed to a Feature Enhancement Module for cross-modal information interaction, as shown in Figure 3.3. This module is designed to recalibrate the features based on the most salient information from both domains [19].

The module first distills a representative feature from each domain. For the spatial domain, we select the central token of the sequence, z_1^l , as the ground truth label of an HSI patch is determined by its center pixel. For the spectral domain, we compute the mean token, z_2^l , which provides a robust, global representation of the entire spectral signature. These two tokens are fused via element-wise averaging:

$$\begin{aligned} \mathbf{z}_1^l &= X_{spa}^l[(N+1)/2, :], \\ \mathbf{z}_2^l &= \frac{1}{M} \sum_{j=1}^M X_{spe}^l[j, :], \\ \mathbf{z}^l &= (\mathbf{z}_2^l + \mathbf{z}_1^l)/2 \end{aligned} \tag{3.7}$$

The fused vector $\mathbf{z}^l$ is passed through a small multi-layer perceptron (MLP) followed by a sigmoid activation function to generate a modulation weight vector $\mathbf{f}^l \in \mathbb{R}^{1 \times D}$. This vector is then broadcasted to match the dimensions of the spatial and spectral token sequences, creating modulation matrices $\mathbf{F}_{spa}^l \in \mathbb{R}^{N \times D}$ and $\mathbf{F}_{spe}^l \in \mathbb{R}^{M \times D}$. This mechanism adaptively re-weights the feature tokens, directing the model’s attention toward the most informative features as guided by both central spatial context and global spectral information. The final enhanced features are computed via element-wise multiplication:

$$\begin{aligned} \mathbf{X}_{spa}^l &= \mathbf{F}_{spa}^l \odot \mathbf{X}_{spa}^l, \\ \mathbf{X}_{spe}^l &= \mathbf{F}_{spe}^l \odot \mathbf{X}_{spe}^l, \end{aligned} \tag{3.8}$$

where $\odot$ means the element-wise multiplication. These enhanced token sequences then serve as the input to the next S3-Mamba block or the final classification head.

3.4 Hybrid Mamba-Transformer Architecture

While the proposed structure-aware Mamba block excels at efficiently modeling local and semi-local dependencies, its capacity for capturing truly global context can be limited. To address this, we explore a synergistic hybrid Mamba-Transformer architecture. This approach is designed to harness the complementary strengths of both Mamba’s linear-complexity state-space models and the Transformer’s powerful global self-attention mechanism. Combining these architectures has emerged as a promising research direction to leverage the efficiency of SSMs for local feature processing and the effectiveness of Transformers for global dependency modeling.

The core of our hybrid design involves strategically replacing the structure-aware state fusion SSM component within our Mamba block with a standard multi-head self-attention

(MHSA) layer. This substitution effectively transforms the block into a Transformer block, capable of dynamically weighing the importance of all other tokens in the sequence to update a given token’s representation, thereby capturing comprehensive long-range dependencies.

Our model architecture consists of a total of four sequential feature extraction blocks. This thesis evaluated four primary configurations to elucidate the trade-off between local feature extraction and global context aggregation:

1. Pure Mamba (4M-0T): The baseline model, utilizing four Mamba blocks. This configuration relies solely on the structure-aware SSM for feature extraction.
2. Mamba-Dominant Hybrid (3M-1T): Three initial Mamba blocks are followed by a final Transformer block. The hypothesis is that the Mamba blocks first efficiently build robust local and regional feature representations, which are then globally integrated by the final attention layer.
3. Balanced Hybrid (2M-2T): An even split, with two Mamba blocks followed by two Transformer blocks, exploring an earlier introduction of global context modeling.
4. Pure Transformer (0M-4T): All four blocks are Transformer blocks. This serves as a benchmark to evaluate a pure self-attention-based approach within our framework.

The empirical results of these experiments, which will be detailed in Chapter 4, revealed that the Mamba-Dominant Hybrid (3M-1T) configuration achieved superior performance. This outcome suggests that our model benefits most from a strong foundation of local and contextual feature extraction provided by the initial Mamba blocks. Subsequently, a single, powerful self-attention layer at a later stage in the network is effective for modeling the global relationships between these refined features.

Chapter 4

Experiments and Discussion

4.1 Datasets

This section details the hyperspectral datasets utilized in this study for the evaluation of our proposed methodologies. We have selected five widely-recognized and challenging datasets: Indian Pines, Pavia University, Houston 2013, WHU-Hi-HanChuan, and WHU-Hi-HongHu. These datasets represent a diverse range of scenes, sensor technologies, and land cover types, providing a robust benchmark for assessing model performance and generalization capabilities.

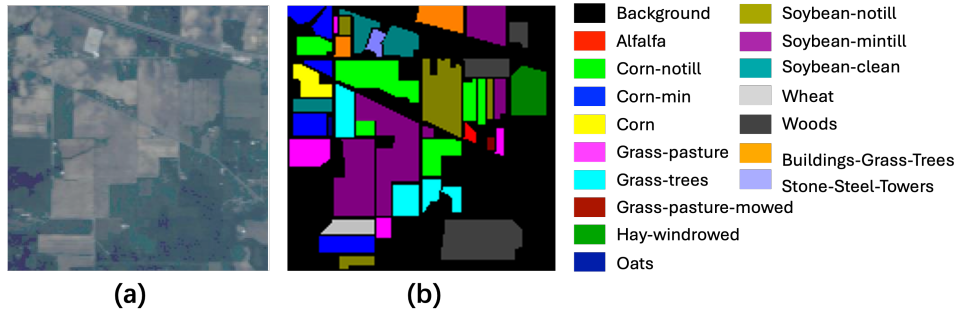


Figure 4.1: Indian Pines dataset. (a) False-color map. (b) Ground-truth map.

Table 4.1: Land cover classes and numbers of samples in the Indian Pines dataset.

No.	Class Name	Training samples	Test samples	Total samples
1	Alfalfa	15	31	46
2	Corn-notill	50	1,378	1,428
3	Corn-mintill	50	780	830
4	Corn	50	187	237
5	Grass-pasture	50	433	483
6	Grass-trees	50	680	730
7	Grass-pasture-mowed	15	13	28
8	Hay-windrowed	50	428	478
9	Oats	15	5	20
10	Soybean-notill	50	922	972
11	Soybean-mintill	50	2,405	2,455
12	Soybean-clean	50	543	593
13	Wheat	50	155	205
14	Woods	50	1,215	1,265
15	Buildings-Grass-Trees	50	336	386
16	Stone-Steel-Towers	50	43	93
Total		695	9,554	10,249

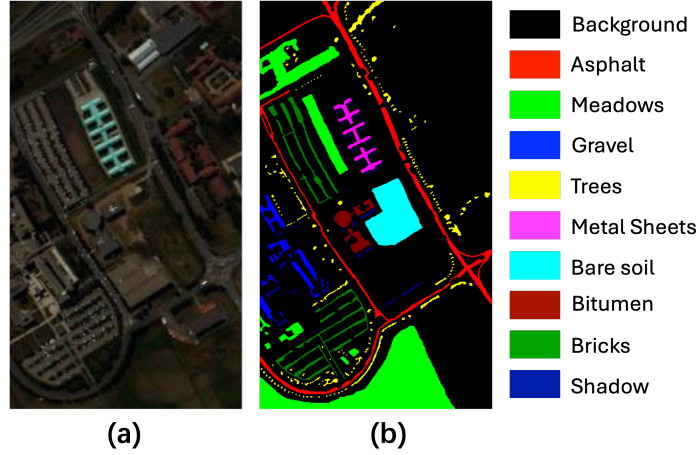


Figure 4.2: Pavia University dataset. (a) False-color map. (b) Ground-truth map.

Indian Pines

The Indian Pines dataset is a seminal benchmark in hyperspectral image analysis. It was acquired in 1992 by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over an agricultural test site in Northwestern Indiana, USA. The scene captures a mixture of agricultural fields, forests, and other perennial vegetation.

The dataset consists of a single hyperspectral image with a spatial dimension of 145×145 pixels and a spatial resolution of approximately 20 meters. It comprises 224 contiguous spectral bands covering the wavelength range from 400 to 2500 nanometers. For standard analysis, the number of bands is typically reduced to 200 by removing 24 water absorption bands (bands 104-108, 150-163, and 220). The ground truth data designates 16 distinct land cover classes, representing various crops such as corn and soybeans at different stages of growth, as well as other features like woods and man-made structures. The dataset's false color and ground-truth maps are illustrated in Figure 4.1, and the distribution of training and test samples is detailed in Table 4.1.

Pavia University

The Pavia University dataset was collected by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over the city of Pavia, Italy. This dataset is a popular choice for urban land cover classification tasks.

The image has a spatial size of 610×340 pixels and a geometric resolution of 1.3 meters.

Table 4.2: Land cover classes and numbers of samples in the Pavia University dataset.

No.	Class Name	Training samples	Test samples	Total samples
1	Asphalt	50	6,581	6,631
2	Meadows	50	18,599	18,649
3	Gravel	50	2,049	2,099
4	Trees	50	3,014	3,064
5	Mental sheets	50	1,295	1,345
6	Bare soil	50	4,979	5,029
7	Bitumen	50	1,280	1,330
8	Bricks	50	3,632	3,682
9	Shadow	50	897	947
Total		450	42,326	42,776

It originally contained 115 spectral bands. After removing 12 noisy and water absorption bands, 103 spectral bands remain for analysis, spanning the wavelength range from 430 to 860 nanometers. The dataset is annotated with nine urban land cover classes, including asphalt, meadows, gravel, trees, painted metal sheets, bare soil, bitumen, self-blocking bricks, and shadows. The dataset’s false color and ground-truth maps are illustrated in Figure 4.2, and the distribution of training and test samples is detailed in Table 4.2.

