Chapter 3

Leveraging Spectral and Spatial Information in HSI classification

3.1 Introduction

HS captures a wide range of wavelengths across the electromagnetic spectrum. It provides vast amounts of data. As we know, standard RGB images capture only three color channels: red, green, and blue. Similarly, multispectral images consist of a limited number of spectral bands. In contrast, HSI contains hundreds of spectral bands. Each band corresponds to a distinct range of wavelengths. This spectral richness makes HS particularly valuable for classification tasks such as land cover classification, vegetation analysis, and environmental monitoring. By analyzing the detailed spectral cues, researchers can differentiate materials with high precision, even those that appear similar in standard images. However, the high-dimensional nature of HSI poses a significant challenge in image classification. To effectively interpret HSI data, two primary types of cues are crucial:

Spectral Cue: This pertains to the unique reflectance values captured in each spectral band of an HSI. These reflectance values help identify materials based on their distinct spectral signatures, which are recorded reflected light patterns from objects such as vegetation, soil, or human-made structures across wavelengths. Spectral data is crucial for distinguishing land cover types that may look similar spatially but differ in spectral reflectance properties. It plays a key role in identifying materials and their properties, even when spatial features are not enough to differentiate them.

Spatial Cue: While spectral information provides valuable insights into the material properties of the scene, spatial information highlights the arrangement and relationships between pixels within an HSI. Spatial features refer to the patterns, textures, and

geometric shapes formed by object types in the spatial domain of the HSI. For example, forests may exhibit distinct spatial patterns with textures distinguishable from urban areas, even when both share similar spectral signatures. Spatial information is crucial for providing context to spectral information, helping to resolve ambiguities, and improving the classification of objects based on their spatial organization.

Current spectral-spatial models face significant limitations, particularly their reliance on large datasets and high computational costs. These challenges have driven us to develop a Morphologically Dilated Convolutional Neural Network (MDCNN) to enhance efficiency and improve performance. Our approach aims to streamline computation while effectively addressing these persistent issues. The primary contributions of this chapter are summarized as follows:

1. **Morphological Feature Integration:** Mathematical morphological operations are applied to extract discriminative spatial features from HSI. These features are then concatenated with the original HSI data and fed into DL framework. This integration enhances spatial feature representation while reducing computational overhead. Since morphological operations pre-compute spatial patterns, they help the CNN to focus on learning higher-level representations more efficiently.
2. **Hybrid Convolutional Architecture:** The proposed DL framework has integrated both 3D and 2D convolutional layers that combine traditional and dilated convolutions. This design has expanded the receptive field, reduced trainable parameters, and mitigated overfitting risks. Moreover, the use of dilated convolutions has resulted in slightly smaller output feature maps that effectively lower the overall complexity of the model while preserving essential spatial-spectral cues.
3. **Spectral-Spatial Feature Extraction:** The model has been designed to capture both spectral and spatial attributes simultaneously, leveraging their relationship for improved classification accuracy. This dual approach has enhanced its ability to differentiate land cover types, even when spectral signatures alone have been insufficient.

This chapter is structured as follows: Section 3.2 reviews the state-of-the-art (SOTA) HSI classification architectures, highlighting methodologies and challenges. Next, Section 3.3 defines the problem statement, while Section 3.4 presents the proposed methodology, detailing its framework and innovations. Section 3.5 describes the experimental setup and analyzes the results through a comparative evaluation. Finally, Section 3.6 summarizes the findings and contributions.

3.2 Related Work

HSI classification is challenging due to its high dimensionality, which, despite providing rich spectral information, leads to noise amplification and the Hughes effect [128, 129]. Dimensionality reduction techniques such as PCA and compression-based algorithms have been explored [38, 39], but they often discard valuable information, highlighting the need for more advanced classification approaches.

Spectral Feature-Based Models: Early HSI classification methods used linear transformations to extract dominant spectral features [130–132]. However, they struggled with non-linear interactions caused by ground scattering effects [128] and often misclassified materials with similar spectral signatures [50]. Moreover, DL models like SAE and DBN improved spectral feature extraction [133, 134] but were computationally expensive due to FC layers. Additionally, converting HSI data into one-dimensional vectors led to spatial information loss, reducing classification accuracy.

