Chapter 2

Literature Review

This chapter provides a comprehensive literature survey on HSI classification methodologies, focusing on their evolution from traditional ML techniques to advanced DL frameworks with respect to feature extraction techniques. Section 2.1 introduces the fundamental concepts of HSI with its wide-ranging applications and the classification methods used. It also traces the progression of HSI classification techniques, starting from ML-based methods and advancing to DL-based approaches. Section 2.2 delves into classical ML models, including support vector machine (SVM), nearest neighbor (KNN), K-means, random forest (RF), and partial least squares discriminant analysis (PLS-DA), highlighting their functionalities and inherent limitations in addressing the complexities of HSI data. The transition to deep learning methods is explored in Section 2.3, where advanced architectures such as stacked autoencoder (SAE), deep belief network (DBN), CNN, RNN with subsection LSTM and ViT, GAN, and GCN are discussed.

Section 2.4 outlines the main categories of DL models, followed by Section 2.5, which classifies them based on feature focus—spectral-only, spatial-only, and spectral-spatial. Section 2.6 further categorizes these models by the contextual scale of information captured: local, global, or both. Section 2.7 highlights the research gaps and objectives identified in the review. Finally, Section 2.8 summarizes the key findings, offering a structured basis for identifying limitations and guiding future research in hyperspectral image classification.

Based on previous research, the literature survey offers an overview of widely used DL-based algorithms for performing HSI classification operations. Figure2.1 presents information on the broad distribution of DL models covered in our survey. This article mainly comprises six parts: a) a description of some standard ML models, b) an outline of the primary issues with HSI classification that standard ML algorithms cannot re-

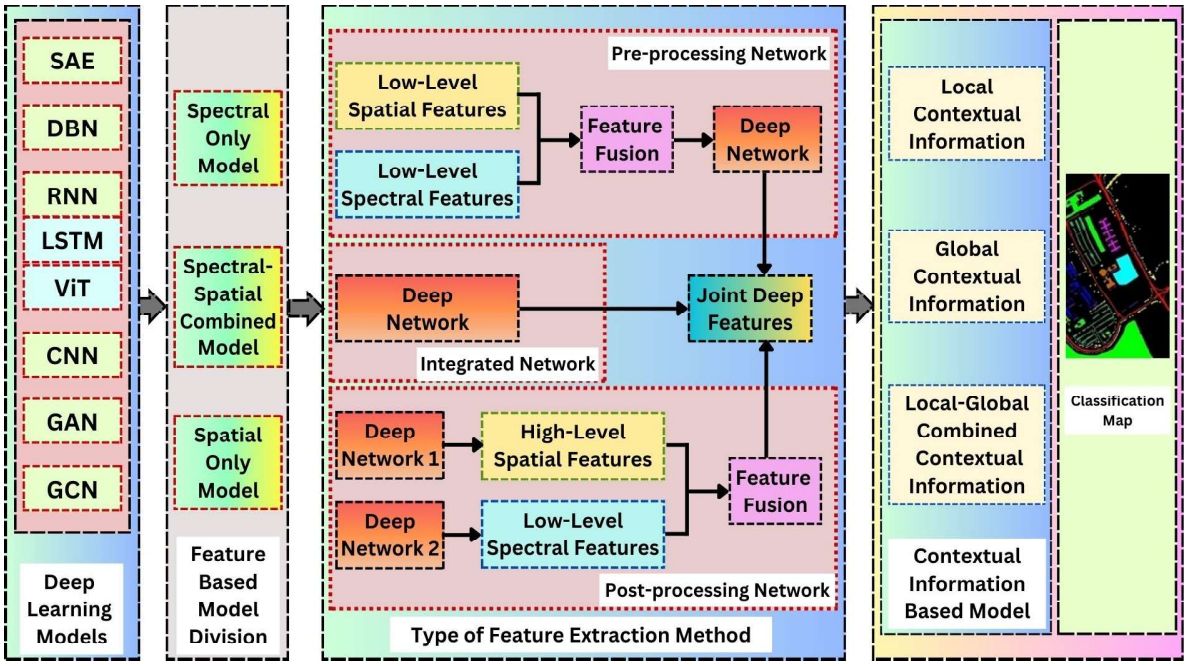


Figure 2.1: The HSI-classification framework employs diverse DL models, categorized across columns. The first outlines DL model types, the second features-based models. The third illustrates feature extraction methods: Pre-processing, Integrated, and post-processing networks. The fourth discusses hybrid methodologies, with limited sample methodologies in the final column.

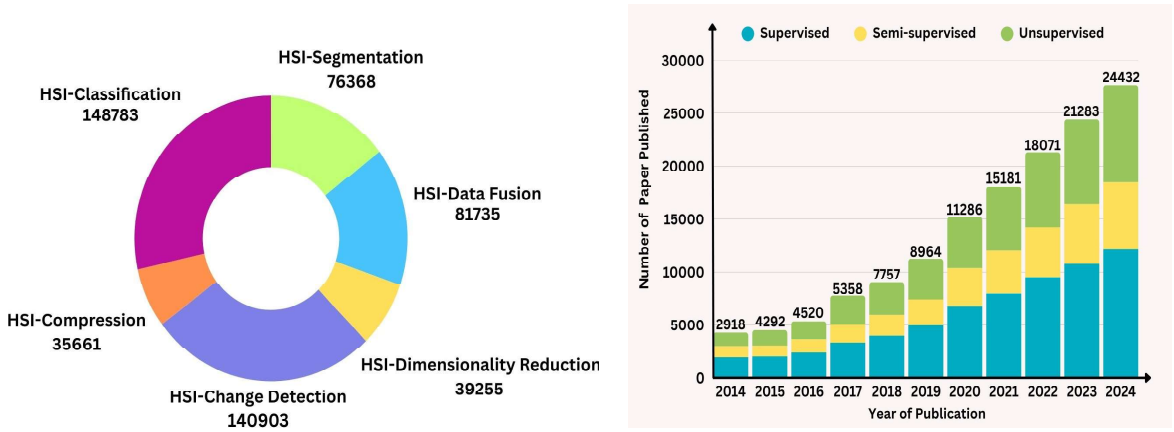
solve, c) a description of some standard DL models, d) a discussion on the capabilities of DL in addressing the previously covered concerns, and e) provision of a framework for classifying the corresponding works. This chapter presents a detailed exploration of various aspects of HSI classification, which include the following components:

1. This survey provides a comprehensive analysis of HSI, highlighting its significance and diverse applications in agriculture, environmental monitoring, defense, healthcare, and industrial inspection, as well as the categorization of techniques based on learning: supervised, unsupervised, and semi-supervised.
2. he survey explores traditional ML approaches, highlighting their contributions to HSI classification. It also discusses the inherent limitations of ML models, including their reliance on handcrafted features and sensitivity to high-dimensional data, which necessitate the transition toward deep learning-based methodologies.
3. The survey categorizes DL based HSI classification models based on feature extraction strategies, distinguishing among spectral-based, spatial-based, and integrated spectral-spatial approaches to enhance classification accuracy.
4. In addition, it classifies DL based HSI models based on the utilization of con-

textual information, categorizing them into local context-based, global context-based, and hybrid approaches that integrate both local and global contextual features for improved robustness and generalization.

2.1 HSI Fundamentals and Applications

HSI, covering wavelengths from 400 to 2500 nm, is a powerful remote sensing tool that enables classification, target detection, spectral unmixing, and anomaly detection [28]. It captures detailed spectral information, forming a data cube where each spectral band contains an equal number of pixels, and spatial resolution is determined by the sensor. This capability makes HSI invaluable across various domains, including military surveillance for monitoring troop movements [29], agriculture for crop health assessment [30], manufacturing for defect detection [31], astronomy for celestial studies [32], and disaster monitoring for tracking floods and droughts [33]. The growing significance of HSI has driven research into key areas such as segmentation [34], data fusion [35], dimensionality reduction [36], change detection [37], compression [38, 39], and classification [40]. The increasing research attention, as highlighted by "app.dimension.ai" in Figure 2.2 (a)[1], underscores HSI's expanding role in scientific and industrial applications.



(a) (b)

Figure 2.2: The pie chart (a) illustrates the distribution of articles related to HSI applications published between 2014 and 2024, while the bar chart (b) presents the year-wise number of publications categorized into supervised, unsupervised, and semi-supervised DL methods during the same period.

[1]https://app.dimensions.ai/discover/publication

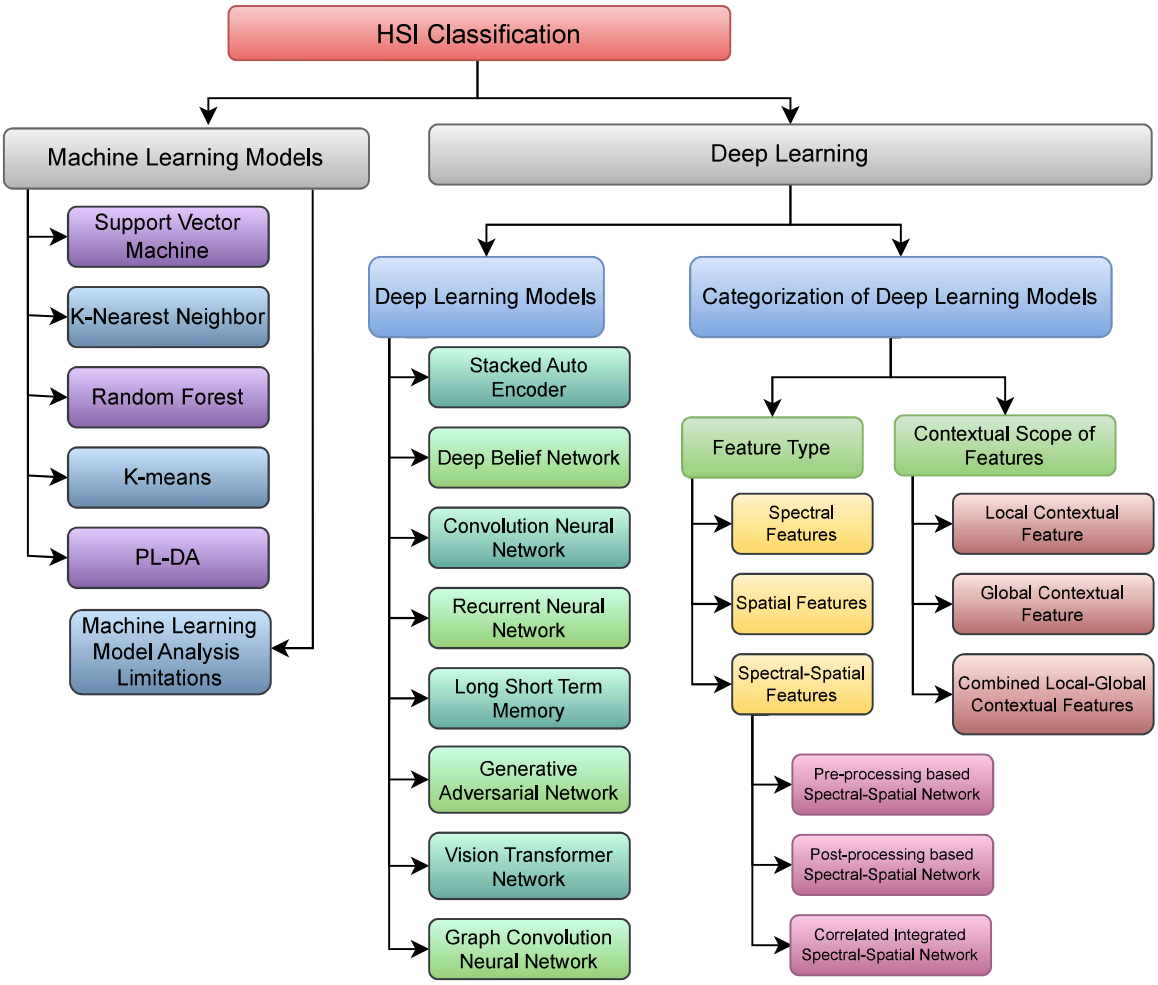


Figure 2.3: Taxonomy of HSI classification methods divided into ML and DL approaches. DL methods are further categorized by feature and by contextual scope.

2.1.1 Challenges and Feature Extraction Strategies in HSI Classification

HSI classification plays a crucial role in pattern recognition, attracting extensive research interest [17]. It involves assigning class labels to pixels based on spectral and spatial features but faces two primary challenges [41, 42]: (a) spectral variability due to lighting conditions, environmental factors, and temporal changes, leading to inconsistencies in classification, and (b) limited labeled training data, which heightens the risk of overfitting in data-driven models. Feature extraction in HSI classification follows two primary approaches: (1) spectral, spatial, or integrated spectral-spatial methods, and (2) local, global, or integrated local-global strategies. These techniques fall within broader training methodologies, which are categorized into supervised learning (relying on labeled data), semi-supervised learning (utilizing both labeled and unlabeled data),

and unsupervised learning (operating solely on unlabeled data). The increasing research focus on HSI classification, as illustrated in Figure 2.2 (b), underscores its expanding applications across diverse fields.