Houston 2013

The Houston 2013 dataset was released as part of the 2013 IEEE Geoscience and Remote Sensing Society (GRSS) Data Fusion Contest [8]. It was acquired by the ITRES CASI-1500 sensor over the University of Houston campus and its surrounding urban area. This dataset is notable for its high spatial resolution and the complexity of its urban environment.

The hyperspectral image has a spatial dimension of 349×1905 pixels with a spatial resolution of 2.5 meters. It contains 144 spectral bands ranging from 380 to 1050 nanometers. The ground truth encompasses 15 distinct classes of urban and natural land cover, providing a challenging scenario for classification algorithms. The dataset’s false color and ground-truth maps are illustrated in Figure 4.3, and the distribution of training and test samples is detailed in Table 4.3.

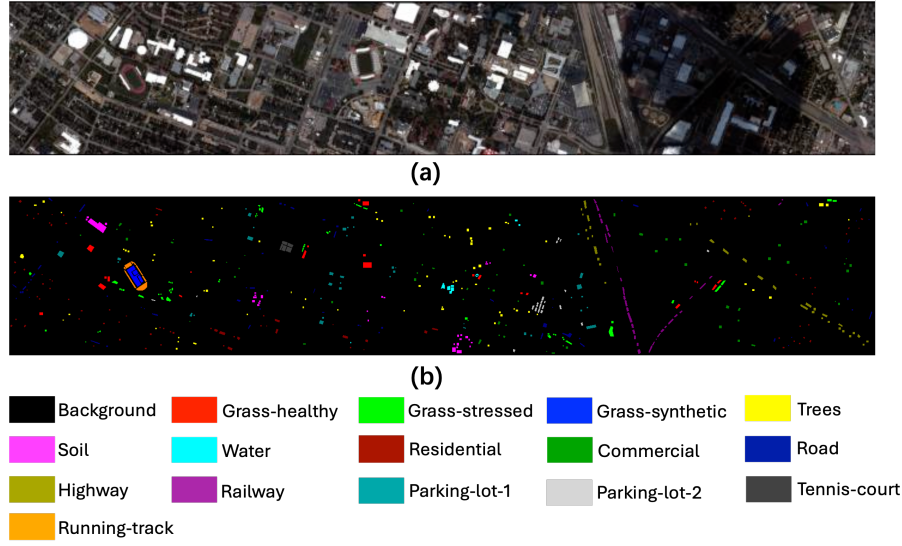


Figure 4.3: Houston 2013 dataset. (a) False-color map. (b) Ground-truth map.

Table 4.3: Land cover classes and numbers of samples in the Houston 2013 dataset.

No.	Class Name	Training samples	Test samples	Total samples
1	Grass-healthy	50	1,201	1,251
2	Grass-stressed	50	1,204	1,254
3	Grass-synthetic	50	647	697
4	Tree	50	1,194	1,244
5	Soil	50	1,192	1,242
6	Water	50	275	325
7	Residential	50	1,218	1,268
8	Commercial	50	1,194	1,244
9	Road	50	1,202	1,252
10	Highway	50	1,177	1,227
11	Railway	50	1,185	1,235
12	Parking-lot-1	50	1,183	1,233
13	Parking-lot-2	50	419	469
14	Tennis-court	50	378	428
15	Running-track	50	610	660
Total		750	14,279	15,029

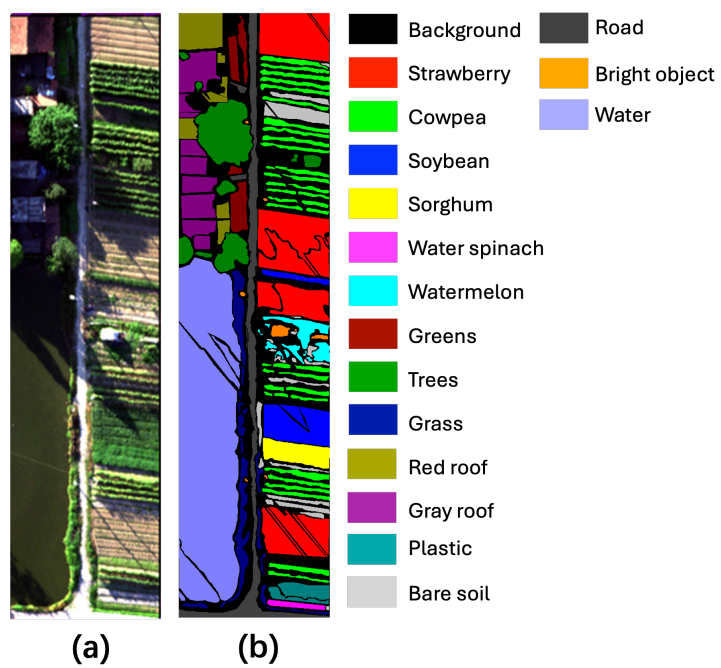


Figure 4.4: WHU-Hi-HanChuan dataset. (a) False-color map. (b) Ground-truth map.

Table 4.4: Land cover classes and numbers of samples in the WHU-Hi-HanChuan dataset.

No.	Class Name	Training samples	Test samples	Total samples
1	Strawberry	50	44,685	44,730
2	Cowpea	50	22,703	22,753
3	Soybean	50	10,237	10,287
4	Sorghum	50	5,303	5,353
5	Water spinach	50	1,150	1,200
6	Watermelon	50	4,483	4,533
7	Greens	50	5,853	5,903
8	Trees	50	17,928	17,978
9	Grass	50	9,419	9,469
10	Red roof	50	10,466	10,516
11	Gray roof	50	16,861	16,911
12	Plastic	50	3,629	3,679
13	Bare soil	50	9,066	9,116
14	Road	50	18,510	18,560
15	Bright object	50	1,086	1,136
16	Water	50	75,351	75,401
Total		800	256,730	257,530

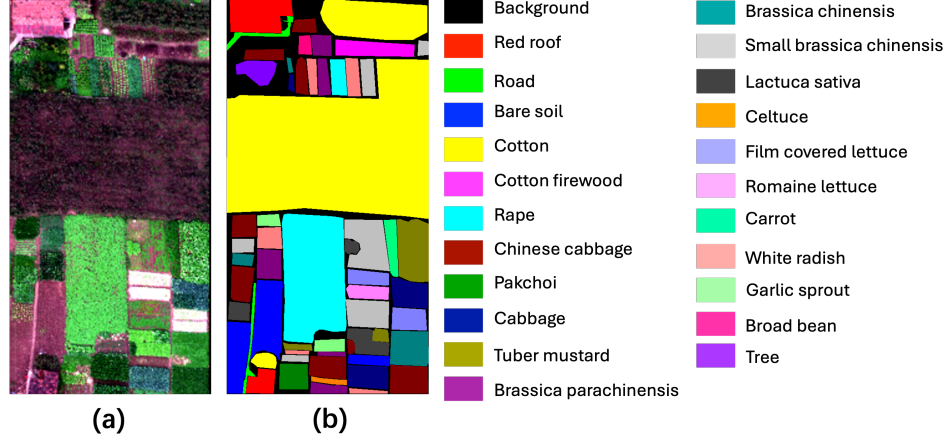


Figure 4.5: WHU-Hi-HongHu dataset. (a) False-color map. (b) Ground-truth map.

WHU-Hi-HanChuan

The WHU-Hi-HanChuan dataset is part of the Wuhan UAV-borne Hyperspectral Image (WHU-Hi) collection, captured by a Headwall Nano-Hyperspec imaging sensor mounted on a UAV platform [36]. The data was acquired over a rural area in Hanchuan, Hubei province, China, in June 2016.

This dataset features a high spatial resolution of approximately 0.109 meters. The image size is 1217×303 pixels. It comprises 274 spectral bands covering the 400 to 1000 nanometer range. The scene is characterized by various crop types and includes 16 labeled land cover classes, making it suitable for fine-grained agricultural classification studies. The dataset’s false color and ground truth maps are illustrated in Figure 4.4, and the distribution of training and test samples is detailed in Table 4.4.

WHU-Hi-HongHu

Also a part of the WHU-Hi collection [36], the WHU-Hi-HongHu dataset was acquired in November 2017 over another agricultural area in Honghu City, Hubei province, China, using the same Headwall Nano-Hyperspec sensor system.

This dataset provides an even higher spatial resolution of about 0.043 meters, with an image size of 940×475 pixels. It consists of 270 spectral bands in the 400 to 1000 nanometer wavelength range. The ground truth for this dataset is particularly detailed, with 22 distinct classes that include various crop types and other typical rural features. The

Table 4.5: Land cover classes and numbers of samples in the WHU-Hi-HongHu dataset.