Spatial Feature-Based Models: CNNs have been employed to capture spatial patterns in HSI data, utilizing sparsely connected layers and shared weights to enhance feature extraction while reducing computational complexity. Approaches like pixel-pair strategies [73], multiscale feature fusion networks [135], and hybrid architectures such as R-3D-CNN [136] have successfully incorporated spatial context for improved classification. However, 2D-CNNs, primarily focusing on spatial features, struggle to integrate spectral information effectively, especially in deeper networks [113, 137]. This limitation hampers their ability to distinguish spectrally similar classes, highlighting the need for models that better exploit spectral-spatial dependencies.

Integrated Spectral-Spatial Feature-Based Models: Recent approaches integrate both types of features to overcome the limitations of the models based only on spectral features or spatial features. Techniques such as 3D-CNNs [138], encoder-decoder architectures [139], and hybrid frameworks that combine 3D-CNN and 2D-CNN layers [110] enhance classification by preserving spatial structures while capturing spectral-spatial relationships. However, these models often require large training datasets and incur high computational costs. While methods like Local Binary Patterns (LBP) [96] attempt to address these issues, they struggle to balance efficiency and complexity. These challenges have driven the development of our MDCNN, which enhances spectral-spatial feature extraction while improving computational efficiency.

3.3 Problem Statement

Given an HSI data cube $\mathbf{I} \in \mathbb{R}^{[H \times W \times B]}$, where H and W represent the spatial dimensions respectively, and B denotes the number of spectral bands. Our aim is to extract fine-grained spectral features while preserving spatial structures efficiently. Many existing methods fail to adequately capture the intricate relationships between spectral features $\mathbf{I}_s \in \mathbb{R}^{[1 \times 1 \times B]}$ at each pixel and spatial features $\mathbf{I}_{sp} \in \mathbb{R}^{[H \times W \times 1]}$, which leads in loss of important contextual cues.

To address this, we first define the feature extraction functions $F_s : \mathbf{I} \rightarrow \mathbb{R}^{[H \times W \times \mathfrak{K}]}$ and $F_{sp} : \mathbf{I} \rightarrow \mathbb{R}^{[H \times W \times \mathfrak{K}]}$ represent the functions that map the $\mathbf{I}$ to extract spectral and spatial feature maps, respectively. Here, $\mathfrak{K}$ is the number of extracted features. Secondly, we focus on improving classification accuracy and combining $\mathbf{I}_s$ and $\mathbf{I}_{sp}$ effectively. Thus, we introduce an integration function as follows: $\mathbb{R}^{[H \times W \times \mathfrak{K}]} \times \mathbb{R}^{[H \times W \times \mathfrak{K}]} \rightarrow \mathbb{R}^{[H \times W \times \mathfrak{K}']}$. Our objective is to maximize the discriminative power of $\mathbf{I}$ while maintaining a balance between computational complexity and classification performance. This is formulated as:

$$\underset{F_s, F_{sp}, F_{\text{integrate}}}{\text{maximize}} \quad \hat{\mathcal{C}}(F_s(\mathbf{I}), F_{sp}(\mathbf{I}), F_{\text{integrate}}(F_s(\mathbf{I}), F_{sp}(\mathbf{I}))), \tag{3.1}$$

where $\hat{\mathcal{C}}$ denotes the classification accuracy, and the goal is to extract and integrate the spatial-spectral features in a manner that maximizes the classification performance $\hat{\mathcal{C}}$, ensuring both high accuracy and efficient computation.

3.4 Proposed Methodology

As per the problem statement, Figure 3.1 describes the workflow of the proposed MDCNN model which includes (a) Dimension reduction of HSI using PCA, (b) Binarization of the low-dimension HSI cube, (c) Mathematical morphological operations on the binarized data, (d) Concatenation of low-dimension HSI cube and morphological feature maps (Erosion, Closing, and Gradient), (e) Extraction of patches for the input of the CNNs, (f) Fine-grain correlated spectral-spatial features extraction using standard 3D convolution and Dilated-3D convolution, (g) Discriminative spatial feature extraction using Dilated-2D convolution, and (h) Classification of objects using softmax.