As illustrated in Figure 2.3, the taxonomy of HSI classification methods is broadly divided into ML and DL approaches. The DL approaches are further divided into conventional DL models and categorization of DL models, which are organized by feature types such as spectral, spatial, and spectral–spatial representations, and by contextual scope such as local, global, and local–global combined. Furthermore, the spectral–spatial combined representation is subdivided into pre-processing based, post-processing based, and integrated feature extraction based approaches.

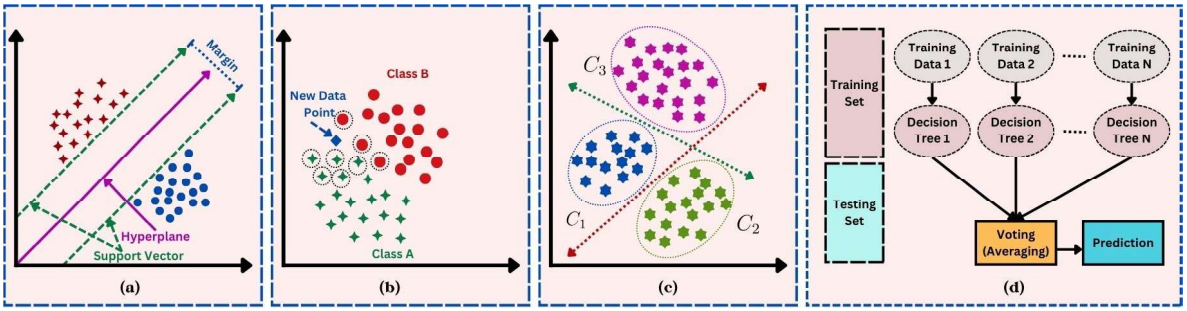


Figure 2.4: Different types of ML-based HSI classification models (a) SVM, (b) KNN, (c) K-means Clustering, and (d) Random Forest. Different colors with different types of shapes represent different types of objects.

2.2 Machine Learning Models

This subsection provides a detailed examination of the algorithms powering five conventional ML models that are prominently applied in HSI classification: SVM, K-NN, K-means clustering, RF, and PL-SDA. These models are analyzed in terms of their operational principles, mathematical frameworks, and applicability to extract meaningful patterns from high-dimensional HSI data.

2.2.1 Support Vector Machine (SVM)

SVM [43] is a powerful binary classification algorithm that separates data using a linear hyperplane, as shown in Figure 2.4 (a). Given n training samples as N-dimensional vectors $x_i \in \mathbb{R}^N$, where $i \in \{1, 2, 3, \ldots, n\}$, each is associated with a class label $y_i \in \{+1, -1\}$. The classifier follows a linear decision function:

$$f(x) = w \cdot x + b, \tag{2.1}$$

where $w \in \mathbb{R}^N$ defines the hyperplane's orientation, and b represents the bias term that adjusts the position of the hyperplane. SVM optimizes the hyperplane to maximize the margin between classes, improving generalization on unseen data. [44] introduced a multi-SVM framework to improve inter-class separability and reduce intra-class compactness. Similarly, [45] proposed SOM-SVM, where SVM extracts key features and self-organizing maps project data into a lower-dimension for better reclassification.

2.2.2 K-Nearest Neighbor (KNN)

The KNN algorithm classifies data using similarity metrics like Manhattan, Euclidean, or P-norm distance. A test instance ($x^{(t)}$) is assigned the majority class of its K nearest training samples (x_i), where $i = 1, ..., n$. As shown in Figure 2.4 (b), this process is crucial for binary classification. The Euclidean distance is given by:

$$d(x^{(t)}, x_i) = \|x^{(t)} - x_i\|. \tag{2.2}$$

Assuming we have identified k nearest neighbors belong to classes $l_1, l_2, ..., l_k$, KNN assigns $x^{(t)}$ to the most frequent class among them. The classification is represented as:

$$l_t = \underset{j=1,2,...,k}{max} \sum_{i=1}^{k} \delta(l_i, j) \tag{2.3}$$

where δ denotes the Kronecker delta function. The choice of distance metric is a critical element of the KNN algorithm, as it significantly influences the determination of the nearest point to $x^{(t)}$. Initially, [46] proposed a KNN-based model leveraging HSI and pixel data to construct a feature space for label assignment via sparse joint modeling. Later, [47] enhanced classification by combining KNN with guided filtering to reduce noise and improve spatial feature extraction.

2.2.3 Random Forest (RF)

RF is an ensemble learning technique that aggregates multiple decision trees, denoted as $\{h(X, \theta_k); \forall k = 1, \ldots, N\}$, where k represents the variable index and N is the total number of variables. It constructs k decision trees by randomly sampling training subsets with replacement. The final classification is determined through majority voting, formulated as:

$$\mathcal{H}(x) = \underset{y_j}{argmax} \sum_{i \in [1,2,..k]} \mathcal{I}(h_i(x) = y_j), j = 1, 2, ..., \mathcal{C}, \tag{2.4}$$

where the composite model is denoted as $\mathcal{H}(x)$, with $h_i(x)$ representing the decision tree model from the i^{th} training subset, and y_j corresponding to the output labels across $\mathcal{C}$ distinct classes. The function $\mathcal{I}(\cdot)$ indicates the decision-making strategy employed. As illustrated in Figure 2.4 (d), the classification process utilizing RF methods is demonstrated. [48] proposed Cascaded Random Forest (CFR), integrating the Hierarchical Random Subspace Method (HRSM) for feature selection and an out-of-bag (OOB) error-based boosting mechanism to enhance decision tree performance. Additionally, [49] introduced RoRF-KPCA, which applies Kernel Principal Component Analysis (Kernel-PCA) to feature subsets before classification using RF, with final predictions determined through majority voting.

2.2.4 K-means

K-means is an unsupervised clustering algorithm that groups data points based on similarity, typically using the Euclidean distance. The objective is to minimize the distance between each data point and its assigned cluster centroid. The algorithm begins with the initialization step, where k centroids, represented as $C = \{c_1, c_2, \ldots, c_k\}$, are randomly selected. In the assignment step, each data point x_j is allocated to the nearest centroid based on the Euclidean norm, which is mathematically expressed as:

$$s_j = \arg\min |x_j - \mu_i| \tag{2.5}$$

where$\|.\|$ represents the Euclidean distance. Once all data points are assigned to clusters, the centroids are updated by computing the mean of all data points within each cluster, given by

$$c_i = \frac{1}{|S_i|} \sum_{j \in S_i} x_j \tag{2.6}$$

where S_i represents the set of points belonging to cluster i. This assignment and update process is repeated iteratively until convergence, ensuring that centroids remain stable. Initially, Zhang et al. [50] propose a hierarchical dropout k-means framework for unsupervised spatial feature extraction. Further, Shu et al. [51] introduce a hybrid approach integrating K-means with PCA for spectral-spatial feature learning, employing SVM for classification. This method refines centroids through averaging and gradient descent. Although K-means is efficient and straightforward, its sensitivity to centroid initialization and difficulty in selecting the optimal kk often result in convergence to a local minimum rather than the global optimum.

2.2.5 Partial Least Square Discriminant Analysis (PLS-DA)

Partial Least Square (PLS) [52], is a parametric linear classification method that leverages predictor variability while maximizing covariance with the response variable. It constructs latent components as linear combinations of predictors to enhance correlation with the target variable [53]. A key aspect of PLS is extracting eigenvectors from spectral matrices, generating scores that capture spectral variance and its correlation with the response variable [54]. Mathematically, it is represented as:

$$X = TP^{'} + E, \qquad Y = UQ^{'} + F \tag{2.7}$$

where X and Y denote the waveband and response variable matrices, respectively, T and U represent factor scores, P and Q are loadings, while E and F account for residuals or noise. In their investigation, [55] examined Sparse PLS-DA (SPLS-DA) and PLS-DA with VIP-selected bands for classifying commercial Pinus species, emphasizing the visible spectrum's role in differentiation. Further, [56] demonstrated a 10% accuracy improvement in invasive weed detection by integrating AEG hyperspectral data with LiDAR using SPLS-DA. Additionally, [57] introduced Tensor PLS (TPLS), which enhances HSI classification by directly processing three-way data for improved robustness. In plastic classification, [58] developed Hierarchical PLS-DA (HI-PLSDA), combining short-wave infrared (1000-2500 nm) and visible spectrum (400-750 nm) data. To refine spectral analysis, [59] proposed Interval Hierarchical PLSDA (HI-iPLSDA), incorporating interval variables for continuous dataset evaluation. Finally, HSI applications extend to food safety and forensics [60].

2.2.6 Machine Learning Model Analysis & Limitations

ML techniques offer efficiency in HSI classification but struggle with spectral complexity that leads to misclassifications. Traditional ML models, such as SVM [44], KNN [46], and RF, rely on handcrafted features, making them time-consuming and less adaptable to high-dimensional HSI data. SVM is prone to noise and lacks spatial awareness, while KNN, despite its computational efficiency, fails to leverage spatial features effectively. Some methods integrate spatial-spectral features, such as PGE-JKNCC-c [47], which enhances spatial context but struggles with adaptability in complex scenes. Similarly, the Cascaded Random Forest (CRF) model [48] effectively handles noise but demands high computational resources. Advanced feature selection techniques like Sparse Partial Least Squares Discriminant Analysis (SPLS-DA) [55, 56] improve efficiency by select-

ing key spectral bands but risk information loss. Tensor-based models such as Tensor Partial Least Squares (TPLS) [57] enhance classification accuracy but introduce computational complexity and interpretation challenges. Despite these advancements, ML models face significant limitations: a) dependence on handcrafted features requiring extensive domain expertise, b) high sensitivity to noise and spectral variability, c) poor generalization in high-dimensional data, leading to overfitting, d) performance reliant on data quality and quantity, and e) scalability issues with large datasets. In contrast, DL overcomes these challenges by automatically extracting spatial-spectral correlations, adapting to high-dimensional data, and improving scalability, making it more robust and effective for HSI classification.

2.3 Transition to Deep Learning Models

The shift from ML models to DL frameworks in HSI classification effectively overcomes the limitations of earlier approaches in handling high dimensionality, spectral-spatial complexity, and non-linearity. Although ML techniques rely on hand-crafted features, they often struggle to generalize in situations with limited labeled samples or complex data distributions. In contrast, DL models have the ability to learn hierarchical feature representations automatically, capturing both low- and high-level patterns. DL architectures such as SAE and DBN are proficient in extracting hidden patterns from HSI data. Furthermore, CNNs excel at learning spatial features, while RNNs are particularly effective at handling sequential spectral analysis. Among the RNN variants, LSTM networks successfully capture long-term dependencies by addressing the vanishing gradient problem, positioning them as well-suited for complex spectral sequences. The introduction of ViT further enhances sequence modeling by employing self-attention mechanisms, which capture long-range dependencies and global contextual information; thus, they often outperform RNNs in terms of scalability and parallelization. In addition, GANs contribute to data augmentation by generating realistic HSI samples to address the issues associated with limited labeled data. GCNs further extend the capabilities of DL by modeling non-Euclidean relationships in HSI data, enabling superior spectral-spatial classification.

2.3.1 Stacked Auto Encoder (SAE)

Autoencoders (AEs), introduced by [61], are specialized neural networks for unsupervised learning that generate compact representations of input data by reconstructing inputs at the output layer. The training process in SAE minimizes reconstruction er-

ror, allowing the representations to capture key input features. Figure 2.5 (a) shows a three-layer AE, featuring an input layer with d neurons, a hidden layer with l neurons, and an output layer with d neurons for reconstruction. When multiple AEs are

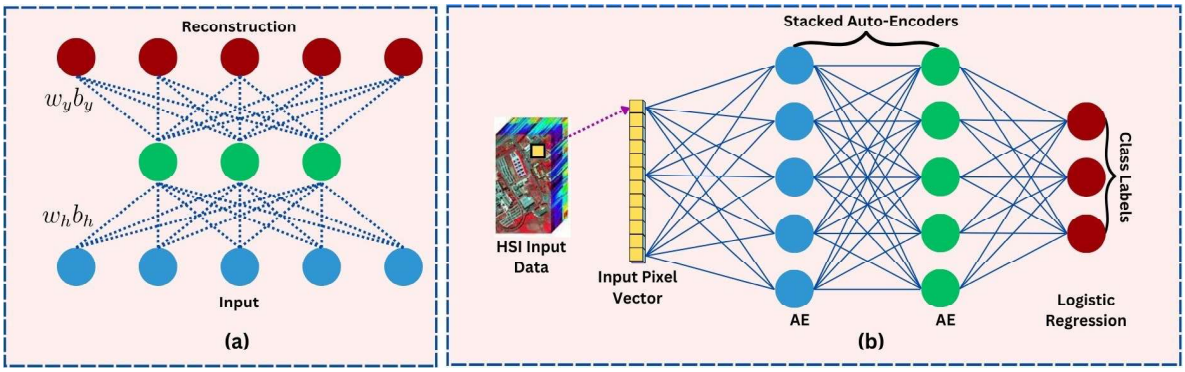


Figure 2.5: (a) Flow chart of Auto-encoder. (b) SAE-based HSI-classification Model.

stacked together, they form an SAE, enabling hierarchical feature extraction across layers. Each AE functions as an independent hidden layer, pre-trained sequentially, where the output of one AE serves as the input to the next.