No.	Class Name	Training samples	Test samples	Total samples
1	Red roof	50	13,991	14,041
2	Road	50	3,462	3,512
3	Bare soil	50	21,771	21,821
4	Cotton	50	163,235	163,285
5	Cotton firewood	50	6,168	6,218
6	Rape	50	44,507	44,557
7	Chinese cabbage	50	24,053	24,103
8	Pakchoi	50	4,067	4,117
9	Cabbage	50	10,769	10,819
10	Tuber mustard	50	12,344	12,394
11	Brassica-p.	50	10,965	11,015
12	Brassica-c.	50	8,904	8,954
13	Small brassica-c.	50	22,457	22,507
14	Lactuca sativa	50	7,306	7,356
15	Celtuce	50	952	1,002
16	Film covered lettuce	50	7,212	7,262
17	Romaine lettuce	50	2,960	3,010
18	Carrot	50	3,167	3,217
19	White radish	50	8,662	8,712
20	Gartic sprout	50	3,436	3,486
21	Broad bean	50	1,278	1,328
22	Tree	50	3,990	4,040
Total		1,100	385,593	386,693

very high spatial resolution of this dataset presents unique challenges and opportunities for hyperspectral image analysis. The dataset’s false color and ground-truth maps are illustrated in Figure 4.5, and the distribution of training and test samples is detailed in Table 4.4.

4.2 Comparison of Methods

4.2.1 Experimental setup

In our experimental evaluation, we selected a series of representative deep learning models for comparison, covering a variety of architectures from Convolutional Neural Networks (CNNs) to the latest Transformers and State Space Models (SSMs). The following is a brief introduction to each benchmark model.

1)2D-CNN[34]: This model primarily utilizes two-dimensional convolutional kernels to extract spatial features from each spectral band independently. To integrate spectral information, it typically requires a pre-processing step, such as Principal Component Analysis (PCA), to reduce the dimensionality of the hyperspectral data.

2)3D-CNN[4]: To leverage both spatial and spectral dimensions simultaneously, the 3D-CNN employs three-dimensional convolutional kernels to directly process hyperspectral image cubes.

3)COLON[32]: COLON is a lightweight CNN architecture designed for hyperspectral image classification. Through a specific network structure and weight-sharing strategies, it aims to achieve efficient feature extraction and classification performance at a lower computational cost.

4)ViT[9]: It operates by partitioning an input image into a series of patches, which are then linearly embedded and fed into a standard Transformer encoder.

5)SpectralFormer[17]: This model is a Transformer architecture specifically tailored for hyperspectral image classification. It employs a well-designed Groupwise Spectral Embedding strategy to learn local relationships between adjacent spectral bands.

6)morphFormer[27]: The morphFormer enhances feature extraction by integrating morphological operations with the Transformer architecture. It utilizes learnable morphological operators (such as erosion and dilation) to capture the structural and textural information of the image.

7)SSFTT[29]: The SSFTT model captures spectral and spatial features through a novel feature tokenization mechanism. It first extracts features in the spectral and spatial

domains separately. These features are then "tokenized" and fed into a joint Transformer encoder, thereby achieving a deep fusion and modeling of spatio-spectral information.

8)GSCViT[35]: GSCViT is an improved Vision Transformer model that introduces a group-wise spatially-constrained mechanism. By restricting the receptive field in the self-attention calculation, the model is encouraged to focus on spatial relationships within local regions.

9)MambaHSI[22]: MambaHSI represents an application of the emerging State Space Model (SSM), Mamba, to hyperspectral image classification. It utilizes the Mamba architecture as its core building block, leveraging its efficient sequence modeling capability to capture long-range dependencies along the spectral dimension.

10)MIM[38]: This model adopts a nested or hierarchical Mamba structure. It first employs an "inner" Mamba network to model the spectral sequence for each pixel. Subsequently, an "outer" Mamba network models the spatial sequence of the processed features, thereby achieving a deep coupling of spectral and spatial dependencies.

11)SSMamba[19]: Also based on the State Space Model, SSMamba is designed to efficiently and simultaneously process both spectral and spatial information. The model first converts HSI cubes into separate spatial and spectral token sequences. The Mamba model is then used to process these sequences, capturing complex joint spatio-spectral dependencies within a unified framework.

All experiments were conducted under a unified set of hyperparameters. We defined the input as a 27×27 spatial patch extracted for each pixel in the hyperspectral image. Within the model architecture, the spatial and spectral downsampling rates were configured to $S_{spa} = 3$ and $S_{spe} = 2$, respectively. The network comprises $L = 4$ stacked blocks, each operating on a feature space with an embedding dimension of $D = 64$. The training process was carried out for 200 epochs, using a batch size of 64.

4.2.2 Evaluation Metrics

To quantitatively assess the performance of the classification models in this study, three standard and widely adopted metrics were employed: Overall Accuracy (OA), Average Accuracy (AA), and the Kappa Coefficient (K). These metrics provide a comprehensive view of a model's performance, capturing not only the general correctness but also its effectiveness across different classes and its agreement beyond chance[11].

Overall Accuracy (OA)

Overall Accuracy is the most straightforward evaluation metric. It represents the proportion of correctly classified pixels out of the total number of pixels in the test set. It provides a general measure of the model’s correctness. OA is calculated as:

$$OA = \frac{\text{Number of Correctly Classified Pixels}}{\text{Total Number of Test Pixels}} \quad (4.1)$$

While useful, OA can be misleading, especially when the dataset suffers from a significant class imbalance, as it tends to be biased towards the majority classes.

Average Accuracy (AA)

To address the limitations of OA in the context of imbalanced datasets, Average Accuracy is used. AA is the mean of the classification accuracies for each individual class. It is calculated by taking the accuracy of each class separately and then averaging these accuracies. This ensures that every class, regardless of its size, contributes equally to the final metric. The formula is:

$$AA = \frac{1}{C} \sum_{i=1}^C \frac{\text{Number of Correctly Classified Pixels in Class } i}{\text{Total Number of Test Pixels in Class } i} \quad (4.2)$$

where C is the total number of classes.

Kappa Coefficient (K)

The Kappa Coefficient is a statistical measure that evaluates the level of agreement between the predicted classification map and the ground truth map, while accounting for the agreement that could be expected to occur by chance[6]. It provides a more robust measure of performance than simple accuracy, especially in the presence of class imbalance[11]. A Kappa value of 1 represents perfect agreement, 0 indicates that the agreement is equivalent to chance, and negative values suggest the agreement is worse than chance. The formula for the Kappa Coefficient is:

$$K = \frac{P_o - P_e}{1 - P_e} \quad (4.3)$$

where P_o is the observed accuracy (equivalent to OA), and P_e is the hypothetical probability of chance agreement.

4.2.3 Classification Results

The classification results of different methods on the five datasets are shown in Tables 4.1 to 4.5. Additionally, the classification maps for different methods on the five datasets are shown in Figures 4.6 to 4.10.

As demonstrated in Tables 4.1 through 4.5, our proposed method (SMaT-HSI) delivers state-of-the-art performance, achieving the highest OA, AA, and κ on four out of the five benchmark datasets. This underscores the superior feature extraction and classification capabilities of our approach. our method achieves an outstanding OA of 97.81%, an AA of 98.83%, and a κ of 0.9748. This represents a significant improvement over all other competing methods, including the strongest baseline, SSMamba, which scored 96.06% OA. Notably, our model demonstrates its robustness by achieving perfect 100.00% accuracy on four distinct classes (Classes 1, 7, 9, and 14). On the Pavia University dataset, which contains a higher spatial resolution, our model continues to demonstrate its superiority. Table 4.2 shows that our method attains the highest scores across all metrics with an OA of 99.72%, an AA of 99.73%, and a κ of 0.9963. It surpasses the second-best model, SSFTT, by 0.83% in OA. The per-class accuracy results are also exceptional, with our model achieving 100.00% accuracy for four of the nine classes (Classes 4, 5, 7, and 8), confirming its effectiveness on this dataset. The classification results for the Houston 2013 dataset are presented in Table 4.3. On this dataset, the performance among the top models is highly competitive. While the SSFTT model achieves the highest overall metrics (98.17% OA), our model performs exceptionally well, securing a very close OA of 97.99%, an AA of 97.91%, and a κ of 0.9783. Importantly, our model achieves the highest accuracy on the largest number of individual classes (7 out of 15), including Classes 1, 2, 3, 4, 10, 11, and 14. This indicates that our model provides a more consistent and balanced classification performance across a wider variety of spectral signatures present in the dataset.

We further validated our model on the two large-scale WHU-Hi datasets, HanChuan and HongHu. For the WHU-Hi-HanChuan dataset (Table 4.4), our method establishes a new state-of-the-art with an OA of 96.21%, an AA of 96.55%, and a κ of 0.9558. It outperforms all baseline models and achieves the best classification accuracy on numerous individual classes, showcasing its powerful discriminative ability. Similarly, on the more complex WHU-Hi-HongHu dataset with 22 classes (Table 4.5), our model demonstrates its strong generalization capability. It achieves the top performance with an OA of 96.83%, an AA of 96.52%, and a κ of 0.9562, surpassing the specialized MambaHSI model.

In summary, the comprehensive quantitative analysis across these five diverse datasets validates the superior performance and robustness of our proposed method. It consistently outperforms a wide range of advanced models, marking a significant advancement

in hyperspectral image classification.

Table 4.6: Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the Indian Pines dataset. The best is in bold.