In MDCNN, feature selection uses PCA-based band reduction to remove spectral redundancy and morphological pre-processing to enhance spatial structures. The selected features are then processed with 3D dilated convolutions for joint spectral–spatial extraction and 2D dilated convolutions for spatial extraction, capturing local spectral–spatial features effectively. The methodology for the proposed MDCNN model can be

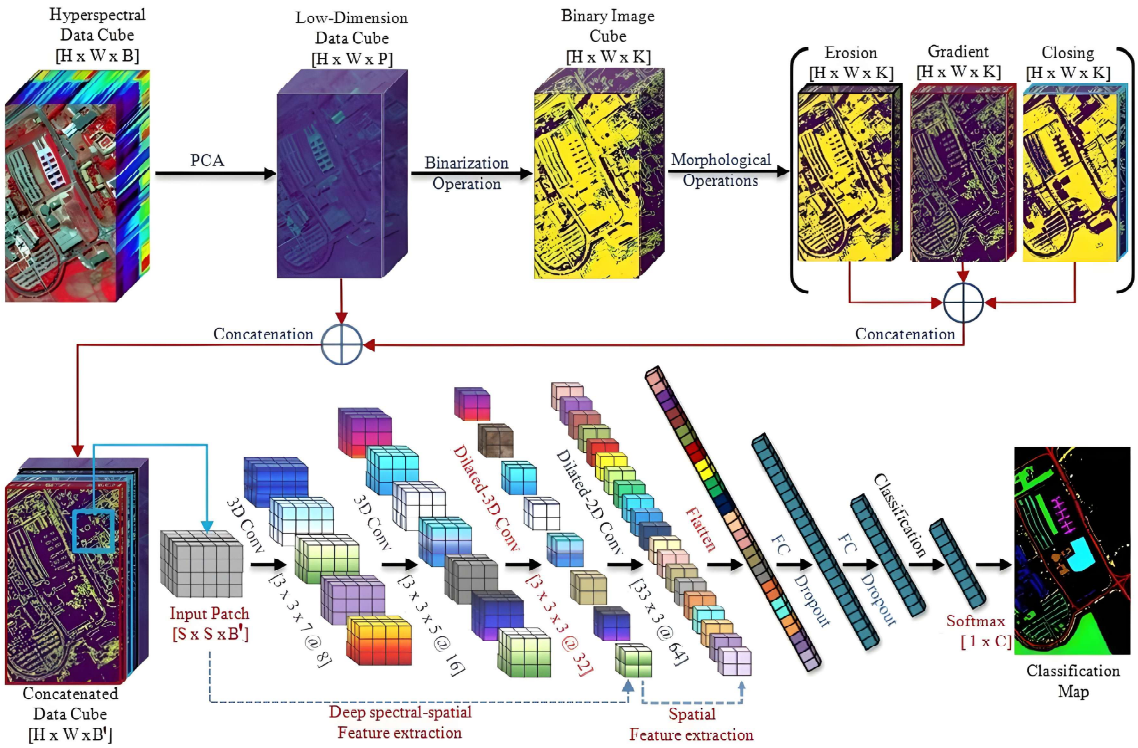


Figure 3.1: MDCNN Framework for HSI Classification includes PCA for Dimensionality Reduction, Morphological Operations (Erosion, Dilation, Gradient), and Feature Extraction using 3D and 2D Dilated CNNs with Softmax Classification.

summarized in the following steps:

1. **HSI Representation:** Represent HSI as a 3D hypercube $\mathbf{I} \in \mathbb{R}^{[H \times W \times B]}$, where H, W, and B denote height, width, and number of spectral bands.
2. **Dimensionality Reduction:** Apply PCA to reduce spectral redundancy while preserving maximum variance. The reduced cube is represented as $\mathbf{I_p} \in \mathbb{R}^{[H \times W \times P]}$.
3. **Spectral Band Selection and Binarization:** Select the first K spectral bands and perform binarization to produce $\mathbf{I_B} \in \mathbb{R}^{[H \times W \times K]}$.
4. **Morphological Feature Extraction:** Apply three parallel morphological operations on $\mathbf{I_B}$: (a) erosion, (b) dilation, and (c) gradient. Each operation outputs $[H \times W \times K]$.
5. **Composite Feature Cube Construction:** Concatenate the PCA-reduced cube $\mathbf{I_p}$ with morphological outputs to form the composite cube $\mathbf{I_c} \in \mathbb{R}^{[H \times W \times B']}$, where $B' = (P + 3K)$.
6. **Patch Extraction:** Extract spatial–spectral patches $\mathbf{I_{Patch}} \in \mathbb{R}^{[S \times S \times B']}$ from the composite cube.

7. **Deep Feature Learning via MDCNN:** Feed patches into MDCNN: (a) 3D standard and dilated convolutions learn fine-grained spectral–spatial correlations; (b) 2D dilated convolutions extract discriminative spatial features.
8. **Classification:** Apply a softmax layer for the final classification of objects/pixels.

3.4.1 HSI Binarization Module

The binarization method [140] converts image pixel values to a binary scale of 0 or 1, simplifying data representation and processing. Equation 3.2 outlines the binarization pipeline, which includes normalization, threshold computation, and binary image generation. In HSI, it segments spectral cues to enhance feature extraction and reduce computational complexity [141]. The complete binarization process is expressed as follows:

$$\Psi_{i,j} = \frac{255 * (\mathbf{I}_{pk_{i,j}} - min(\mathbf{I}_{pk}))}{\mathbf{I}_{pk_{i,j}}},\ \Theta = \frac{\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\Psi_{i,j}}{H \times W \times K},\ \mathbf{I}_{\mathbf{B_{i,j}}i,j} = \begin{cases} \mathbf{1}, & \textbf{if } \Psi_{i,j} \geq \Theta \\ \mathbf{0}, & \textbf{if } \Psi_{i,j} < \Theta \end{cases} \tag{3.2}$$

HSI $\mathbf{I} \in \mathbb{R}^{[H \times W \times B]}$ undergoes PCA-based spectral reduction, yielding $\mathbf{I_P} \in \mathbb{R}^{[H \times W \times P]}$. The top K principal components form $\mathbf{I_{pk}}$, capturing spectral information while reducing redundancy. Next, $\mathbf{I_{pk}}$ is normalized to $\Psi_{i,j}$, rescaling pixel intensities to $[0, 255]$. The global threshold Θ, computed as the mean intensity of Ψ, defines the binarization boundary. Pixels in Ψ meeting or exceeding Θ are set to 1, and others to 0, producing the binary image $\mathbf{I_B} \in \mathbb{R}^{[H \times W \times K]}$. The process enhances region segmentation and spatial feature extraction while preserving spectral characteristics.

3.4.2 Mathematical Morphology Module

HSIs possess high spatial resolution and well-defined object boundaries with minimal mixed pixels, making them highly suitable for extracting spatially discriminative features in land cover analysis. Morphological analysis, introduced by Serra et al. [142], provides a structural and boundary analysis framework using Structuring Elements (SEs). The proposed mathematical morphology module enhances spatial features through three key operations: erosion, closing, and gradient. These operations helps in improving classification accuracy. These operations are mathematically defined as follows:

$$\textbf{Erosion:}\ \ (\mathbf{I_B} \ominus \mathbf{g})(i,j) = \min[\mathbf{I_B}(m+i, n+j) - \mathbf{g}(m,n)] \tag{3.3}$$

$$\textbf{Dilation:}\ \ (\mathbf{I_B} \oplus \mathbf{g})(i,j) = \max[\mathbf{I_B}(i-m, j-n) + \mathbf{g}(m,n)] \tag{3.4}$$

$$\textbf{Closing:} \quad (I_B \cdot g)(i,j) = (\mathbf{I_B} \oplus \mathbf{g}) \ominus g \tag{3.5}$$

$$\textbf{Gradient:} \quad \mathbf{G}(i,j) = \big((\mathbf{I_B} \oplus \mathbf{g}) - (\mathbf{I_B} \ominus \mathbf{g})\big)(i,j) \tag{3.6}$$