The encoding process in SAE maps the input vector $\mathbf{x} \in \mathbb{R}^d$ into a latent representation $h \in \mathbb{R}^L$, which is then reconstructed back to output $\mathbf{y} \in \mathbb{R}^d$ using the transformation:

$$h = \mathfrak{V}h(\mathbf{W}h\mathbf{x} + b_h), \quad \mathbf{y} = \mathfrak{V}y(\mathbf{W}yh + b_y), \tag{2.8}$$

where $\mathbf{W}_h \in \mathbb{R}^{d\times L}$ and $\mathbf{W}_y \in \mathbb{R}^{L\times d}$ are weight matrices, while b_h and b_y are biases for encoding and decoding, respectively. The activation functions $\mathfrak{V}_h$ and $y\mathfrak{V}_y$ introduce non-linearity, enabling effective feature transformation. The AE parameters are optimized by minimizing the reconstruction error, measured by the Euclidean distance $\|\mathbf{x}-\mathbf{y}\|_2^2$, ensuring the extracted features preserve essential information. Finally Stacking multiple AEs forms a SAE, where each AE is pre-trained sequentially, with the output of one layer serving as input to the next. This hierarchical structure enhances feature extraction and deep representation learning. Figure 2.5 (b) illustrates an SAE integrated with a logistic regression classifier, where each pixel vector is processed individually, improving model performance across diverse applications.

2.3.2 Deep Belief Network (DBN)

In the DBN structure, illustrated in Figure 2.6 (a), the visible layer of the first RBM receives input data, while the output of each RBM serves as the input for the next.

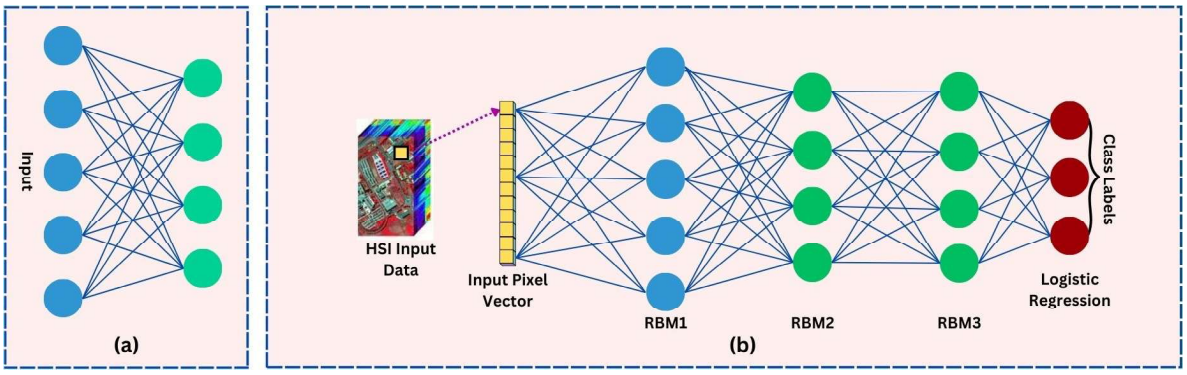


Figure 2.6: (a) Flow chart of RBM. (b) DBN-based HSI-classification Model

Training follows a two-step process: individual RBMs are pre-trained using an unsupervised greedy algorithm, followed by supervised fine-tuning via backpropagation [62]. For an RBM with n visible and m hidden neurons, the energy function $\mathcal{E}(v,h|\theta)$ and joint probability distribution $\mathcal{P}(v,h|\theta)$ are defined as:

$$\mathcal{E}(v,h|\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j,\ \mathcal{P}(v,h|\theta) = \frac{e^{-\mathcal{E}(v,h|\theta)}}{\mathcal{Z}(\theta)},\ \mathcal{Z}(\theta) = \sum_{v,h} e^{-\mathcal{E}(v,h|\theta)} \tag{2.9}$$

where a and b are biases for visible and hidden layers, v_i and h_j denote neuron states, and w_{ij} represents connection weights. Training optimizes $\theta = [w, a, b]$. Stacking multiple RBMs forms a DBN, extracting deep hierarchical features. Figure 2.6 (b) illustrates a DBN with RBM layers and a logistic regression classifier for classification.

2.3.3 Convolution Neural Network (CNN)

CNNs [63] are a class of feed-forward neural networks that integrate convolutional, pooling, and fully connected (FC) layers. Unlike traditional Artificial Neural Networks (ANNs), CNNs are designed to capture spatial hierarchies effectively, particularly for image recognition. Through a local connection scheme among adjacent neurons, CNNs exploit spatially local correlations, which are similar to the human visual cortex. This structure enables CNNs to process complex multi-dimensional data, as illustrated in Figure 2.7.

HSI data is represented as a hypercube $\mathbf{I} \in \mathbb{R}^{[H\times W\times B]}$ with height H, width W, and B spectral bands. The high dimensionality of HSI data poses computational challenges, which are addressed by the Dimension Reduction step, in generally PCA reduces the data to a lower-dimensional form $\mathbf{I}_R \in \mathbb{R}^{[H\times W\times P]}$, where P is the number of principal components. Input patches $\mathbf{I}^P \in \mathbb{R}^{[S_1\times S_2\times P]}$ are then extracted from $\mathbf{I}_R$ and processed by

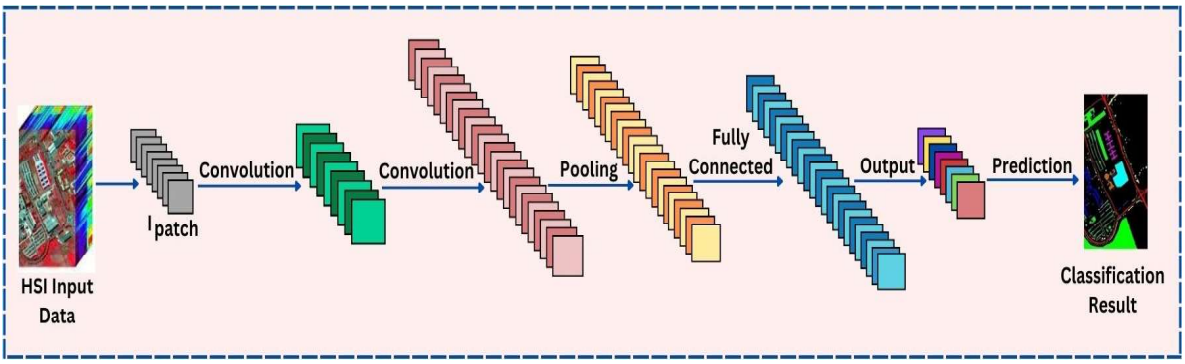


Figure 2.7: CNN-based HSI-classification Model. The model includes a convolution layer, a pooling layer, and a FC layer.

the CNN. The feature extraction phase of CNNs typically consists of three core layers, which are the fundamental components of most CNN architectures. These layers are:

A. Convolutional Layer: In a convolutional layer, the input patch $\mathbf{I}^P$ is convolved using a set of learnable filters, generating multiple feature maps. If $\mathbf{I}^P$ possesses dimensions $S_1 \times S_2 \times P$, where $S_1 \times S_2$ denotes the spatial dimensions and P indicates the number of channels, the i^{th} feature map can be denoted as $\mathbf{I}_i^P$. With k filters in a convolutional layer, each j^{th} filter can be characterized by its weight $\mathbf{W}_j$ and bias b_j, leading to the j^{th} output of the layer described by the expression:

$$\mathcal{L}_i = \sum_{i=0}^{P-1} \mathfrak{V}(\mathbf{I}_i^P * \mathbf{W}_j + b_j), \qquad j = 0, 1, \ldots, k-1 \tag{2.10}$$

where $*$ represents the convolution operation, and $\mathfrak{V}(\cdot)$ denotes an activation function that adds non-linearity to the network. The Rectified Linear Unit (ReLU) activation function is favored for its rapid convergence and resistance to the vanishing gradient issue. Its mathematical formulation, $\sigma(\cdot)$, is as follows: $\sigma(\mathbf{I}^P) = max(0, \mathbf{I}^P)$.

B. Pooling Layer: Pooling operations aggregate neuron outputs while preserving invariant features, effectively reducing spatial dimensions, parameters, and computations. This transformation enhances feature abstraction, making representations more compact. For instance, average pooling over an $m \times m$ window, denoted as $\mathcal{S}$, can be described as follows:

$$\mathfrak{Z} = \frac{1}{\mathcal{F}} \sum_{(i,j \in \mathcal{S})} \mathbf{x}_{ij}, \tag{2.11}$$

where $\mathcal{F}$ indicates the element count in $\mathcal{S}$, and $\mathbf{x}_{ij}$ represents the activation value at the position (i, j). $\mathfrak{Z}$ is the output from the pooling layer.

C. Fully Connected Layer: After the pooling layers, the feature maps are flattened and input into FC layers. These layers transform the feature maps into an n-dimensional vector, allowing for the extraction of more abstract features. An FC layer can be defined as follows:

$$\mathcal{Y} = \sum_{i=0}^{N-1} \mathfrak{V}(\mathbf{W}\mathbf{x}_i + b) \tag{2.12}$$

where $\mathbf{x}_i$, $\mathcal{Y}$, $\mathbf{W}$, and b denote the input feature map, output feature map, weight, and bias, respectively, of the FC layer.

2.3.4 Recurrent Neural Network (RNN)

RNN [64] extends traditional neural networks by integrating feedback loops, enabling them to process sequential data effectively. Unlike feed-forward networks, RNNs retain past information through internal memory, allowing previous computations to influence current output. This makes them particularly suited for tasks that require contextual understanding. Unlike CNNs, where each layer operates independently with unique weights, RNNs maintain consistent weights in hidden layers across sequences, facilitating pattern recognition and long-term dependency modeling.

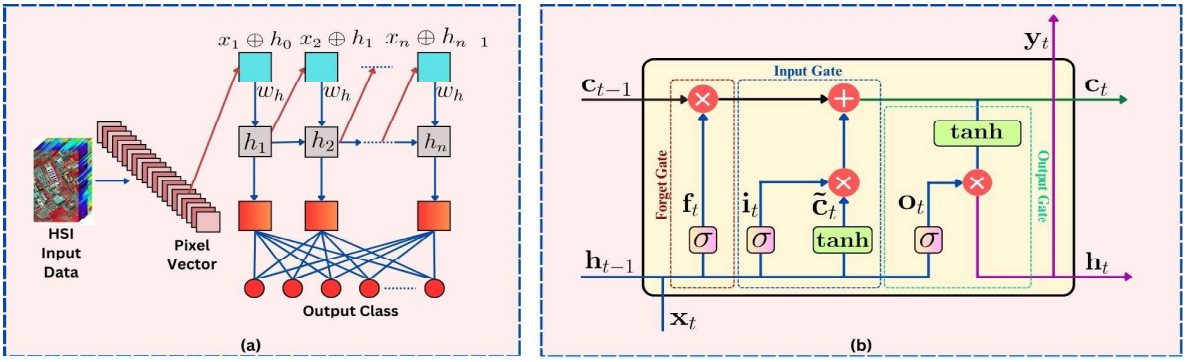


Figure 2.8: Flowchart illustrating (a) the RNN framework with sequential data processing through input, hidden states, and feedback loops, and (b) the LSTM cell framework highlighting the roles of input, forget, and output gates in managing cell states for improved sequential learning.

Figure 2.8 (a) depicts the framework of an RNN. The following update equations should be used for each time step from time $t = 1$ to $t = \tau$, given a sequence of $\{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(\tau)}\}$ values:

$$h^{(t)} = \mathfrak{V}_h(\mathbf{W}_i\mathbf{x}^{(t)} + \mathbf{W}_r h^{(t-1)} + b_h), \qquad y^{(t)} = \mathfrak{V}_y(\mathbf{W}_y h^{(t)} + b_y), \tag{2.13}$$

where the weight matrices for connections from hidden units to hidden units, from hidden units to output, and from hidden units to input are denoted as $\mathbf{W}_r$, $\mathbf{W}_y$, and $\mathbf{W}_i$. The variables $\mathbf{x}^{(t)}$, $h^{(t)}$, and $y^{(t)}$ represent the input, hidden state, and output at time step t. The bias vectors are b_h and b_y. The initial hidden state $h^{(0)}$ is initialized using values from a Gaussian distribution. For classification, a softmax function is applied at time step $t = \tau$ to determine the probability of the input being linked to the i^{th} class is expressed as:

$$\mathcal{P}(\mathcal{Y} = i|\alpha, b) = \mathcal{S}(y^{(\tau)}) = \frac{e^{\alpha_i y^{\tau} + b_i}}{\sum_{j=1}^{C} e^{\alpha_j y^{\tau} + b_j}} \tag{2.14}$$

where $\mathcal{Y}$ denotes the label of HSI data, α and b denote the weight matrix and bias vector, respectively. C denotes the number of classes.