Class No.	CNNs			Transformers					Mamba			SMaT-HSI
	2D CNN	3D CNN	CLOLN	ViT	SpectralFormer	morphFormer	SSFTT	GSCViT	MambaHSI	MIM	SSMamba	
1	93.94	92.30	92.31	88.57	70.54	95.39	100.00	95.39	98.06	90.29	99.84	100.00
2	63.19	59.23	71.31	74.10	65.57	90.75	95.63	84.37	92.99	90.27	93.42	96.79
3	70.25	58.48	66.17	73.86	86.35	93.97	94.75	91.11	93.88	97.08	94.03	95.83
4	81.94	64.92	78.41	77.12	90.57	97.40	95.65	81.94	99.20	93.54	94.10	92.11
5	74.90	84.55	94.03	90.50	96.38	98.60	96.67	97.03	96.58	98.72	97.96	97.36
6	91.05	91.87	90.23	91.65	99.73	98.69	99.05	99.34	98.19	100.00	99.30	98.91
7	37.68	57.78	52.00	89.66	77.60	100.00	83.87	92.86	99.23	91.24	88.34	89.65
8	96.63	98.13	94.38	99.53	78.04	99.88	100.00	100.00	99.91	93.11	97.14	100.00
9	76.92	15.87	58.82	41.67	72.70	100.00	100.00	45.46	100.00	88.97	94.29	100.00
10	71.91	63.06	14.33	79.11	99.70	93.64	94.14	90.10	95.54	100.00	92.90	97.51
11	71.87	63.60	81.70	82.02	97.76	92.67	95.08	89.71	94.15	98.40	95.78	98.69
12	62.13	49.47	88.53	63.38	74.24	95.06	90.51	79.10	96.43	81.20	97.01	94.07
13	97.78	93.90	98.10	97.18	92.78	97.18	95.39	99.68	100.00	84.91	99.87	98.41
14	88.59	92.46	98.29	92.67	97.50	99.47	98.70	98.16	99.10	97.50	99.41	99.75
15	78.80	81.11	83.51	78.66	71.67	97.02	94.84	91.34	99.73	75.86	97.48	99.11
16	85.71	74.55	86.87	88.66	35.71	92.47	95.56	94.38	99.07	76.92	96.57	97.72
OA (%)	75.27	70.65	79.50	81.84	79.96	94.98	95.79	90.68	95.95	92.08	96.06	97.81
AA (%)	86.25	83.63	88.04	90.10	82.27	97.76	97.76	95.02	97.63	94.11	98.07	98.83
Kappa	0.7195	0.6685	0.7647	0.7938	0.7733	0.9424	0.9517	0.8934	0.9535	0.9096	0.9548	0.9748

Table 4.7: Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the Pavia University dataset. The best is in bold.

Class No.	CNNs			Transformers					Mamba			SMaT-HSI
	2D CNN	3D CNN	CLOLN	ViT	SpectralFormer	morphFormer	SSFTT	GSCViT	MambaHSI	MIM	SSMamba	
1	89.53	89.36	91.90	94.49	90.16	98.87	98.63	97.01	96.99	96.34	98.22	99.65
2	94.56	92.15	96.78	92.91	91.51	98.77	99.56	97.59	98.56	92.25	98.85	99.87
3	75.41	74.07	77.69	90.77	85.95	97.43	97.85	91.45	97.30	86.98	99.39	99.44
4	96.87	92.38	96.58	94.50	96.08	97.59	98.62	97.27	92.14	94.21	96.71	99.04
5	99.76	95.67	99.80	99.96	99.96	99.61	99.53	99.96	100.00	99.12	99.92	100.00
6	80.73	72.68	86.67	76.27	75.14	95.95	98.53	93.69	99.48	81.76	98.31	99.99
7	76.50	74.49	92.42	97.04	74.86	99.68	98.38	97.94	98.73	97.99	99.92	100.00
8	79.38	80.01	84.09	90.31	90.46	97.24	97.35	94.13	96.64	90.28	97.67	99.18
9	97.14	94.77	98.43	97.85	98.88	98.46	98.23	99.27	99.34	99.13	99.88	99.94
OA (%)	89.53	86.95	92.79	91.22	88.86	98.21	98.89	96.34	97.81	90.56	98.54	99.72
AA (%)	89.27	87.23	92.89	93.83	91.34	98.42	98.78	96.83	97.69	91.87	99.34	99.73
Kappa	0.8624	0.8284	0.9034	0.8849	0.8544	0.9762	0.9852	0.9516	0.9709	0.8912	0.9807	0.9963

4.3 Discussion

4.3.1 Ablation Studies

This study conducted a series of ablation studies to meticulously analyze the contribution of each key component in the proposed model and to investigate the optimal architectural

Figure 4.6: (a) Ground Truth, (b) 2D CNN, (c) 3D CNN, (d) CLOLN, (e) ViT, (f) SpectralFormer, (g) morphFormer, (h) SSFTT, (i) GSCViT, (j) MambaHSI, (k) MiM, (l) SSMamba, (m) SMaT-HSI (Ours).

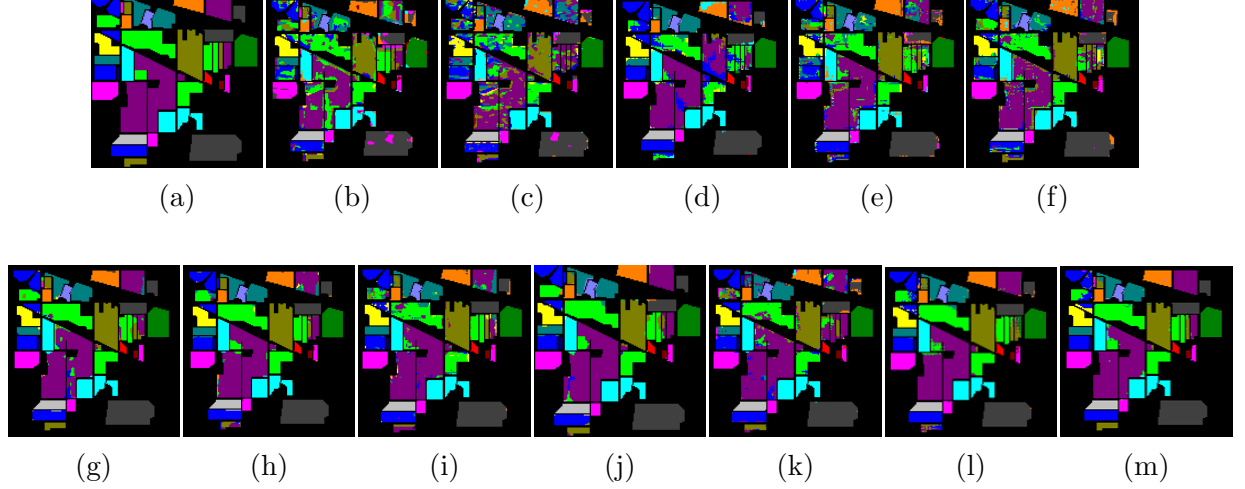


Table 4.8: Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the Houston 2013 dataset. The best is in bold.

Class No.	CNNs			Transformers					Mamba			SMaT-HSI
	2D CNN	3D CNN	CLOLN	ViT	SpectralFormer	morphFormer	SSFTT	GSCViT	MambaHSI	MiM	SSMamba	
1	93.86	95.28	95.17	89.44	97.22	97.41	97.52	96.94	97.57	93.12	99.12	99.46
2	85.92	96.23	96.22	97.69	97.99	98.86	98.18	98.03	97.72	96.45	98.92	99.33
3	98.83	94.52	99.30	99.84	95.89	99.76	99.61	99.23	99.75	99.32	100.00	100.00
4	98.89	97.25	96.63	90.40	96.30	98.27	99.16	97.94	98.66	95.41	99.83	99.87
5	90.87	96.34	98.46	97.42	98.90	98.47	100.00	99.58	99.91	98.85	95.58	95.89
6	87.86	89.79	100.00	97.51	88.78	98.39	99.27	96.02	99.13	97.97	98.71	97.99
7	73.99	89.67	95.47	94.30	89.57	96.83	97.38	94.84	95.76	90.11	97.32	97.30
8	79.40	85.89	90.96	89.19	88.80	93.91	97.37	92.10	90.56	86.02	96.26	93.69
9	65.11	86.04	91.40	86.38	87.90	93.98	97.90	88.98	94.15	83.17	96.32	97.53
10	70.50	85.55	96.34	87.82	90.90	98.52	97.76	91.71	97.64	91.20	98.32	98.61
11	70.15	84.46	99.40	89.84	86.88	98.18	97.97	94.71	97.72	88.95	99.49	99.37
12	64.23	77.89	88.47	84.57	83.72	93.11	98.29	89.54	94.99	87.19	94.47	94.22
13	43.46	88.14	98.34	88.06	73.63	95.60	92.32	94.50	98.50	91.89	95.70	93.62
14	88.44	96.77	100.00	100.00	95.17	100.00	100.00	100.00	100.00	96.78	100.00	99.47
15	99.16	97.90	97.13	97.13	97.70	97.44	99.59	97.41	99.84	97.33	97.67	97.99
OA (%)	79.63	90.31	95.48	91.72	91.74	97.01	98.17	94.97	96.96	91.72	97.72	97.63
AA (%)	80.70	91.76	96.33	93.10	91.95	97.54	98.48	95.72	97.46	93.12	97.75	97.91
Kappa	77.97	0.8952	0.9511	0.9105	0.9106	0.9676	0.9802	0.9456	0.9671	0.9118	0.9753	0.9743

Figure 4.7: (a) Ground Truth, (b) 2D CNN, (c) 3D CNN, (d) CLOLN, (e) ViT, (f) SpectralFormer, (g) morphFormer, (h) SSFTT, (i) GSCViT, (j) MambaHSI, (k) MiM, (l) SSMamba, (m) SMaT-HSI (Ours).