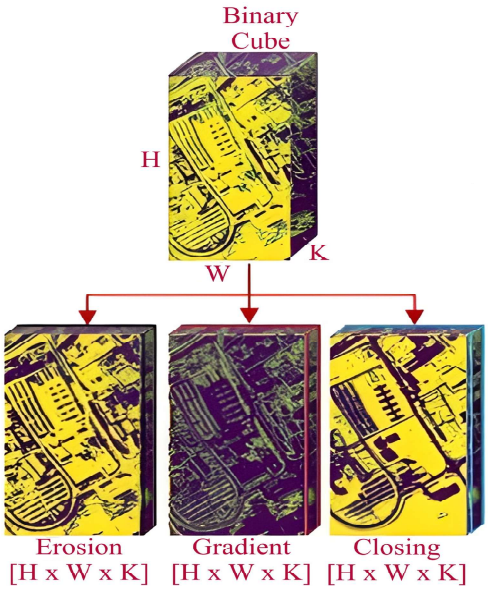


Figure 3.2: Morphological Feature Maps erosion, gradient and closing.

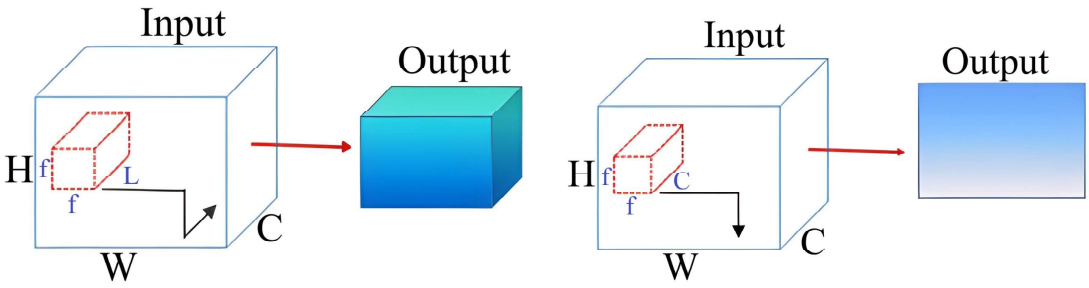
where, $\mathbf{I_B}(i,j)$ represents the binary image, $\mathbf{g}(m,n)$ is the structuring element, and $\ominus$ and $\oplus$ denote erosion and dilation, respectively. Erosion, as expressed in Equation 3.3, contracts objects by removing boundary pixels, effectively isolating structures and suppressing noise. Dilation, as expressed in Equation 3.4, expands object boundaries by adding pixels, bridging gaps, and enhancing connectivity. The gradient operation, expressed in Equation 3.5, is computed as the difference between dilation and erosion, capturing boundary details and aiding in precise object contour detection. The proposed module uses a 3×3 cross-shaped SE defined as:

$$\mathbf{g(i,j)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.7}$$

This SE ensures computational efficiency while capturing key spatial characteristics of $\mathbf{I_B}$. Figure 3.2 depicts morphological feature maps from erosion, dilation, and gradient operations, highlighting extracted spatial information.

3.4.3 Fine-Grained Spectral-Spatial Feature Extraction Module

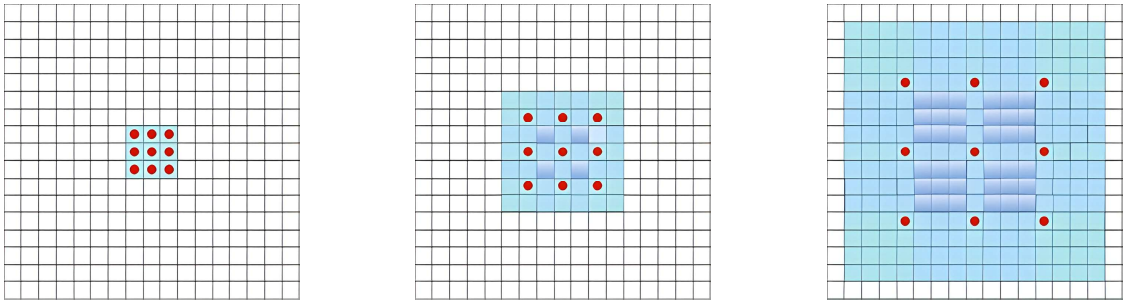
Nebauer et al. [143] pioneered the application of CNNs in visual recognition. In a conventional 3D convolution process a 3D input of size $[\hat{H} \times \hat{W} \times \hat{B}]$ convolve with a