RNNs are effective for sequential data processing but struggle with long-range dependencies due to the vanishing gradient problem. This hampers their ability to retain crucial information over extended sequences, leading to suboptimal performance in tasks like language modeling and time series prediction. To address these challenges, LSTM networks were developed, enhancing traditional RNNs by efficiently capturing long-term dependencies. Similarly, ViTs extend these capabilities to image data by leveraging self-attention mechanisms, allowing them to model global contextual relationships more effectively than conventional RNNs or CNNs.

2.3.5 Long Short Term Memory (LSTM)

LSTM networks [65] enhance RNNs by mitigating the vanishing and exploding gradient issues, enabling the learning of long-term dependencies in sequential data. Unlike standard RNNs, LSTMs utilize specialized memory cells with three key gates: the forget gate (f_t), input gate (i_t), and output gate (o_t), which regulate information flow. The forget gate determines whether past information is retained, the input gate controls updates to the memory cell, and the output gate influences the final state. Intermediate computations, including the candidate cell state ($\tilde{c}_t$) and updated cell state (c_t), contribute to the hidden state (h_t) or output (y_t) [66]. This architecture enables LSTMs to capture long-term dependencies effectively, overcoming the limitations of conventional RNNs. Figure 2.8 (b) illustrates this architecture, while the governing equations are detailed below:

$$f_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + b_f), \tag{2.15}$$

$$i_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + b_i), \quad \tilde{c}_t = \mathbf{tanh}(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + b_c), \tag{2.16}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad o_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + b_o), \quad h_t = y_t = \mathbf{tanh}(c_t) \odot o_t. \quad (2.17)$$

where, the forget gate (f_t) regulates how much of the past cell state (c_{t-1}) should be retained or discarded. The input gate (i_t) and candidate cell state ($\tilde{c}_t$) determine the amount of new information added to the cell state. The updated cell state (c_t) integrates past and new information, ensuring the model captures long-term dependencies. Finally, the output gate (o_t) modulates the final hidden state (h_t), which influences the output (y_t). The output gate (o_t) determines the visible output (y_t) based on the current cell state (c_t) and its transformation through the **tanh** function. In these equations, $\mathbf{W}_f$, $\mathbf{W}_f$, $\mathbf{W}_o$, $\mathbf{W}_c$, $\mathbf{U}_f$, $\mathbf{U}_i$, $\mathbf{U}_o$, and $\mathbf{U}_c$ are weight matrices, while b_f, b_i, b_o, and b_c represent bias vectors. The sigmoid function: $\sigma(x) = 1/(1+\exp(-x))$, and $\odot$ facilitate non-linear transformations and element-wise operations, respectively.

2.3.6 Vision Transformer Network (ViT)

The Vision Transformer (ViT), introduced by Dosovitskiy et al. [67], revolutionized image classification by adapting the Transformer model of Vaswani et al. [68] for vision tasks. Unlike LSTMs, which process data sequentially and struggle with long-range dependencies, Transformers utilize self-attention mechanisms for global contextual learning, which makes them particularly effective for HSI classification by integrating spectral and spatial information efficiently.

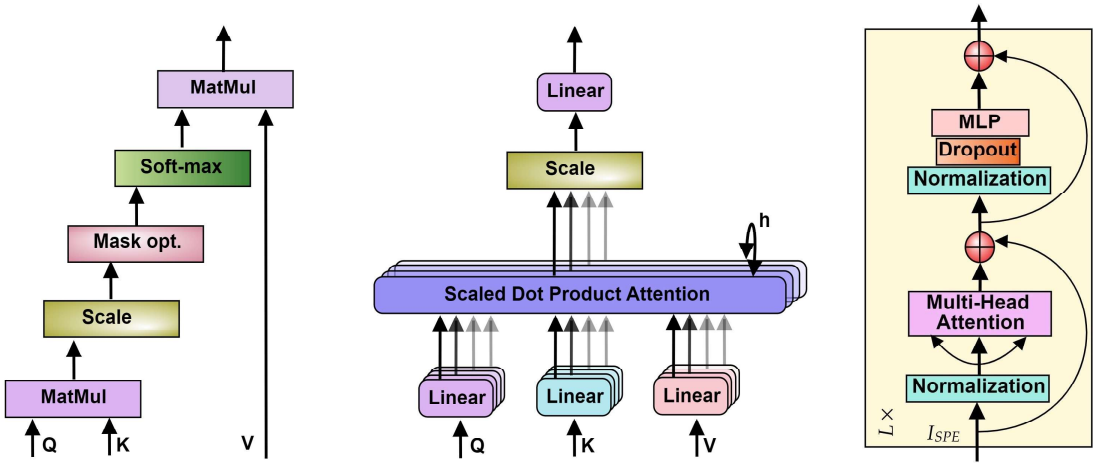


Figure 2.9: Illustration of the attention mechanism and the transformer encoder (a) Self-attention module, (b) Multi-head attention, and (c) Transformer Encoder.

The Transformer model for HSI classification includes two main components: the en-

coder, which extracts features from the input HSI data, and the classifier, which assigns these features to specific HSI classes. **Transformer Encoder:** The encoder is a key component of the ViT architecture, utilizing MHSA and feed-forward neural networks. As shown in Figure 2.9 (a) and (b), the self-attention mechanism and MHSA project input data into multiple feature subspaces, enabling parallel processing by independent attention heads. This allows the model to focus on important spectral and spatial dependencies. Figure 2.9 (c) illustrates the coordinated steps involved in the MHSA operation. Intially, the input HSI patch I_{SSF} is transformed into three matrices: query Q, key K, and value V, each of dimension $\mathbb{R}^{[N\times d]}$, where N is the sequence length, and d. The transformation equations are as follows:

$$Q = \mathbf{W}_q I_{SSF}, \qquad K = \mathbf{W}_k I_{SSF}, \qquad V = \mathbf{W}_v I_{SSF}. \tag{2.18}$$

here, $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$ are weight matrices of dimensions $d \times d$. The input sequence length N corresponds to the number of patches extracted from the HSI. Subsequently, Q, K, and V are split into h segments along the feature dimension, enabling parallel processing for computational efficiency. These segments are represented as:

$$Q = [Q_1, Q_2, \ldots, Q_h], \quad K = [K_1, K_2, \ldots, K_h], \quad V = [V_1, V_2, \ldots, V_h] \tag{2.19}$$

Each segment, Q_i, K_i, and V_i, has dimensions $\mathbb{R}^{[N\times d/h]}$, where h is the number of attention heads. For each head i, the scaled dot-product attention computes attention scores as follows:

$$\mathcal{Z}_i = Attention(Q_i, K_i, V_i) = Softmax(Q_i K_i^T/\sqrt{f_K})V_i \tag{2.20}$$

here, f_K is the dimensionality of the key vector, and $\sqrt{f_K}$ stabilizes the scaled softmax function to avoid large gradients. The outputs from the h attention heads are concatenated and linearly transformed to produce the final MHSA result:

$$MHSA(Q, K, V) = \mathbf{Concat}(\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3, \ldots, \mathcal{Z}_h)W, \tag{2.21}$$

where W is the parameter matrix. Finally, the output undergoes normalization, dropout, and an MLP layer, as depicted in Figure 2.9 (c), to enhance feature representation and capture long-range dependencies for HSI classification.

2.3.7 Generative Adversarial Network (GAN)

[69] introduced GAN, a groundbreaking DL architecture comprising two key components: a generator (G) and a discriminator (D). As depicted in Figure 2.10, the generator G is designed to approximate the true data distribution $p_d(x)$ by learning a generative distribution $p_g(x)$ that closely mimics $p_d(x)$. Concurrently, the discriminator (D) acts as a binary classifier, differentiating between samples originating from the true data distribution and those synthesized by the generator. The training process

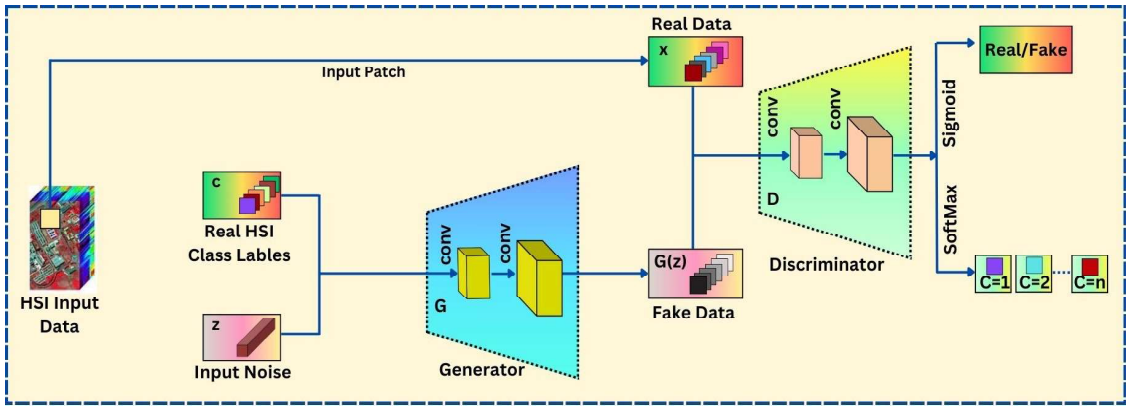


Figure 2.10: The generator (G) maps noise (z) to synthetic data, while the discriminator (D) differentiates real from generated samples. Their adversarial training process between G and D follows a min-max optimization process, enhancing the genuineness of generated outputs.

of GAN is formulated as a two-player min-max optimization problem, defined by the value function $V(G, D)$, which is expressed as follows:

$$\min_G \max_D V(G, D) = \mathbb{E}x \sim p_d(x)[\log D(x)] + \mathbb{E}z \sim p_z(z)[\log(1 - D(G(z)))] \quad (2.22)$$

here, x represents real data samples from the true distribution $p_d(x)$, and z denotes noise from a prior distribution $p_z(z)$. The G transforms the noise z into synthetic samples $G(z)$, while D, parameterized as a neural network, estimates the probability that a sample originates from p_d rather than p_g. Adversarial training updates the parameters of G and D iteratively. Meanwhile, the generator minimizes $\log(1-D(G(z)))$ to enhance its data realism, while the discriminator maximizes $\log D(x)$ and minimizes $\log(1 - D(G(z)))$ to distinguish real from generated data. This ongoing interplay helps $p_g(x)$ converge toward $p_d(x)$ over iterations and reduces overfitting in scenarios with limited samples. Figure 2.10 illustrates the roles of G and D in this process.

2.3.8 Graph Convolution Neural Network (GCN)

GCN [70] provides a robust framework for processing HSI data using non-Euclidean graph representations. As shown in Figure 2.11,HSI data form an undirected graph $G = (V, E)$, where vertices (V) represent pixels, and edges (E) capture spectral similarities. The adjacency matrix A reflects these connections, with each element $A_{i,j}$ measuring the similarity between pixels V_i and V_j using a radial basis function (RBF), defines as:

$$A_{i,j} = exp(- \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2 / \sigma^2), \tag{2.23}$$

where x_i and x_j are the spectral feature vectors corresponding to V_i and V_j, and σ controls the RBF's width. The graph convolution process begins with the construction of the adjacency matrix. Each pixel's spectral vector is used to calculate similarity values, forming a graph structure. Subsequently, the GCN applies its propagation rule to learn high-level feature representations iteratively. The propagation rule for the l^{th} layer is given by:

$$H^{(l+1)} = f(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}H^{(l)}\mathbf{W}^{(l)} + b^{(l)}), \tag{2.24}$$

where $\widetilde{A} = A + I$ is the normalized adjacency matrix (including self-loops), and $\widetilde{D}_{i,j} = \sum_j \widetilde{A}_{i,j}$ is the degree matrix. This normalization enhances training stability. Here, $H^{(l)}$ represents the layer's input features, $f(\cdot)$ is the activation function (e.g., ReLU), $W^{(l)}$ is the learnable weight matrix, and $b^{(l)}$ is the bias term.