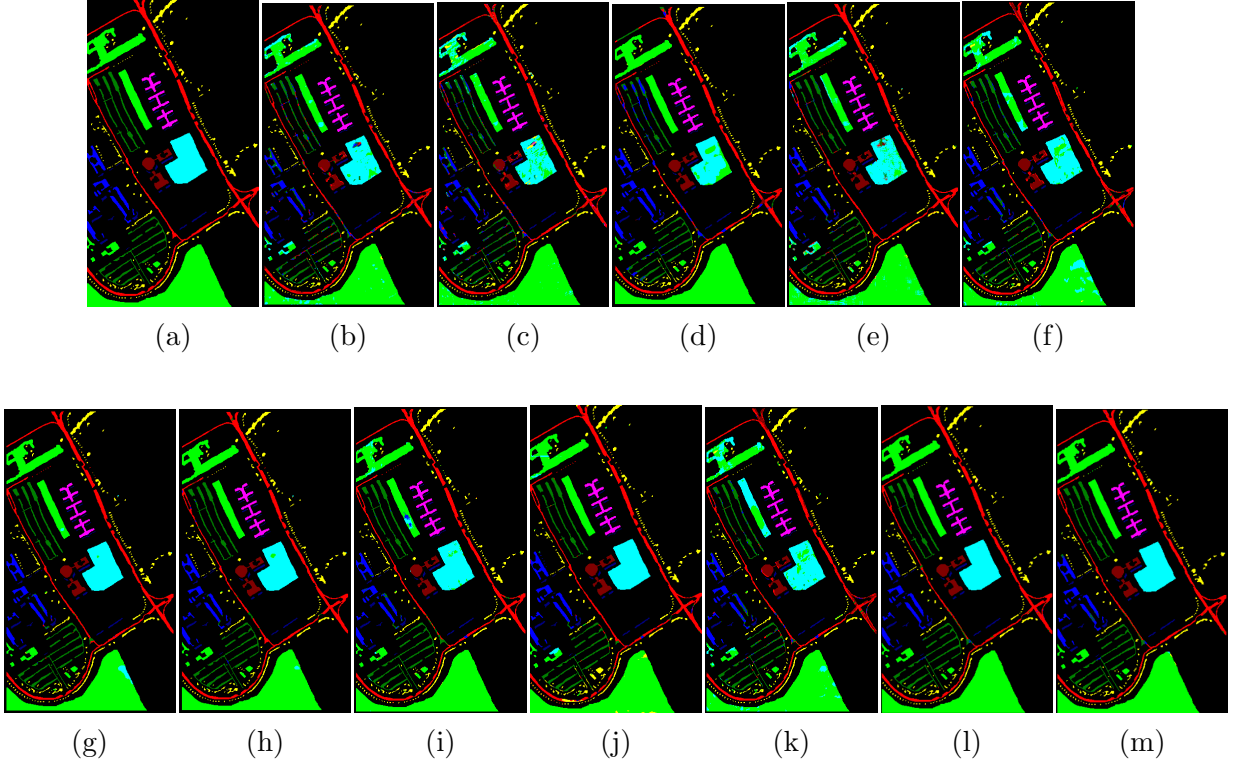


Figure 4.8: (a) Ground Truth, (b) 2D CNN, (c) 3D CNN, (d) CLOLN, (e) ViT, (f) SpectralFormer, (g) morphFormer, (h) SSFTT, (i) GSCViT, (j) MambaHSI, (k) MiM, (l) SSMamba, (m) SMaT-HSI (Ours).

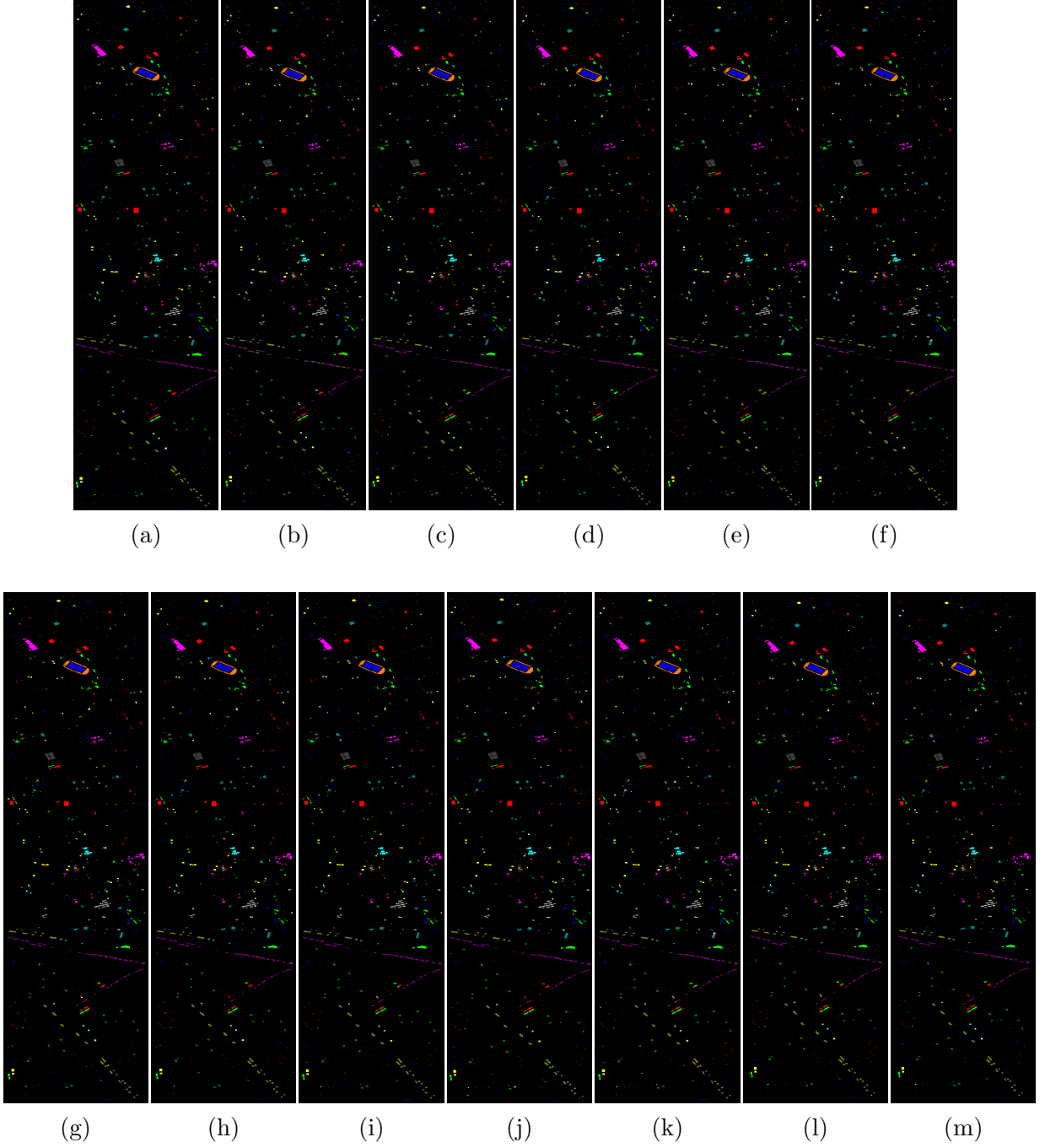


Table 4.9: Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the WHU-Hi-HanChuan dataset. The best is in bold.

Class No.	CNNs			Transformers					Mamba			SMaT-HSI
	2D CNN	3D CNN	CLOLN	ViT	SpectralFormer	morphFormer	SSFTT	GSCViT	MambaHSI	MIM	SSMamba	
1	90.52	77.83	88.62	90.99	91.37	95.72	95.72	89.54	95.07	89.72	96.58	98.48
2	81.98	82.54	84.67	82.21	76.84	93.92	91.25	80.26	89.69	80.37	95.37	96.59
3	76.83	72.04	85.03	75.60	78.29	92.23	96.94	81.14	97.63	94.87	98.60	97.61
4	90.20	86.85	88.96	94.89	94.85	91.56	98.82	96.81	98.67	98.45	96.36	99.40
5	62.47	47.63	84.15	53.29	38.25	94.88	81.26	76.98	99.98	98.96	92.48	97.58
6	38.46	33.96	53.72	47.54	41.38	77.16	70.97	33.71	85.45	76.65	78.50	83.00
7	82.14	83.65	78.91	88.62	78.97	89.04	82.96	85.00	97.41	92.61	92.91	92.21
8	76.83	69.39	82.40	71.88	71.58	90.81	89.10	77.34	83.16	70.77	91.31	90.40
9	69.81	65.22	79.80	75.98	63.84	89.95	84.89	73.90	92.67	89.12	92.48	93.13
10	85.50	78.37	93.57	97.06	88.53	95.61	96.84	94.87	96.17	94.44	97.31	94.73
11	90.36	91.00	77.97	93.81	89.28	94.18	93.85	90.09	93.59	89.56	94.36	93.42
12	72.24	55.71	69.72	64.07	59.08	88.84	90.98	60.09	95.00	88.81	97.71	95.88
13	68.21	54.35	63.91	69.96	61.35	76.92	81.08	70.82	84.57	78.69	88.00	88.41
14	89.46	83.15	72.59	86.46	79.70	94.05	93.13	87.60	94.72	89.68	95.30	96.38
15	82.96	81.97	91.78	72.75	60.08	93.11	92.47	88.64	95.23	94.42	96.57	94.56
16	99.22	99.10	96.60	99.11	98.75	99.23	99.23	98.83	99.03	98.28	99.58	99.10
OA (%)	86.78	81.81	85.44	87.49	84.49	94.25	93.81	87.06	94.47	90.23	95.94	96.21
AA (%)	84.38	79.90	85.33	85.75	81.21	93.84	93.07	84.72	93.63	89.27	95.73	96.55
Kappa	0.8462	0.7895	0.8313	0.8543	0.8198	0.9330	0.9277	0.8496	0.9354	0.8917	0.9525	0.9558

Table 4.10: Quantitative performance of different methods in terms of OA, AA, and Kappa, as well as the accuracies for each class on the WHU-Hi-HongHu dataset. The best is in bold.