(a) (b)

Figure 3.3: Illustration of 3D and 2D convolutions: (a) 3D Convolution: A 3D input convolved with a 3D kernel having fewer channels than the HSI produces a 3D feature map ($L < C$). (b) 2D Convolution: A 3D input convolved with a 3D kernel having the same channels as the HSI generates a 2D feature map.

kernel of size $[f_1 \times f_1 \times \dot{B}]$, where $\dot{B} < \hat{B}$, to capture hierarchical fine grain spectral-spatial features. The resulting 3D feature map retains spectral dependencies across channels, as illustrated in Figure 3.3 (a). Conversely, a conventional 2D convolution applies a kernel $[f_2 \times f_2 \times \hat{B}]$ on the 3D input of size $[\hat{H} \times \hat{W} \times \hat{B}]$, reducing dimensionality and extracting dominant spatial features, as shown in Figure 3.3 (b). While effective for local feature extraction, both methods have limited receptive fields, restricting their ability to model long-range dependencies. Dilated convolution addresses this limitation by introducing a fixed gap between kernel elements, determined by the dilation rate, enabling the capture of a broader spatial context. This approach eliminates the need for pooling, enhances feature learning, and reduces overfitting. In contrast, standard convolution, with a dilation rate of $r = 1$, lacks this flexibility and captures only local features. Figure 3.4 illustrates the effect of different dilation rates—(a) one, (b) two, and (c) three—highlighting its role in extracting fine-grained spectral-spatial features.



(a) (b) (c)

Figure 3.4: Dilated convolution with dilation rates: (a) one, (b) two and (c) three.

3.4.3.1 Conventional 3D Convolution

3D convolution operates on volumetric data, making it well-suited for HSI analysis. As we know that in the HSI data cube, spectral and spatial information is inherently interconnected. Given a 3D input patch $\mathbf{I}_{\text{patch}} \in \mathbb{R}^{[S\times S\times B'\times 1]}$, where $S\times S$ represents the spatial dimensions and B' denotes the spectral bands, the mathematical formulation of 3D convolution operation is defined as follows:

$$\mathbf{I}_{3D1}(i,j,k) = \sum_m \sum_n \sum_p \mathbf{I}_{\text{patch}}(i-m, j-n, k-p) \cdot \mathbf{K_a}(m,n,p), \tag{3.8}$$

where $\mathbf{K_a}$ is the 3D kernel of size $[K_{a_1}, K_{a_2}, K_{a_3}]$, and the number of filters is f_a. The output feature map $\mathbf{I}_{\text{3D1}}$ has dimensions:$[(S+1-K_{a_1})\times(S+1-K_{a_2})\times(B'+1-K_{a_3})\times f_a]$. A second 3D convolution further refines spectral-spatial features:

$$\mathbf{I}_{3D2}(i,j,k) = \sum_m \sum_n \sum_p \mathbf{I}_{3D1}(i-m, j-n, k-p) \cdot \mathbf{K_b}(m,n,p), \tag{3.9}$$

where $\mathbf{K_b}$ has kernel size $[K_{b_1}, K_{b_2}, K_{b_3}]$ with f_b filters. The resulting feature map $\mathbf{I}_{\text{3D2}}$ has dimensions: $[(S+4-K_{a_1}-K_{b_1}+2-r_c(K_{c_1}-1))\times(S+4-K_{a_2}-K_{b_2}+2-r_c(K_{c_2}-1))\times(B'+4-K_{a_3}-K_{b_3}+2-r_c(K_{c_3}-1))\times f_c]$. Although conventional 3D convolution effectively captures spectral-spatial features, it is computationally expensive and limited by a localized receptive field. Dilated 3D convolution overcomes this limitation by expanding the receptive field without increasing the number of parameters. It also enables better modeling of long-range spectral-spatial dependencies.