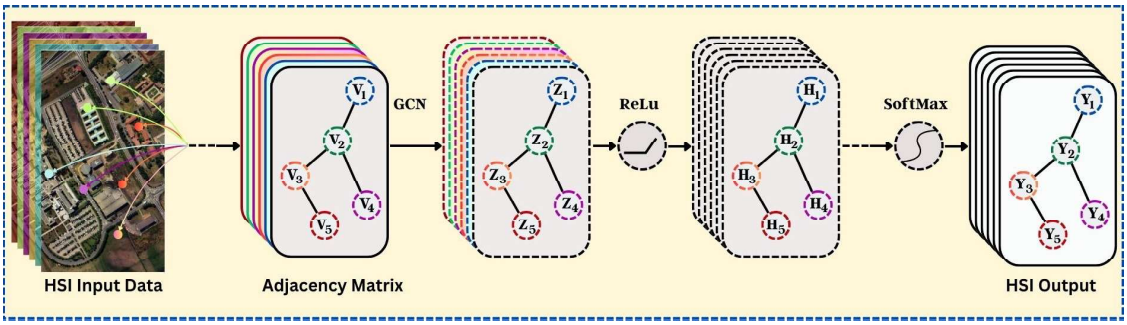


Figure 2.11: A simplified representation of the GCN process for HSI classification, showing input data as a graph, adjacency matrix construction, feature propagation through GCN layers, and final classification using softmax.

In the figure 2.11, the GCN updates node features sequentially for each node V. It starts with raw HSI data, performing graph convolution to create intermediate features $(Z_1, Z_2, \dots)$ through learned weight transformations. Next, ReLU activations enhance non-linear representation, and the output layer applies a softmax function to classify each pixel into its spectral category, yielding the HSI output.

2.4 Categorization of DL Models for HSI Classification

DL has significantly advanced computer vision [71] and has become instrumental in HSI classification. It outperforms traditional ML [10, 17] by autonomously extracting high-level abstract features while effectively integrating spatial and spectral properties. Additionally, DL models can be categorized based on two criteria:

1. **Feature Type:** This includes spectral models that focus on spectral information, spatial models that capture structural patterns, and hybrid models that integrate both to mitigate intra-class variability and inter-class similarity
2. **Context of Features:** This classification distinguishes between models that capture local features, global features, or a combination of both, ensuring long-range dependencies are preserved for enhanced classification.

.

2.5 DL Model Categorization Based on Feature Type

DL-based HSI classification methods can be classified into spectral, spatial, and hybrid models. Spectral models analyze pixel-wise reflectance for precise material differentiation, spatial models capture structural patterns, and hybrid models integrate both to address intra-class variability and inter-class similarity.

2.5.1 DL Models Leveraging Only Spectral Features

HSI analysis encapsulates both spatial and spectral information, with spectral features being crucial for identifying material characteristics. Spectral feature-based DL models focus on extracting discriminative spectral vectors from image patches. Table 2.1 summarizes their advantages, limitations, and OA, along with future directions. [72] introduced a 1D Convolutional Neural Network (1D-CNN) for HSI classification using spectral vectors through convolution, max-pooling, FC, and output layers. Extending this, [73] proposed a Pixel-Pair Framework (PPF), incorporating neighboring pixels via majority voting during CNN training. To mitigate overfitting due to limited samples, [74] developed a Bayesian CNN (1D-B-CNN) that utilizes Bayesian inference for improved generalization. Given the sequential nature of spectral data, RNNs are also effective in this context. [75] designed an RNN for HSI classification, further enhancing the Gated Recurrent Unit (GRU) with tanh activation. Later, [76] proposed a Convolutional-RNN (CRNN), integrating CNN and RNN with decision fusion to improve classification accuracy. However, DL models based solely on spectral features

Table 2.1: DL Models using Spectral Features. Here, all the models take 200 samples per class for training except 1D-B-CNN and RNN with GRU.

Model	Advantages	Limitations & Future Scope	Performance
1D-CNN [72]	Exploits spatially local correlations by enforcing a regional connectivity pattern between neurons of adjacent layers.	Prone to overfitting with limited training samples. GPU implementation can optimize training/testing time.	OA for IP, PU, SA: 90.16%, 92.56%, 92.60%.
CNN-PPF [73]	Pixel-pair method improve sample diversity and OA via neighborhood pixel classes.	Limited sample variations risk overfitting. Determining optimal pixel-pair configurations remains computationally expensive.	OA for IP, PU, SA: 94.34%, 96.48%, 94.80%.
1D-B-CNN [74]	Bayesian CNN reduces overfitting for small datasets.	OA of unlabeled data labeling remains a bottleneck. Transfer learning (TL) may prove beneficial in enhancing OA.	OA for IP, SA, KSC: 86.15%, 93.57%, 97.27%.
RNN-GRU [75]	PRetanh activation ensures high learning rates without divergence during training.	RNN can only store one instance of the previous memory. So, LSTM can offer better performance compared to RNN.	OA for IP, PU, HU: 88.63%, 88.85%, 89.85%.
CRNN [76]	Combines CNN's local invariance and RNN's spectral-contextual capabilities.	Suffers from vanishing/exploding gradient issues and class imbalance. An adaptive loss functions could improve results.	OA for IP, HU: 96.98%, 98.61%.

encounter specific limitations—for instance, 1D-CNN [72] and CNN-PPF [73] risk overfitting with limited samples. Moreover, CNN-PPF lacks sample diversity, amplifying this issue. Similarly, 1D-B-CNN [74] struggles with accurately labeling unlabeled data. Meanwhile, RNN-GRU [75] has limited memory, which LSTM could mitigate. Finally, CRNN [76] suffers from gradient issues and class imbalance, highlighting the need for advanced augmentation and adaptive loss strategies.

2.5.2 DL Model Leveraging Only Spatial Features

Integrating spatial correlations improves the accuracy and efficiency of HSI classification. This section explores DL-based networks that extract spatial features from HSIs. Table 2.2 outlines their advantages, limitations, performance metrics, and future research directions. In [77–79] authors have used PCA for dimensionality reduction to minimize computation, followed by 2D-CNN for hierarchical spatial feature extraction.

Moreover, [80] employed PCA to extract the first Principal Component (PC) as training labels, using a deconvolution layer for map reconstruction and Extreme Learning Machines (ELM) for classification. In [81], a deep attention module with varied convolutional kernels and residual connections enhances spatial context. Finally, the dual-channel Residual 2D-CNN (R-2D-CNN) [82] processes two patch sizes and merges feature maps before the FC layer. In contrast, Fang et al. [83] propose a Deep Hashing Neural Network (DHNN), where a hashing layer after the FC layer converts real-valued features into binary codes for efficient distance computation. Additionally, Cao et

Table 2.2: DL Models Utilizing Spatial Features. Here, TL is transfer learning.

Model	Advantages	Limitations & Future Scope	Performance
DEEP0 ([77])	Sparse representation leverages a high-dimensional subspace to minimize computations and improve feature discrimination.	Sparse representation algorithms face memory and processing constraints. This leads to develop more efficient methods for large datasets.	OA for IP and PU: 97.25%, 98.35% with 10% training samples.
[78]	CNN enables global average pooling (GAP), reinforcing correlations between feature maps and classes.	GAP deviates from dominant features while using averaging. Reducing computational cost is crucial for broad spatial coverage in HSI data.	OA for IP, PU, SA: 64.19%, 81.75%, 85.24% with 3 samples per class for training.
[80]	Deconvolution layers densify feature maps, improving output resolution.	Mean square error as a loss function is unsuitable for multi-class classification and fails to handle class imbalance.	OA for IP and PU: 96.70% and 95.11%.
AI-NET [81]	Attention inception modules with residual connections adapt to small training data using various filter sizes.	Attention mechanisms add weights, increasing training time. Dilated convolution could accelerate training.	OA for IP and SA: 93.07% and 94.64% with 200 samples per class for training.
R-2D-CNN [82]	The model uses two input patches of different sizes to reduce noise, with the final classification relying on pixel cues as the patches shrink.	The large number of parameters and slight variations in training samples may cause overfitting. Investigating TL techniques could help overcome these limitations.	OA for PU, PC, SA, KSC, BS: 99.19%, 99.88%, 99.47%, 99.22%, 98.54%, respectively with 30% samples for training.
DHNN [83]	Hash learning efficiently computes feature distances without relying on Euclidean distance.	Loss function underperforms with class imbalance. Focal loss could mitigate this issue.	OA for PU and SA: 98.61%,99.34%.Trains on 200 samples per class.
CNN-AL-MRF [84]	AL identifies key pixels for annotation, and MRF leverages spatial correlations, assuming neighboring pixels are similar.	Focus on spatial features limits the ability to extract patterns. Extending to semi-supervised or unsupervised frameworks could enhance versatility.	OA: IP, PU, PC are 94.28%, 98.17%, and 99.15%, respectively.
FCSN [85]	Enhances spatial diversity in land-cover distributions via fine-label HSI cube generation.	Generating realistic distributions with the proposed cube is challenging. Semi-supervised networks could utilize labeled and unlabeled pixels for better outcomes.	OA for IP and PU are 90.45% and 98.19% with 10% and 5% of training data.
HDCFE-Net ([86])	Dilated convolution expands the receptive field while retaining spatial features, aiding in feature extraction.	Despite improving the receptive field, dilated convolution may smooth object edges. Incorporating spectral features could boost accuracy.	OA for IP, PU, and SA are 77.99%, 89.23%, and 91.19%, respectively.
MGCNN ([87])	Combines Gabor wavelets with CNN kernels to handle gray value variations and expand receptive fields.	Additional steps with Gabor wavelets increase computational complexity compared to standard CNNs.	OA for Blood cell 1-3, Blood cell 2-2, and White-blood cell are 94.03%, 94.40%, and 97.75%.
LSTM-CNN ([88])	Black Widow Optimization as well as Mayfly Optimization Approach (HBWO-MOA) improves feature selection with bias field correction.	While CNN captures spatial features, LSTM addresses temporal dependencies. The model lacks full spatial data utilization, potentially lowering performance in spatial-context tasks.	OA for DDTI is 98.80% with 80% of training data.
CNN-SVM ([89])	CNN + SVM with a 1×1 kernel for efficient parameter use.	2D-CNNs have limited capacity to capture both spatial and spectral features, complicating high-dimensional hyperspectral data handling.	OA for C. indicum HSI data is 93.48% with 70% of training data.

al. [84] integrate Active Learning (AL) with CNN, where CNN extracts 3D-HSI features and AL selects the most informative pixels. Similarly, Sun et al. [85] rearrange spectral vectors in horizontal or zig-zag order for cube extraction and apply a Fully Convolutional Spatial Network (FCSN) to label all pixels simultaneously. In the med-

ical field, [87] presents Modulated Gabor CNN (MGCNN), combining Gabor wavelets with CNN kernels for blood cell classification, while [88] developed an LSTM-CNN with VGG-19 for thyroid disease detection. Beyond healthcare, Han et al. [89] employ 2D-CNN with SVM to evaluate Canarium indicum quality via peroxide values (PV). DL models relying only on spatial features face high computational cost and memory demand, especially with high-resolution HSIs [77]. They are also vulnerable to overfitting with limited data, as the absence of spectral information reduces discrimination [82]. Moreover, spatial extraction alone may fail to capture inter-class variability in complex scenes. Future work may emphasize lightweight architectures to lower complexity, while semi-supervised learning can mitigate data scarcity by leveraging unlabeled pixels. Furthermore, attention mechanisms and advanced pooling may enhance spatial feature extraction while maintaining efficiency.

2.5.3 DL Model Leveraging Spectral-Spatial Features

In 3D-HSI data, spectral and spatial dimensions are inherently linked, supporting applications such as land cover mapping, environmental monitoring, and mineral exploration. Yet, many models focus on either spectral or spatial features, limiting comprehensive feature extraction and reducing classification accuracy. To address this, DL models integrating both have shown superior performance. Two key approaches for extracting spectral-spatial features have emerged:

1. Independent Feature Extraction: Spectral and spatial features are extracted separately and later fused within the DL network. This preserves distinct representations while enabling structured integration, thereby improving classification accuracy.

2. Collaborative Feature Extraction: Unlike the independent approach, this method jointly processes spectral and spatial information from 3D-HSI data. Architectures such as 3D Convolutional Neural Networks (3D-CNNs) capture complex spectral–spatial patterns, enhancing robustness and overall performance.