Class No.	CNNs			Transformers					Mamba			SMaT-HSI
	2D CNN	3D CNN	CLOLN	ViT	SpectralFormer	morphFormer	SSFTT	GSCViT	MambaHSI	MIM	SSMamba	
1	96.44	92.48	96.84	97.00	95.27	97.41	96.71	95.51	97.43	92.34	98.59	98.29
2	84.92	74.98	90.82	78.40	74.26	93.28	81.46	84.23	97.08	83.23	88.79	95.18
3	88.90	88.57	92.28	89.10	87.17	96.49	88.17	87.63	95.77	89.77	97.49	96.44
4	96.33	93.41	95.38	96.81	97.11	98.67	97.49	98.68	98.45	95.99	98.48	99.00
5	65.96	55.98	45.80	69.60	75.02	81.51	72.32	78.41	97.59	59.97	81.46	85.54
6	87.11	90.18	94.82	86.54	93.60	97.38	96.63	94.30	96.32	97.23	96.50	98.02
7	69.30	63.93	64.28	73.10	81.95	91.46	90.60	71.16	89.82	84.92	92.23	91.32
8	27.04	33.38	33.64	47.28	40.56	85.60	76.43	52.26	98.76	67.66	93.07	89.73
9	97.80	90.78	86.64	97.45	96.45	98.83	97.89	96.92	96.16	97.45	98.88	99.23
10	72.83	71.51	80.37	76.30	70.21	94.01	93.40	79.76	93.04	89.14	94.27	96.01
11	62.81	46.82	78.18	72.12	68.50	89.27	89.35	62.38	94.25	90.34	94.09	93.79
12	67.44	61.48	56.34	71.92	65.50	84.09	82.83	75.24	91.52	80.64	91.49	89.81
13	61.56	69.22	81.78	73.17	68.31	90.20	84.57	72.95	92.38	78.61	87.28	92.38
14	84.00	76.02	84.17	82.10	82.53	97.48	93.48	87.30	95.16	90.19	93.58	94.75
15	64.40	61.21	84.95	87.73	70.47	92.86	86.45	71.14	99.66	88.63	94.43	90.69
16	90.48	88.65	96.87	96.72	91.14	98.79	97.28	96.53	97.14	96.94	96.90	99.56
17	81.73	75.98	93.31	92.52	77.20	96.23	97.46	88.98	99.20	93.24	95.17	99.02
18	61.72	56.87	74.86	73.15	76.39	89.43	89.23	77.31	97.40	87.87	92.11	89.19
19	75.48	75.15	88.03	79.49	81.75	89.51	94.66	86.75	92.55	91.87	90.34	91.68
20	80.15	49.42	90.98	80.66	80.01	87.96	84.91	86.51	99.31	88.78	93.98	94.83
21	43.52	44.07	50.07	48.30	41.61	83.56	54.56	37.12	99.95	74.98	93.77	89.27
22	51.33	31.07	89.49	51.10	60.52	80.51	62.14	69.28	99.91	85.05	79.23	91.58
OA (%)	84.13	81.05	86.32	87.27	87.19	95.40	92.92	89.18	96.39	91.68	95.58	96.52
AA (%)	81.85	76.76	86.88	86.15	84.26	95.39	93.52	85.38	96.31	93.42	96.55	96.83
Kappa	0.8028	0.7666	0.8306	0.8412	0.8401	0.9422	0.9113	0.8637	0.9545	0.8995	0.9444	0.9562

Figure 4.9: (a) Ground Truth, (b) 2D CNN, (c) 3D CNN, (d) CLOLN, (e) ViT, (f) SpectralFormer, (g) morphFormer, (h) SSFTT, (i) GSCViT, (j) MambaHSI, (k) MiM, (l) SSMamba, (m) SMaT-HSI (Ours).

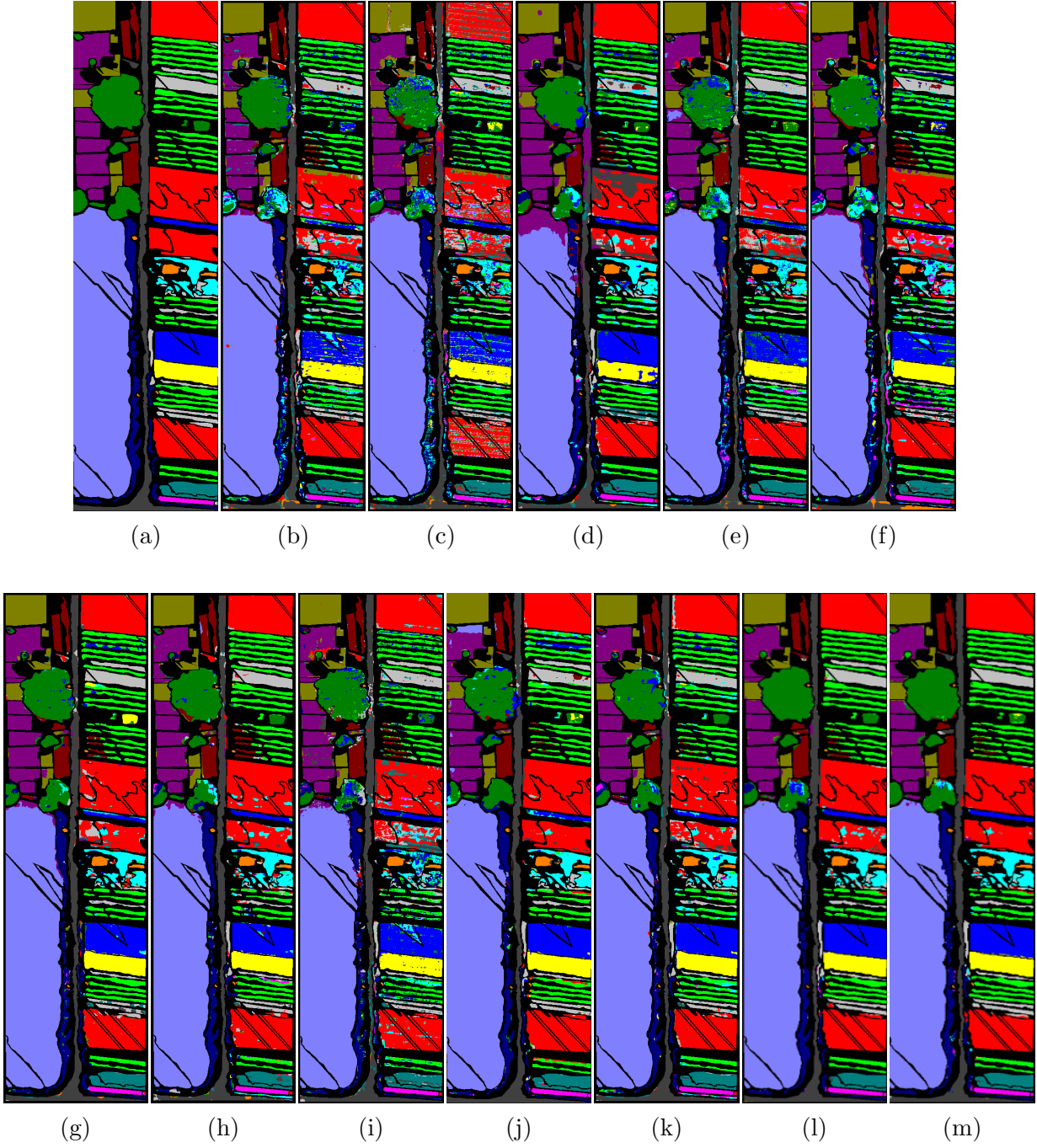
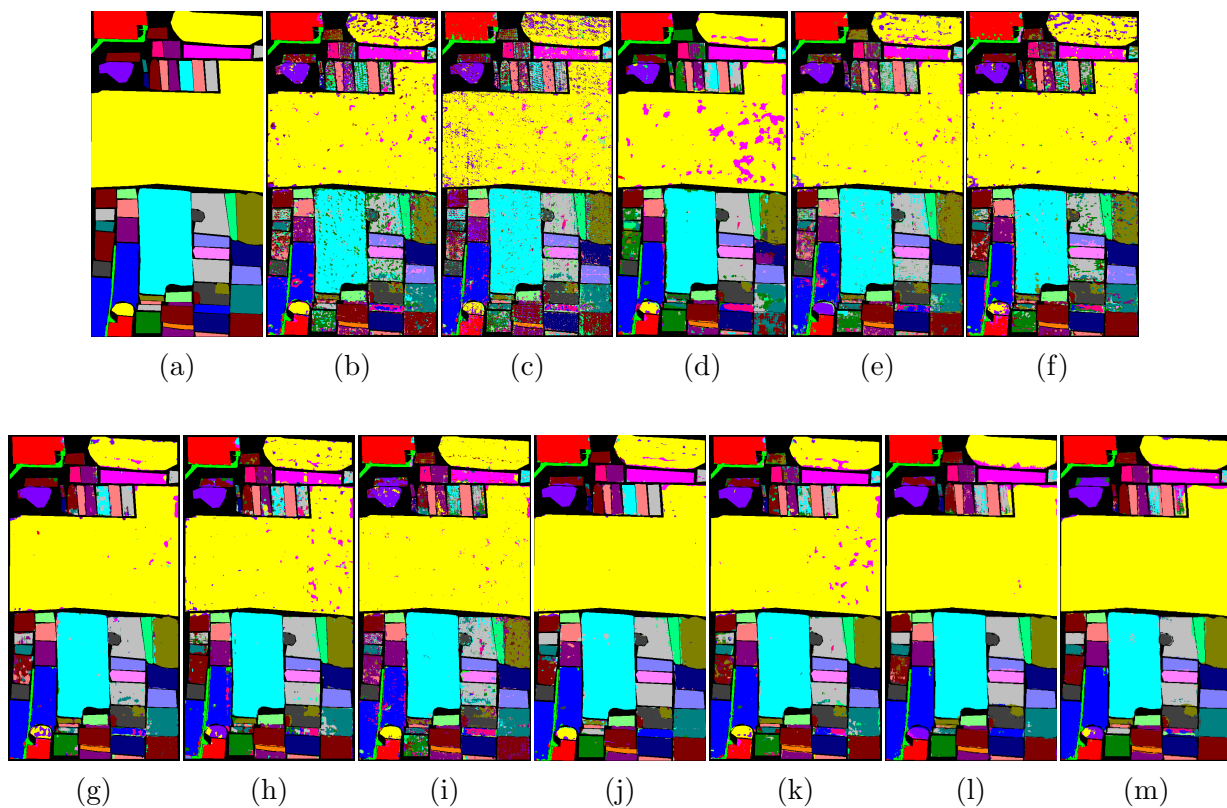