2.5.3.1 Pre-processing based Spectral-Spatial Network

The integration of spectral-spatial setworks extracts spectral and spatial features independently before merging them, ensuring structured fusion that enhances HSI classification by capturing fine details. Table 2.3 presents preprocessing-based spectral–spatial fusion techniques with their advantages, limitations, and performance. For example, [90] proposed a CNN framework combining probabilistic PCA for spectral and Gabor filters for spatial features. Similarly, [91] employed attention modules to

Table 2.3: Overview of HSI Classification Models Utilizing Combined Spectral-Spatial Features with Feature Fusion in the Pre-processing Stage

Model	Advantages	Limitations & Future Scope	Performance
[90]	The approach utilizes hashing techniques to leverage semantic information, while the Gabor filter is used to extract spatial features across multiple scales and directions.	2D-CNN struggles to capture spectral-spatial features with context effectively, and the training and testing runtime is excessively high; parallelization can offer a solution.	OA for IP, PU and SA are 99.02%, 99.94% and 99.94%. 30% of total samples for training.
SSAtt [91]	An attention-driven spectral-spatial CNN enhances discriminative and spatial features, utilizing the differences between spectral and spatial data to improve HSI classification.	The primary limitation of the attention mechanism is its introduction of additional weight parameters, resulting in prolonged training times. Implementing GPU acceleration can expedite the model's training and testing processes.	OA of HU-2013 and HU-2018 are 90.38% and 72.57% with 18.84% and 6.83% of total samples are used for training.
Spectral NET [92]	The model applies wavelet transforms to identify key spectral features, which are more efficient than the calculations required for 3D CNNs.	The complexity of the proposed CNN model raises concerns about potential overfitting. Choosing 3D-CNN over 2D-CNN might yield improved classification accuracy.	OA of IP, PU and SA are 98.76%, 99.71% and 99.96% with 10% of total samples for training.
SA3-DDRN [93]	SA3-DDRN effectively captures spectral and spatial information while maintaining performance, thanks to its residual network design.	AEs may result in the loss of key features by diminishing the height, width, and spectral bands of hyperspectral images. TL can boost training efficiency and enhance classification accuracy.	OA of IP, PU and SA are 98.97%, 99.69% and 79.24% with 10%, 5% and 1% of total samples are used for training respectively.

improve feature focus, while [92] extended 2D-CNNs with wavelet transforms for multi-resolution HSI classification.

However, these methods face challenges: reliance on 2D-CNN limits spectral–spatial correlation, reducing context capture [90]; attention mechanisms improve focus but add parameters, raising computational cost [91]; complex models like SpectralNET risk overfitting on limited data [92]; and lossy approaches such as autoencoders in SA3-DDRN may discard critical information, highlighting the need for transfer learning (TL) to improve efficiency [93].

2.5.3.2 Post-processing based Spectral-Spatial Network

In HSI classification, an effective strategy for integrating spectral and spatial features is independent extraction using distinct DL architectures, where high-level features are merged before classification to improve performance. Table 2.4 outlines post-processing fusion techniques, their advantages, limitations, and accuracy.

In this direction, [94] combined Balanced Local Discriminant Embedding (BLDE) for spectral features with a 2D-CNN for spatial features, followed by logistic regression (LR) for classification. Similarly, [95] developed a five-layer 3D-CNN optimized for GPU implementation, while [96] proposed LBP-DC-CNN, where 1D-CNNs extract spectral features and local binary patterns (LBP), fused via a softmax classifier. Furthermore, [97]

Table 2.4: Overview of HSI Classification Models Utilizing Combined Spectral-Spatial Features with Feature Fusion in the Post-processing Stage.

Model	Advantages	Limitations & Future Scope	Performance
SSFC [94]	The BLDE balancing method addresses the singularity issue, incorporates data diversity, and enhances classification accuracy.	The proposed method needs a strategy or optimization function to determine the size of training samples, a critical aspect of DL.	OA for PU and PC are 96.98% and 99.87%. 40 samples per-class for training.
[95]	The model will apply 3D kernels to learn the local signal change in the spatial and spectral domains of HSIs, which provide more discriminant spectral-spatial features.	3D convolutions introduce a substantial parameter increase, which can result in overfitting. A suitable approach is to explore semi-supervised 3D-CNN models, which are especially beneficial for the largely unlabeled samples in HSI data.	OA for IP and PU are 98.37% and 97.86% with spatial input sizes as 29×29 and 27×27 with 200 samples per-class for training.
LBP-DC-CNN [96]	The proposed model yields a less noisy classification map compared to the dual-channel CNN or the pre-processed LBP feature with 2D-CNN.	LBP-DC-CNN processes one-dimensional data, leading to larger LBP features and longer training times, which may cause overfitting. Using 3D-CNN can yield more discriminative features than a 1D-CNN.	OA for IP, PU and SA are 98.57%, 99.54%, and 99.55%. 200 samples per-class for training.
2D-3D-D [97]	This model extracts rich spectral features from the available HSI, and subsequently, the 3D block refines these features by incorporating information from neighboring bands	Optimizing the number of 3D-convolution layers is essential to avoid overfitting due to a high parameter-to-sample ratio. Starting with a 2D-CNN limits the model's ability to capture detailed features.	OA for PU, SA, and KSC are 99.54%, 99.88%, and 99.47% with 5%, 5%, and 10% of total samples for training.
DcCaps-GAN [98]	Models leverage CapsNet and GAN to analyze spectral-spatial correlations, generate pseudo-spectral data, and augment samples, with CapsNet enhancing GAN discriminator stability.	Employing separate spectral and spatial extraction can lead to overfitting and increased computation times. The capsule structure adds to these costs, but TL can reduce training time and boost classification accuracy.	OA for PU, SA, and KSC are 97.98%, 93.87%, and 98.20% with 40, 20, and 25 samples per-class used for training, respectively.

introduced a hybrid 2D-3D-D network that captures dominant spectral–spatial correlations, and [98] designed DcCapsGAN, integrating Capsule Networks (CapsNet) with GAN to mitigate mode collapse and gradient vanishing while reducing parameters through octave and multiscale convolutions.

However, post-processing fusion faces challenges like feature redundancy, which reduces efficiency [97], and reliance on high-dimensional fused features increases overfitting with limited data [96]. Therefore, future work should prioritize lightweight designs, adaptive fusion, and semi-supervised learning for better efficiency and generalization [98].

2.5.3.3 Correlated Integrated Spectral-Spatial Network

Unlike earlier methods relying on indirect spectral–spatial extraction, recent models adopt 3D convolution for joint hierarchical learning, achieving higher accuracy [82]. Table 2.5 outlines the strengths, limitations, and performance of DL models based on integrated spectral–spatial features. Given the 3D nature of HSI data, 3D-CNNs are inherently more effective. To improve parameter efficiency, [99] proposed Dimension Variation-CNN (DV-CNN), combining 3D-, 2D-, and 1D-CNNs. Moreover, [100] in-

troduced a fast 3D-CNN that segments HSI cubes into overlapping patches to enhance spatial feature learning and classification precision. Furthermore, [101] integrated 3D- and 2D-CNNs with the Mish activation function to improve gradient propagation and avoid saturation. [102] combined 3D-CNNs and 2D-CNNs with a non-local block for refined spatial correlations and hierarchical feature fusion, strengthening discrimination.

Additionally, [103] proposed a 3D fast-learning CNN with a dimension-reduction block and 2D-CNN, improving spectral–spatial extraction and efficiency. In contrast, [93] introduced SAE-3DDRN, integrating a convolutional SAE for dimensionality reduction with 3D-CNN and residual networks for better classification. Similarly, [104] developed a 3D-GAN framework that combines 3D-CNN with ResNet in a semi-supervised setting, refining feature extraction and outperforming 3D-ResNet using both labeled and unlabeled data.

However, CNN-based models face challenges with uniform input requirements and local feature limitations. On the other hand, GCN-based models handle irregular inputs and capture both local and global relations. In this direction, [105] proposed a dynamic GCN with multi-scale graph convolution to enhance spatial representation, while [106] introduced a dual-channel hybrid GCN integrating miniGCN with CNN for efficient mini-batch training. Further, [107] presented Deep-Hyper, a 3D-CNN with a 3D-attention module for white blood cell (WBC) classification in microscopy HSI, built on ResNeXt for effective feature extraction. Similarly, [108] proposed the Dimension-Driven Multi-Path Attention Residual Network (DDMARN) for pixel-level classification in membranous nephropathy (MN). In food quality control, [109] developed a 3D-CNN with multi-head attention for pine nut defect detection, employing Generalized Gradient-Weighted Class Activation Mapping (Grad-CAM++) to highlight critical wavelengths and regions.

Overall, integrated spectral–spatial DL models, particularly 3D-CNNs, improve classification by jointly capturing hierarchical features. Hybrid approaches like DV-CNN enhance parameter efficiency, while techniques such as Mish activation and non-local blocks [102] refine feature learning. Nevertheless, computational complexity, overfitting, and CNNs' inability to model sequential dependencies remain challenges. ViTs and CNN–ViT hybrid models represent promising directions for adaptive feature extraction and efficient HSI processing.

Table 2.5: Overview of HSI Classification Models Utilizing Spectral-Spatial Features with Feature Fusion in the Correlated Integrated Stage.

Model	Advantages	Limitations & Future Scope	Performance
R-3D-CNN [82]	The model progressively reduces patch sizes, causing the final classification to rely on pixel-level information rather than patch-level detail.	R-3D-CNN employs multiple 3D-CNN layers, resulting in high complexity and potential overfitting due to numerous parameters. Optimizing network depth is essential for balanced and effective training.	OA for IP, PU, PC, SA, KSC and BS are 99.50%, 99.97%, 96.79%, 99.80%, 99.85% and 99.38%.
SA3-DDRN [93]	Model effectively extracts spectral-spatial information without experiencing any performance decrease due to the inclusion of residual network.	Autoencoders, being lossy, may lose essential features during dimension reduction. Additionally, they reduce not only the bands of the image but also their spatial size or resolution.	OA for IP, PU and SA are 98.97%, 99.69% and 99.24% with 10%, 5% and 1% of total samples for training.
DV-CNN [99]	By optimizing the dimensionality of feature maps, the deep network effectively enhances the classification accuracy of limited-labeled HSI data.	The wider model introduces a higher level of complexity, which can lead to overfitting. Utilizing data augmentation is crucial for broadening class samples and improving the performance of the model.	OA for IP and PU are 87.60% and 98.28%. 10% of total samples are used for training.
[100]	The model utilizes a 3D convolutional kernel to generate feature maps, capturing spectral-spatial dependencies across adjacent bands.	While 3D convolutions are costly and prone to overfitting, Dilated convolutions can be helpful as they reduce output size and parameters in the next layer and lower overfitting risk.	OA for IP, PU and SA are 97.75%, 98.40% and 98.06%. 42% of total samples are used for training.
[101]	The Mish activation function improves gradient flow over ReLU, prevents saturation, and handles negative values better, enhancing model performance.	Excessive parameters in FC layers hinder computational efficiency. Incorporating data augmentation techniques using GANs can diversify class samples and improve accuracy.	OA for IP, PU SA and BS are 96.07%, 99.52%, 99.51% and 96.44%. 5% of total samples for training.
CACNN [102]	The NonLocalBlock enhances spatial correlations, the Conv-block captures abstract features, and hierarchical fusion ensures effective extraction.	The increased model complexity raises risk of overfitting, especially with limited, similar data types. Future efforts should aim to create fusion strategies that improve generalization and reduce overfitting risks.	OA for IP, PU, SA, and HU are 97.38%, 99.17%, 98.55%, and 87.89%. 200 samples per class for training.
3D-2D-CNN [103]	3D depth-wise separable and factorized convolution reduce computation and model size, improving efficiency.	Excessive utilization of 3D convolutional layers can escalate model complexity, increasing the risk of overfitting.	OA for IP, PU, and SA is 97.14%, 98.90%, and 99.07%.
3D-GAN [104]	Adversarial training enhances robustness with limited data by leveraging generator-discriminator interplay, which produces synthetic samples.	GAN training is challenging due to the need for diverse data and high parameter counts from CNNs, increasing complexity and slowing performance.	OA for IP, SA and KSC are 89.09%, 93.02% and 96.89%. 200 samples per-class for training.
MDGCN [105]	The graph is dynamically updated throughout the convolutional process, reducing the model's computational time.	Segmentation influences feature determinism, potentially affecting accuracy. Minimizing the size of the adjacency matrix helps streamline computations.	OA for IP, PU and KSC are 93.41%, 95.68% and 99.79%.
FuNet-C [106]	The miniGCN enables GCN training in a mini-batch mode, significantly lowering the computational demand for adjacency matrix formation.	Softmax function generates pixel class probability vectors from extracted features but falls short in achieving intra-class compactness and discriminative representations, limiting HSI classification performance.	OA for IP, PU and BS are 79.89%, 92.20% and 87.39% with 10%, 10% and 20% of total samples for training.
Deep-hyper [107]	Group and depth-wise convolutions reduce the complexity of the model, and the attention enhances feature extraction.	Limited spectral features hinder adaptability to diverse WBC types. Incorporate more discriminative spectral features for improved versatility.	OA for WBC data ($512 \times 512 \times 16$) is 97.72%. 70%, of the samples for training.

Table 2.5: Continued: Feature Fusion in the Correlated Integrated Stage.

Model	Advantages	Limitations & Future Scope	Performance
DDMA-RN [108]	Multiscale features capture complex patterns, and DDMARN enhances depth features through channel attention to process different information levels.	The pixel-level classification approach ignores spatial context, limiting its ability to capture complete MN lesion features. Multiscale feature extraction reduces decision-making clarity.	OA for MN data is 96.22%, with 80%, of the total samples randomly selected for training.
FX-3D-CNN [109]	The Grad-CAM++ method visualizes key spectral ranges and highlighted pixel regions, demonstrating the effectiveness of CNN in defect identification.	Leveraging transfer and incremental learning in deep learning models can expand their applicability to new defect types, improving practical use for pine nut quality assessment.	OA for FX10 ($72 \times 72 \times 180$) and FX17 ($72 \times 72 \times 180$) data are 73.63% and 78.29%.
HyBrid-SN [110]	HybridSN synergizes spatial-spectral and purely spatial features by integrating 3D and 2D convolutional operations.	The unoptimized configuration of 3D convolutional layers impacts training efficiency. Techniques such as mini-batch normalization can mitigate overfitting.	OA for IP, PU and SA are 98.39%, 99.72% and 99.98%. Training with 10% samples.
ADGAN [111]	Adaptive DropBlock (AdapDrop) is used in the generator and discriminator to prevent mode collapse by creating flexible drop masks suited for diverse ground object shapes.	Training GANs is challenging due to the need for continuous data diversity, complicating evaluation. Combining operations in one step raises computational demands, prolonging training and testing durations.	OA for IP, PU, and BS are 97.23%, 95.38%, and 95.89%, with 1000, 1000, and 300 of total samples selected for training.
SS-DCGAN [112]	The model uses 3D DL to retain spectral-spatial properties, with a semi-supervised approach enhancing accuracy through unlabeled and generated samples.	Imbalanced training data affects performance of the model. Explore algorithms to reduce computational costs while improving the quality of augmented samples.	OA for IP, PU, and SA are 99.25%, 99.48%, and 99.52%, 150 samples per class for training.

2.6 Categorization Based on Contextual Scope

HSI classification assigns labels to each pixel using spectral and spatial information. While traditional methods emphasize spectral, spatial, or spectral–spatial features, recent advances focus on contextual features, capturing both local and global dependencies for improved accuracy and robustness. Local features highlight fine-grained variations, whereas global features capture long-range relationships, together enhancing scalability and efficiency. This section reviews models leveraging local, global, and combined contextual features, outlining their strengths, limitations, and computational trade-offs.

1. **Local Contextual Feature Models:** Capture fine-grained spectral–spatial variations to improve classification.
2. **Global Contextual Feature Models:** Learn long-range dependencies, strengthening accuracy and robustness.
3. **Combined Local–Global Contextual Feature Models:** Fuse fine-grained and long-range features for balanced performance.
4. **Challenges and Optimization:** Address sequential data processing, computational costs, and long-term dependencies.

By analyzing these models, this study highlights their advantages and drawbacks, guid-

ing future research aimed at improving HSI classification performance.

2.6.1 Local Contextual Features-Based Models

Local contextual features (Lo-CF) capture fine spectral–spatial details within localized regions, enabling precise classification. These features are extracted through convolutional operations that emphasize neighboring pixels and local structures. However, Lo-CF struggles with broader patterns, limiting performance in complex scenes with large or spatially distributed objects. Table 2.6 lists existing Lo-CF models, their advantages, limitations, and performance.

[113] introduced the Spectral–Spatial Residual Network (SSRN), an end-to-end model processing raw 3D HSI cubes without feature engineering. SSRN employs spectral and spatial residual blocks with identity mapping for efficient backpropagation, while batch normalization improves regularization and accuracy. Moreover, [114] proposed the Spectral–Spatial Fully Convolutional Network (SSFCN), a pixel-to-pixel classifier that avoids redundant patch computations, adaptively weights spectral and spatial features, and employs a mask matrix for sparsity, with dense Conditional Random Fields (CRF) enhancing local–global balance. Further, [115] introduced the Spatial Manifold Representation Network (SMRN), which leverages a Graph Convolutional Network (GCN) for feature extraction and local label propagation within a semi-supervised framework, improving performance in sparse and noisy datasets. Finally, [116] developed the Spectral–Spatial Self-Attention Network (SSSAN), which employs spatial and spectral self-attention with score-weighted fusion to refine classification by capturing both contextual information and spectral correlations.

Advantages: Lo-CF models excel in fine-grained classification by detecting subtle spectral–spatial variations. They are computationally efficient, suitable for resource-constrained systems, and reduce complexity by processing smaller patches, making them practical for real-time applications.

Limitations and Future Directions: Lo-CF models cannot capture long-range dependencies [115], reducing effectiveness for large or spatially dispersed objects. Their sensitivity to noise and small-scale variations [114] can also degrade performance in complex datasets. Future research should explore hybrid designs that integrate global contextual features. For example, the Spectral–Spatial Self-Attention Network (SSAN) [116] enhances performance by combining local and long-range dependencies.

Table 2.6: Overview of HSI Classification Models Using Local Contextual Feature.

Model	Advantages	Limitations & Future Scope	Performance
SSRN [113]	The use of identity mapping in the residual blocks facilitates efficient gradient back-propagation, counteracting accuracy degradation observed in other deep learning models.	Focusing on small-scale regions limits the model's ability to capture long-range dependencies in HSI data. While efficient for smaller regions, 3-D convolutions demand high memory for large datasets and risk overfitting with limited labeled data.	OA for IP, PU and KSC are 99.19%, 99.61% and 99.79%. 20%, 10% and 10% of samples of IP, PU and KSC for training.
SSFCN-CRF [114]	SSFCN improves efficiency by eliminating patch-wise redundancy and adaptively balances spectral-spatial features.	The model's reliance on CRF increases computational overhead and sparse training data limits performance scalability.	OA for PU, HU and SA are 98.11%, 95.51% and 98.48%.
SMRN [115]	The model embeds latent spatial manifold structures for feature extraction, improving accuracy. Its semi-supervised approach utilizes both labeled and unlabeled samples.	The GCN introduces computational overhead, and reliance on spatial manifold representation may hinder scalability for large datasets. Future work should optimize computational efficiency for better generalization.	OA for IP and PU are 99.26%, and 99.50%. 200 samples per class used for training. training.
SSSAN [116]	SSSAN improves local and long-range dependencies for enhanced spectral-spatial feature representation. Modular design allows easy integration into CNN architectures with minimal computational impact.	SSSAN's self-attention mechanisms can increase computational complexity with larger HSI datasets. Future research could optimize these mechanisms for efficiency and explore TL to broaden its applicability.	OA for PU, HU and SA are 98.11%, 95.51% and 98.48%. 150 samples per class used for training.

2.6.2 Global Contextual Features-Based Models

Global contextual features (Go-CF) capture relationships across the entire HSI, emphasizing long-range dependencies and large-scale structures. They are crucial for classifying regions that may be spectrally similar yet spatially distant. Models such as LSTMs capture sequential spectral dependencies, while GCNs and ViTs effectively represent global context. Table 2.7 summarizes existing Go-CF models, their strengths, limitations, and performance.

Firstly, [117] proposed the Spectral–Spatial LSTM (SSLSTM), which employs spectral-LSTM (SeLSTM) for spectral dependencies and spatial-LSTM (SaLSTM) for spatial features. PCA extracts spatial features, while decision fusion integrates spectral and spatial outputs for improved classification. GCNs model HSI as graphs, where nodes represent pixels or patches and edges define spectral or spatial relations. For instance, [118] developed a nonlocal GCN that constructs HSI as a nonlocal graph with pixels as vertices, using semi-supervised training for classification.

In contrast, ViTs capture global context through self-attention by treating HSI patches as sequences. [119] introduced SpectralFormer, a transformer framework that generates group-wise spectral embeddings and employs cross-layer skip connections to prevent information loss. Similarly, [120] proposed CASST, a dual-branch spatial–spectral transformer that integrates cross-attention to improve consistency and efficiency.

Table 2.7: Overview of HSI Classification Models Using Global Contextual Feature

Model	Advantages	Limitations & Future Scope	Performance
SSLS-TM [117]	The decision fusion strategy enhances robustness and ensures balanced contributions from spectral and spatial domains.	The sequential LSTM operations increase complexity for large datasets. Future work can focus on lightweight architectures and optimized fusion for better scalability.	OA for IP, PU and KSC are 95%, 98.48% and 97.89%.
Non-Local GCN [118]	The nonlocal graph representation improves classification by enhancing reasoning across images, resulting in finer boundaries and reduced noise.	The nonlocal graphs are computationally complex and may struggle with noisy or imbalanced data. Future research should focus on optimizing graph construction and developing scalable architectures.	OA for IP, PU and SA are 87.92%, 90.04% and 92.48%.
Spectral Former [119]	Model excels in capturing spectral sequence attributes and holding vital information across layers, improving accuracy.	The model's high complexity and large dataset requirements limit its use in constrained settings. Future work can optimize efficiency and broaden adaptability.	OA for IP, PU and SA are 81.76%, 91.07% and 88.01%. 10% of samples for training.
CASST [120]	The model excels at spatial-spectral feature fusion, improving classification accuracy. Its attention mechanism reduces computational burden while enhancing feature consistency.	A limitation of the CASST is its reliance on global contextual features, which may overlook finer local patterns within the HSI data. Future work could focus on optimizing the attention mechanism and incorporating self-supervised learning.	OA for IP, PU and SA are 87.92%, 90.04% and 92.48%.

Limitations and Future Directions: Despite their strengths, Go-CF models face challenges such as high computational cost, risk of overgeneralization, and difficulty in capturing fine-grained spatial details [117]. Overemphasis on large-scale patterns can reduce pixel-level discrimination [119]. Future research should explore hybrid frameworks that integrate both local and global features for robust HSI classification.

2.6.3 Combined Local-Global Contextual Feature-Based Models

Combined local–global contextual feature (Lo–Go-CF) models integrate fine-grained local details with long-range dependencies, enhancing classification accuracy and robustness. By addressing the limitations of single-context approaches, these hybrid models adapt better across diverse tasks. Table 2.8 summarizes existing models, their strengths, limitations, and performance. [106] proposed Fu-NetC, an enhanced miniGCN framework that integrates CNN and GCN through minibatch training. It extracts complementary spectral–spatial and graph-based features, improving computational efficiency and fusion strategies. Similarly, [121] introduced EMS-GCN, which refines superpixel boundaries using a learnable segmentation module and mixhop convolution, adaptively capturing spectral–spatial representations. RNN- and transformer-based frameworks have further advanced hybrid modeling. For instance, [122] developed an attention-based Bi-LSTM that strengthens spectral correlation modeling by integrating spatial–spectral attention. Likewise, [123] proposed Interactformer, a transformer–CNN hybrid that fuses local and global features for HSI super-resolution. Expanding on this, [124] introduced SSFTT, combining CNN and transformers for hierarchical feature extraction

and efficient classification.

Furthermore, ViT-based models have also evolved to overcome CNN and transformer limitations. [125] proposed LGSA-ViT, which employs a hybrid spatial–spectral tokenizer and Gaussian position bias in a lightweight attention mechanism. [126] presented CAF-Former, bridging CNN and ViT through multiscale CNN-based local feature extraction and Gaussian Transformer-based global context modeling. Finally, [127] introduced LiT, which employs lightweight self-attention with convolutional tokenization, supported by stratified sampling for efficient training and improved generalization.

Table 2.8: HSI Classification Using Local-Global Contextual Feature Based Model

Model	Advantages	Limitations & Future Scope	Performance
EMS-GCN [121]	The learning-based superpixel module refines boundaries, preserves edges, and dynamically updates graphs for better feature representation.	The reliance on superpixel segmentation may struggle with irregular or nonhomogeneous regions in HSI data. Integrating imbalance-aware loss functions to address class imbalance issues effectively.	OA for IP, PU, and HU are 95.87%, 98.47%, and 88.57%. Trains on 30, 30, and 200 samples per class for IP, PU, and HU.
Bi-LSTM [122]	Bi-LSTM captures bidirectional spectral correlations, enhancing feature representation with spatial-spectral attention while minimizing redundancy.	The bidirectional structure and attention mechanisms increase computational cost. Future work should focus on parallel processing or lightweight attention for efficiency.	OA for SA, PU and PC are 98.88%, 97.63% and 99.20%. 10% of total samples are used for training.
Interact-former [123]	The separable self-attention ensures linear complexity, reducing memory overhead while maintaining global context.	The reliance on predefined interaction mechanisms in the interactive attention unit (IAU) might limit its adaptability to diverse HSI datasets.	OA for PC are 95% with only 1% of total samples are used for training.
SSFTT [124]	Hierarchical CNN extracts shallow spatial-spectral features, while a Gaussian-weighted tokenization module enhances semantic representation.	SSFTT reliance on Gaussian-weighted tokenization may limit generalizability to data with distinct spectral distributions. Future work could involve integrating adaptive weighting mechanisms and exploring domain-specific tokenization.	OA for IP, PU, and HU are 97.47%, 99.21%, and 98.92%. Trains on 10%, 10%, and 5% of samples for IP, HU, and PU.
LGSA-ViT [125]	Utilizing Q,X,X instead of Q,K, V reduces computations and parameters. Gaussian position bias enhances feature weighting in the central query block.	Hybrid spatial-spectral tokenizer adds complexity, potentially limiting efficiency for ultra-large HSIs. Future work should explore alternative tokenization strategies to optimize computational demands.	OA for IP, PU, HU, and SA are 98.85%, 99.88%, 98.50% and 99.93%. Trains on 10% of total samples.
CAF-Former [126]	Dynamic-CNN improves multiscale feature extraction with adjustable expansion rates and kernel sizes. CAF enhances local-global feature interaction for better spatial-spectral representation.	Parallel CNN-Transformer branches may increase training and inference times, with CAF adding computational overhead that affects scalability. Future work should explore lightweight CAF-Former variants to mitigate these constraints.	OA for YRE, HR-L, and WHU-HC are 98.20%, 83.08%, and 96.26%, using 1% of total samples for YRE and WHU-HC, and 3% for HR-L.
LiT [127]	Employs channel and position lightweight self-attention modules to reduce memory and computation, while controlled multiclass stratified sampling mitigates overfitting and enhances generalization.	Convolutional blocks might miss fine-grained multiscale details compared to hybrid tokenizers. ViTs risk overfitting with limited data, necessitating additional regularization. Hybrid tokenizers or dynamic convolutional layers can improve local feature learning.	OA for IP, SA, and DFC-2018 are 86.31%, 89.03%, and 85.71%, using 20%, 4%, and 7% of total samples for training.