Figure 4.10: (a) Ground Truth, (b) 2D CNN, (c) 3D CNN, (d) CLOLN, (e) ViT, (f) SpectralFormer, (g) morphFormer, (h) SSFTT, (i) GSCViT, (j) MambaHSI, (k) MiM, (l) SSMamba, (m) SMaT-HSI (Ours).



configuration. These experiments were performed on all five benchmark datasets to ensure the robustness of our conclusions.

Effectiveness of Core Components

I first evaluated the individual and combined effects of the spectral branch (SpeMB), the spatial branch (SpaMB), and the structure-aware state fusion module. As detailed in Table 4.11, I tested four configurations: a spectral-only model, a spatial-only model, a combined model without fusion, and our full proposed model.

The results clearly show that both spectral and spatial information are vital, with the spatial context being particularly impactful. Comparing the "spectral only" and "spatial only" models, the spatial branch (SpaMB) consistently outperforms the spectral branch (SpeMB) across all five datasets. For example, on the Indian Pines (IP) dataset, the spatial-only model achieves an OA of 89.13%, which is significantly higher than the 63.87% OA from the spectral-only model. This underscores the rich discriminative power of spatial features.

Furthermore, I examined the role of our proposed fusion mechanism. By comparing the combined model without fusion (SpeMB + SpaMB) with the full model, the necessity of the state fusion module becomes evident. Adding the fusion module provides a substantial performance gain on all datasets. For instance, on the Pavia University (PU) dataset, its inclusion boosts the OA from 98.82% to 99.72% and the Kappa from 0.9876 to 0.9963. This demonstrates that our structure-aware state fusion is crucial for effectively integrating the features from the two branches to achieve superior classification accuracy.

Analysis of the Hybrid Transformer-Mamba Architecture

To determine the optimal arrangement of Mamba (SSM) and Transformer (self-attention) layers, we compared four different configurations, as shown in Table 4.12: a pure Mamba model (M4), a pure Transformer model (T4), and two hybrid models (M3T1 and M2T2).

The results reveal that a hybrid architecture is superior to a pure one, with the M3T1 model (3 Mamba layers followed by 1 Transformer layer) emerging as the most effective configuration. The M3T1 variant achieves the best overall performance on the majority of the datasets. For instance, on the Houston 2013 dataset, it obtains the highest AA (97.91%), OA (97.63%), and Kappa (0.9743).

Notably, the pure Mamba model (M4) generally outperforms the pure Transformer model (T4), highlighting the inherent strength of SSMs in capturing the long-range spatial-spectral dependencies in HSI data. The superior performance of the M3T1 model suggests

Table 4.11: Ablation experiment with different component in model on five datasets

SpeMB		✓		✓	✓
SpaMB			✓	✓	✓
State fusion					✓
IP	OA(%)	63.87	89.13	96.15	97.81
	AA(%)	78.21	90.26	97.28	98.83
	Kappa	0.7567	0.8942	0.9686	0.9748
PU	OA(%)	78.41	95.62	98.82	99.72
	AA(%)	76.52	94.89	98.73	99.73
	Kappa	0.7823	0.9876	0.9912	0.9963
HU2013	OA(%)	75.62	88.56	96.22	97.63
	AA(%)	76.29	88.91	97.06	97.91
	Kappa	0.7546	0.8927	0.9687	0.9743
HanChuan	OA(%)	85.12	90.12	96.04	96.21
	AA(%)	86.73	89.97	96.23	96.55
	Kappa	0.8662	0.8982	0.9513	0.9558
HongHu	OA(%)	74.35	89.07	95.86	96.52
	AA(%)	73.23	88.97	95.54	96.83
	Kappa	0.7413	0.8913	0.9501	0.9562

an effective synergy: the Mamba layers efficiently extract powerful contextual features, which are then globally refined and integrated by the final self-attention layer. This balance leverages the complementary strengths of both mechanisms and validates our architectural design choice.

Table 4.12: Comparison of different combinations of Transformer and Mamba blocks across five datasets.

Dataset	Metrics	M4	M3T1	M2T2	T4
IP	AA(%)	98.89	98.83	98.68	98.52
	OA(%)	97.54	97.81	97.81	97.43
	Kappa	0.9718	0.9748	0.9756	0.9704
PU	AA(%)	99.48	99.73	99.32	98.48
	OA(%)	98.72	99.72	99.43	98.46
	Kappa	0.9865	0.9963	0.9967	0.9832
HU2013	AA(%)	96.43	97.91	96.32	96.12
	OA(%)	96.37	97.63	95.98	95.78
	Kappa	0.9558	0.9743	0.9564	0.9512
HanChuan	AA(%)	96.23	96.55	96.17	95.78
	OA(%)	96.08	96.21	95.98	95.66
	Kappa	0.9587	0.9558	0.9526	0.9520
HongHu	AA(%)	96.38	96.83	96.57	96.12
	OA(%)	96.76	96.53	96.97	95.88
	Kappa	0.9536	0.9562	0.9624	0.9513

4.3.2 Complexity Analysis

To provide a comprehensive evaluation of the computational efficiency of our proposed method, we analyze two critical metrics: the number of trainable parameters and the floating-point operations (FLOPs). The former indicates the model’s size and memory requirements, while the latter quantifies its computational cost. A comparison was conducted against several state-of-the-art (SOTA) methods using the Pavia University dataset.

The FLOPs were calculated based on the multiply-accumulate operations (MACs) for each layer in the network, where one MAC is approximately equivalent to two FLOPs. For a given input tensor, the total FLOPs are the sum of operations across all layers, including

convolutional, fully-connected, and attention layers. Specifically, the calculation considers the forward pass for a single hyperspectral image patch with spatial dimensions of 27×27 and B spectral bands (where $B = 103$ for the Pavia University dataset). The results of this analysis are detailed in Table 4.13.

The results clearly demonstrate the efficiency of our approach. Our model has only 280.15 K parameters and requires just 73.62 M FLOPs. From the table, we can observe our method achieves a lower computational cost than most SOTA models. For example, Our model is significantly lighter than the 3D CNN, reducing the parameter count by nearly 90% (from 2620 K) and the FLOPs by over 94% (from 1420 M). While some models like SSFTT (152.83 K) have fewer parameters, our model requires substantially fewer FLOPs than methods like GSCViT (484.18 M) and morphFormer (257.33 M). Additionally, Our model is notably more efficient than its Mamba-based counterparts, including MIM and SSMamba, in both parameter count and computational operations.

In conclusion, while achieving state-of-the-art accuracy as shown in the previous sections, our model maintains a low computational profile. It strikes an excellent balance between performance and efficiency, making it a practical and effective solution for hyperspectral image classification.

Table 4.13: Comparison of different models on Parameters and FLOPs.