Limitations and Future Directions: Despite their effectiveness, Lo–Go-CF models face challenges such as high memory consumption and training complexity, particularly in transformer-based frameworks. Future work should emphasize lightweight hybrid ar-

chitectures, adaptive fusion strategies, and semi-supervised learning to balance accuracy with efficiency.

2.7 Research Gaps and Objectives

The comprehensive survey on HSI classification using deep learning highlights significant advancements in fine-grained feature-based networks and global contextual architectures that capture long-term dependencies. Traditional methods primarily focus on spectral feature extraction, whereas modern hybrid models integrate both spectral and spatial information to enhance classification accuracy. Additionally, recent approaches leverage sequential data modeling to improve temporal and spatial coherence, enabling more robust global feature learning. However, several challenges remain, including inefficient global feature representation, high computational complexity, and difficulties in effectively balancing local feature extraction with global contextual modeling and long-term dependencies. Table 2.9 presents a comparative analysis of various HSI classification models across benchmark datasets. CNN-based models, such as MGCNN [87], excel at extracting local spatial features but struggle with long-range dependencies. Transformer-based architectures, including SSFTT [119] and Interactformer [123], mitigate this issue by modeling global context but introduce higher computational costs. Hybrid models, such as LSGA-ViT [125], CAF-Former [126], and LiT [127], attempt to optimize both aspects but still face challenges in spectral-spatial feature fusion. The proposed model aims to address these gaps by enhancing feature extraction efficiency while maintaining high classification accuracy across diverse HSI datasets.

2.7.1 Research Gaps:

1. **Limited Spectral–Spatial Feature Integration:** Existing models often emphasize either spectral information (e.g., 1D-CNN [72]) or spatial context (e.g., 2D-CNN [82]), leading to incomplete joint representation. Since HSI data inherently combine spectral signatures and spatial correlations, neglecting either aspect weakens classification performance.
2. **Inability to Capture Global Context:** CNN-based models such as MGCNN [87] capture only local dependencies and fail to model global relationships across the HSI scene. This limits their ability to classify classes with large or complex spatial structures.
3. **High Computational Overhead of Transformers:** Transformers such as SSFTT [119] and Interactformer [123] achieve strong global modeling but suffer

Table 2.9: HSI classification related DL models that consist of lists model name, year, datasets, and whether they incorporate spectral, spatial, and integrated features via pre, post, or integration (Ingrtd) processing, along with Lo-CF and Go-CF.

Model	Year	Datasets	Spectral	Spatial	Pre	Post	Ingrtd	Lo-CF	Go-CF
1D-CNN [72]	2015	IP, PU, SA	✓	×	×	×	×	✓	×
CNN-PPF [73]	2016	IP, PU, SA	✓	×	×	×	×	✓	×
1D-B-CNN [74]	2018	IP, KSC, SA	✓	×	×	×	×	✓	×
RNN-GRU [75]	2017	IP, HU, PU	✓	×	×	×	×	×	✓
CRNN [76]	2017	IP, HU	✓	×	×	×	×	✓	✓
DEEP0 [77]	2016	IP, PU	×	✓	×	×	×	✓	×
AI-NET [81]	2018	IP, SA	×	✓	×	×	×	✓	×
R-2D-CNN [82]	2018	PU,SA,PC,BS	×	✓	×	×	×	✓	×
DHNN [83]	2019	PU, SA	×	✓	×	×	×	✓	×
CNN-AL-MRF [84]	2020	IP, Pu, PC	×	✓	×	×	×	✓	×
FCSN [85]	2021	IP, PU	×	✓	×	×	×	✓	×
HDCFE-Net [86]	2021	IP, SA	×	✓	×	×	×	✓	×
MGCNN [87]	2019	–	×	✓	×	×	×	✓	×
LSTM-CNN [88]	2023	DDTI	×	✓	×	×	×	✓	✓
CNN-SVM [89]	2021	C.indium	×	✓	×	×	×	✓	×
[90]	2020	IP, PU, SA	✓	✓	✓	×	×	✓	×
SSAtt [91]	2020	HU	✓	✓	✓	×	×	✓	×
Spectral NET [92]	2021	IP, PU, SA	✓	✓	✓	×	×	✓	×
SA3-DDRN [93]	2021	IP, PU, SA	✓	✓	✓	×	×	✓	✓
SSFC [94]	2016	PU, PC	✓	✓	×	✓	×	✓	×
[95]	2018	IP, PU	✓	✓	×	✓	×	✓	×
LBP-DC-CNN [96]	2019	IP, PU, SA	✓	✓	×	✓	×	✓	×
2D-3D-D [97]	2020	PU, SA, KSC	✓	✓	×	✓	×	✓	×
DcCaps-GAN [98]	2021	PU, SA, KSC	✓	✓	×	✓	×	✓	✓
R-3D-CNN [82]	2018	PU,PC,KSC,BS	✓	✓	×	×	✓	✓	×
DV-CNN [99]	2018	IP, PU	✓	✓	×	×	✓	✓	×
[101]	2020	IP, PU, SA, BS	✓	✓	×	×	✓	✓	×
CACNN [102]	2020	IP, PU, SA, HU	✓	✓	×	×	✓	✓	✓
3D-2D CNN [103]	2021	IP, PU, SA	✓	✓	×	×	✓	✓	×
3D-GAN [104]	2018	IP, PU, KSC	✓	✓	×	×	✓	✓	×
MDGCN [105]	2019	IP, PU, KSC	✓	✓	×	×	✓	✓	✓
FuNet-C [106]	2020	IP, PU, BS	✓	✓	×	×	✓	✓	✓
Deep-hyper [107]	2021	WBC-3D	✓	✓	×	×	✓	✓	×
DDMA-RN [108]	2023	MN-HSI	✓	✓	×	×	✓	✓	×
FX-3D-CNN [109]	2024	FX10, FX17	✓	✓	×	×	✓	✓	×
HyBrid-SN [110]	2019	IP, PU, SA	✓	✓	×	×	✓	✓	×
ADGAN [111]	2020	IP, PU, SA	✓	✓	×	×	✓	✓	×
SS-DCGAN [112]	2020	IP, PU, SA	✓	✓	×	×	✓	✓	×
SSRN [113]	2017	IP, PU, KSC	✓	✓	×	×	✓	✓	×
SSFCN-CRF [114]	2019	PU, HU, SA	✓	✓	×	×	✓	✓	×
SMRN [115]	2021	IP, PU	✓	✓	×	×	✓	✓	×
SSSAN [116]	2022	PU, HU, SA	✓	✓	×	×	✓	✓	×
SSLS-TM [117]	2019	IP, PU, KSC	✓	✓	×	×	✓	×	✓
Non-Local GCN [118]	2020	IP, PU, SA	✓	×	×	×	×	×	✓
Spectral Former [119]	2021	IP, PU, HU	✓	×	×	×	×	×	✓
CASST [120]	2022	IP, PU, SA	✓	✓	×	×	✓	×	✓
EMS-GCN [121]	2022	IP, PU, HU	✓	×	×	×	×	✓	✓
Bi-LSTM [122]	2021	SA, PU, PC	✓	×	×	×	×	✓	✓
Interactformer [123]	2022	PC	✓	✓	×	×	✓	✓	✓
SSFTT [124]	2022	IP, PU, HU	✓	✓	×	×	✓	✓	✓
LGSA-ViT [125]	2023	IP,PU,HU,SA	✓	✓	×	×	✓	✓	✓
CAF-Former [126]	2024	YRE,HR-L,HC	✓	✓	×	×	✓	✓	✓
LiT [127]	2023	SA,DFC-2018	✓	✓	×	×	✓	✓	✓

from quadratic complexity in self-attention, making them costly for large-scale or real-time applications.

4. **Weak Discrimination of Spectrally Similar Classes:** Many approaches misclassify spectrally overlapping classes, such as crop varieties or urban materials, due to insufficient feature separability. This problem is critical in applications like agriculture and urban monitoring where fine-grained discrimination is needed.
5. **Class Imbalance Problem:** HSI datasets are often imbalanced, with majority classes dominating over minority ones. This biases models toward common land-cover types while reducing recognition of rare categories. Even hybrid CNN–ViT models such as LSGA-ViT [125], CAF-Former [126], and LiT [127] still struggle with this issue.

2.7.2 Research Objectives:

1. **Enhance Spectral-Spatial Discrimination:** Implement adaptive feature fusion techniques to improve classification accuracy in HSI datasets.
2. **Develop an Efficient Hybrid Model:** Integrate CNN and ViT-based approaches in a balanced manner to enhance both local-global feature extraction.
3. **Reduce Computational Complexity:** Improve model efficiency by incorporating group convolution and an efficient attention mechanism.
4. **Address Class Imbalance:** Incorporate loss balancing strategies to improve model performance on hard-to-classify classes with limited representation.

The survey of existing HSI classification methods reveals four major gaps that guide this thesis. First, CNN-based models overemphasize local spectral–spatial features, which requires high computation and reduces scalability. Second, hybrid CNN–ViT approaches, though promising, are still computationally heavy and lack lightweight designs for efficient local–global feature extraction. Third, Transformer-based models generally suffer from quadratic complexity, limiting their deployment on large-scale hyperspectral datasets. Finally, class imbalance remains a critical challenge, leading to poor recognition of minority and hard-to-classify classes. To address these issues, the thesis advances through four contributions, which are:

- Chapter 3 introduces MDCNN, which strengthens local feature learning by incorporating morphological preprocessing to extract more discriminative spatial features and morphological dilation to reduce parameters—effectively acting as convolution plus pooling.
- Chapter 4 presents LogGroupFormer, a CNN–ViT hybrid where CNN captures local patterns and ViT models global sequential dependencies; to improve ef-

ficiency, the CNN branch employs logarithmic convolution, which significantly reduces parameters and FLOPs.

- Chapter 5 develops CKGFLNet, which tackles the quadratic time complexity of Transformers, reducing it from $\mathcal{O}(N^2d)$ to $\mathcal{O}(Nd^d)$ through linear attention mechanisms and Gaussian–Kaiming initialization, thereby improving scalability on large HSI datasets.
- Chapter 6 proposes HieraKGTNet, which integrates CNN and ViT for hierarchical local–global modeling, leverages LGASS pre-processing to enhance feature quality, applies linear multi-head attention to maintain efficiency, and adopts multiclass poly-focal loss to mitigate class imbalance and strengthen recognition of minority classes.

Together, these contributions form a coherent progression toward accurate, efficient, and robust HSI classification.

2.8 Chapter Summary

This chapter reviewed HSI classification methods, which highlight the shift from traditional ML to DL. Section 2.1 introduced HSI, which includes its applications, key challenges, and feature extraction techniques. Section 2.2 covered classical ML approaches, which include SVM, KNN, K-means, RF, and PLS-DA. Their limitations were analyzed in Section 2.2.6. Section 2.3 discussed DL-based models, which include SAE, DBN, CNN, RNN, LSTM, ViT, GAN, and GCN. Section 2.4 categorized DL models. Sections 2.5 and 2.6 classified ML and DL models based on feature types (spectral, spatial, spectral-spatial) and contextual scope (local, global, hybrid). Section 2.7 outlined key research gaps. Major challenges include spectral-spatial feature integration, long-range dependency modeling, and computational efficiency. CNNs, which capture local spatial patterns, struggle with global contextual learning. RNNs, which model sequential spectral dependencies, face similar limitations. ViTs, which address this issue, introduce high computational costs. Moreover, distinguishing spectrally similar classes is also remains difficult in many scenarios.

To overcome these issues, the goal is to develop a hybrid model, which balances local and global feature extraction. It should improve efficiency with group convolutions and attention mechanisms, which enhance spectral-spatial discrimination using adaptive feature fusion. Ensuring model generalization across benchmark datasets will be essential for validation.