Models	CNNs			Transformers					Mamba			
	2D CNN	3D CNN	CLOLN	ViT	SpectralFormer	morphFormer	SSFTT	GSCViT	Mamba(HSI)	MiM	SSMamba	SMaT-HSI
Parameters (K)	1480	2620	4.55	358.6	301.74	226.92	152.83	613.26	412.24	1104	325.58	280.15
FLOPs (M)	78.87	1420	5.38	85.02	69.44	257.33	149.82	484.18	134.62	449.47	85.61	73.62

Chapter 5

Conclusion

5.1 Summary

In this study, I revisit three mainstream paradigms in deep learning architectures—CNNs, Transformers, and the recently proposed state-space Mamba architecture—in the context of hyperspectral image classification. Comprehensive evaluations on five benchmark public hyperspectral datasets consistently demonstrate that the Mamba-based backbone rivals or surpasses Transformer architectures, particularly excelling in capturing intricate spectral-spatial patterns while maintaining linear computational complexity. This balance of performance and efficiency makes Mamba a promising candidate for large-scale, resource-constrained hyperspectral image classification scenarios.

However, the adoption of one-dimensional scanning techniques in Mamba results in an inherent loss of spatial structure, potentially compromising the representational richness crucial for hyperspectral tasks involving spectral-spatial data. To address this limitation, this study proposes a structure-aware state-fusion module. This module explicitly encodes local spectral and spatial relationships within the latent feature space, effectively reducing redundancy and enhancing feature discriminability. The introduction of structure-aware encoding significantly bolsters the model’s capability to capture local dependencies, which is reflected in improved classification performance metrics across the evaluated hyperspectral datasets.

Additionally, several hybrid model configurations combining the strengths of Mamba and self-attention mechanisms were investigated. Among these, appending a lightweight self-attention stage to the tail end of the Mamba network proved effective. This configuration effectively harnesses the strengths of both architectures, maximizing global con-

text modeling with minimal computational overhead. Consequently, this hybrid approach yielded superior overall scores compared to purely Transformer-based or purely Mamba-based models in hyperspectral image classification tasks.

In summary, the findings presented in this paper underscore the potential of integrating structure-aware modules with the efficient and powerful Mamba architecture, augmented with strategic self-attention components, for enhanced hyperspectral image classification.

5.2 Limitations and future work

Despite the promising results, this study has several limitations that open avenues for future research.

First, our methodology relies on processing small, localized patches (e.g., 27x27 pixels) for classification. This patch-based approach, while common in HSI analysis, does not fully leverage two of the primary advantages of the Mamba architecture: its proficiency in capturing long-range dependencies and its linear computational complexity with respect to sequence length. The short sequences derived from small patches may not present a sufficient challenge to demonstrate Mamba’s full scaling potential compared to models with quadratic complexity like the Transformer.

Second, the study was conducted on legacy HSI datasets (e.g., Indian Pines, Pavia University) that are relatively small and have been extensively studied. While this allows for direct comparison with existing literature, training and testing on the same dataset provides limited insight into the model’s generalization capabilities. The resulting model is highly specialized to the specific sensor and geographic area of the training data and would likely exhibit poor performance if directly applied to a new, unseen HSI dataset from a different sensor or region.

Acknowledging these limitations, I propose the following directions for future work: First, future work can focus on curating a large-scale, diverse HSI dataset from multiple sensors and geographic locations. Training on such a dataset would enable the development of a more general, foundational model for HSI analysis with significantly improved generalization performance. Second, given the high cost and difficulty of acquiring labeled HSI data, leveraging the vast amount of available unlabeled data is crucial. Future research can investigate semi-supervised and unsupervised learning paradigms, such as contrastive learning or masked autoencoders, to pre-train the model. Third, to improve practical efficiency and fully utilize the model’s long-range modeling capabilities, future work can transition from patch-based classification to full-scene semantic segmentation.

References

- [1] Telmo Adão, Jonáš Hruška, Luís Pádua, José Bessa, Emanuel Peres, Raul Morais, and Joaquim Joao Sousa. Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry. *Remote sensing*, 9(11):1110, 2017.
- [2] Jón Atli Benediktsson, Jón Aevor Palmason, and Johannes R Sveinsson. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):480–491, 2005.
- [3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [4] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram 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.
- [5] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram 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.
- [6] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
- [7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [8] Christian Debes, Andreas Merentitis, Roel Heremans, Jürgen Hahn, Nikolaos Frangiadakis, Tim Van Kasteren, Wenzhi Liao, Rik Bellens, Aleksandra Pižurica, Sidharta Gautama, et al. Hyperspectral and lidar data fusion: Outcome of the 2013 grss data

- fusion contest. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2405–2418, 2014.
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
 - [10] Mathieu Fauvel, Yuliya Tarabalka, Jon Atli Benediktsson, Jocelyn Chanussot, and James C Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675, 2012.
 - [11] Giles M Foody. Status of land cover classification accuracy assessment. *Remote sensing of environment*, 80(1):185–201, 2002.
 - [12] Pall Oskar Gislason, Jon Atli Benediktsson, and Johannes R Sveinsson. Random forests for land cover classification. *Pattern recognition letters*, 27(4):294–300, 2006.
 - [13] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
 - [14] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.
 - [15] Jisoo Ham, Yangchi Chen, Melba M Crawford, and Joydeep Ghosh. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):492–501, 2005.
 - [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
 - [17] Danfeng Hong, Zhu Han, Jing Yao, Lianru Gao, Bing Zhang, Antonio Plaza, and Jocelyn Chanussot. Spectralformer: Rethinking hyperspectral image classification with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2021.
 - [18] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015(1):258619, 2015.

- [19] Lingbo Huang, Yushi Chen, and Xin He. Spectral-spatial mamba for hyperspectral image classification. *arXiv preprint arXiv:2404.18401*, 2024.
- [20] David Landgrebe. Hyperspectral image data analysis. *IEEE Signal processing magazine*, 19(1):17–28, 2002.
- [21] Shutao Li, Weiwei Song, Leyuan Fang, Yushi Chen, Pedram Ghamisi, and Jon Atli Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE transactions on geoscience and remote sensing*, 57(9):6690–6709, 2019.
- [22] Yapeng Li, Yong Luo, Lefei Zhang, Zengmao Wang, and Bo Du. MambaHSI: Spatial–Spectral Mamba for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16, 2024.
- [23] Guolan Lu and Baowei Fei. Medical hyperspectral imaging: a review. *Journal of biomedical optics*, 19(1):010901–010901, 2014.
- [24] Wenping Ma, Qifan Yang, Yue Wu, Wei Zhao, and Xiangrong Zhang. Double-branch multi-attention mechanism network for hyperspectral image classification. *Remote Sensing*, 11(11):1307, 2019.
- [25] Konstantinos Makantasis, Konstantinos Karantzalos, Anastasios Doulamis, and Nikolaos Doulamis. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In *2015 IEEE international geoscience and remote sensing symposium (IGARSS)*, pages 4959–4962. IEEE, 2015.
- [26] Farid Melgani and Lorenzo Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(8):1778–1790, 2004.
- [27] Swalpa Kumar Roy, Ankur Deria, Chiranjibi Shah, Juan M Haut, Qian Du, and Antonio Plaza. Spectral–spatial morphological attention transformer for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023.
- [28] Gary Shaw and Dimitris Manolakis. Signal processing for hyperspectral image exploitation. *IEEE Signal processing magazine*, 19(1):12–16, 2002.
- [29] Le Sun, Guangrui Zhao, Yuhui Zheng, and Zebin Wu. Spectral–spatial feature tokenization transformer for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.

- [30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [31] Chaodong Xiao, Minghan Li, Zhengqiang Zhang, Deyu Meng, and Lei Zhang. Spatial-mamba: Effective visual state space models via structure-aware state fusion. *arXiv preprint arXiv:2410.15091*, 2024.
- [32] Fulin Xu, Ge Zhang, Chao Song, Hui Wang, and Shaohui Mei. Multiscale and cross-level attention learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023.
- [33] Judy X Yang, Jun Zhou, Jing Wang, Hui Tian, and Alan Wee Chung Liew. Hsimamba: Hyperpsectral imaging efficient feature learning with bidirectional state space for classification. *arXiv preprint arXiv:2404.00272*, 2024.
- [34] Jun Yue, Wenzhi Zhao, Shanjun Mao, and Hui Liu. Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sensing Letters*, 6(6):468–477, 2015.
- [35] Zhuoyi Zhao, Xiang Xu, Shutao Li, and Antonio Plaza. Hyperspectral image classification using groupwise separable convolutional vision transformer network. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–17, 2024.
- [36] Yanfei Zhong, Xin Hu, Chang Luo, Xinyu Wang, Ji Zhao, and Liangpei Zhang. Whu-hi: Uav-borne hyperspectral with high spatial resolution (h2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with crf. *Remote Sensing of Environment*, 250:112012, 2020.
- [37] Zilong Zhong, Jonathan Li, Zhiming Luo, and Michael Chapman. Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework. *IEEE transactions on geoscience and remote sensing*, 56(2):847–858, 2017.
- [38] Weilian Zhou, Sei-ichiro Kamata, Haipeng Wang, Man Sing Wong, and Huiying Cynthia Hou. Mamba-in-mamba: Centralized mamba-cross-scan in tokenized mamba model for hyperspectral image classification. *Neurocomputing*, 613:128751, 2025.