3.4.3.2 Dilated 3D Convolution

Dilated 3D convolution extends conventional 3D convolution by introducing a dilation factor r_c, which expands the receptive field while maintaining the same number of parameters. This facilitates improved spectral-spatial feature extraction by capturing long-range dependencies. Figure 3.4 illustrates how dilation enlarges the receptive field, allowing better aggregation of contextual information. Given a kernel $\mathbf{K_c}$ of size $[K_{c_1} \times K_{c_2} \times K_{c_3}]$ with f_c filters and a dilation factor $r_c = 2$, the dilated 3D convolution is defined as:

$$\mathbf{I}_{R3D3}(i,j,k) = \sum_m \sum_n \sum_p \mathbf{I}_{3D2}(i-r_cm, j-r_cn, k-r_cp) \cdot \mathbf{K_c}(m,n,p). \tag{3.10}$$

The output feature map $\mathbf{I}_{R3D3}$ has dimensions: $[(S+4-K_{a_1}-K_{b_1}+2-r_c(K_{c_1}-1)+2-r_d(K_{d_1}-1))\times(S+4-K_{a_2}-K_{b_2}+2-r_c(K_{c_2}-1)+2-r_d(K_{d_2}-1))\times f_d]$.

3.4.3.3 Dilated 2D Convolution

While dilated 3D convolution enhances multi-scale spectral-spatial feature extraction, it does not fully leverage fine-grained spatial structures. To address this, dilated 2D convolution is applied, refining spatial representation while maintaining computational efficiency. First, the feature map $\mathbf{I}_{\mathrm{R3D3}}$ is reshaped by merging its last two dimensions into a single spatial depth. Then, a dilated 2D convolution is performed using a kernel $\mathbf{K}_d$ of size $[K_{d_1}\times K_{d_2}]$ with f_d filters and a dilation factor r_d, formulated as:

$$\mathbf{I}_{R2D4}(i,j)=\sum_m\sum_n\sigma(\mathbf{I}_{R3D3})(i-r_dm,j-r_dn)\cdot\mathbf{K}_d(m,n), \tag{3.11}$$

where r_d controls the spacing between kernel elements, and σ denotes the reshaping operation. The output feature map $\mathbf{I}_{R2D4}$ has the dimensions:$[(S+4-K_{a_1}-K_{b_1}+2-r_c(K_{c_1}-1)+2-r_d(K_{d_1}-1))\times(S+4-K_{a_2}-K_{b_2}+2-r_c(K_{c_2}-1)+2-r_d(K_{d_2}-1))\times f_d.]$ By incorporating dilated 2D convolution, the model enhances spatial feature extraction, leading to a more refined representation and improved classification accuracy.

3.4.4 MDCNN Framework

The proposed methodology processes HSI data using PCA for dimensionality reduction while preserving key spectral details. The PCA-reduced data undergoes binarization, followed by three morphological operations—erosion, dilation, and gradient—before being concatenated to form a composite feature representation. Input patches from this composite data are then fed into the CNN module for feature extraction. The MDCNN framework extracts spectral-spatial features using two conventional 3D convolution layers: a dilated 3D convolution layer and a final dilated 2D convolution layer. This design balances local feature extraction with long-range dependencies while optimizing computational efficiency. Table [3.1] outlines the MDCNN framework's layer specifications, output dimensions, and parameter counts for the IP, PU, and SA datasets.

The process begins with an input layer of size $[21\times21\times B'\times1]$, where B' represents the spectral bands. The first convolutional layer, Conv3DA, applies a conventional 3D convolution with a kernel size of $K_a=[3\times3\times7]$ and $f_a=8$ filters, capturing fundamental spectral-spatial features. The second layer, Conv3DB, also employs a conventional 3D convolution with $K_b=[3\times3\times5]$ and $f_b=16$ filters, refining the

Table 3.1: Structure of MDCNN Framework for (a) IP (b), PU, and (c) SA datasets.

Datasets		IP		PU		SA	
Layer	Filter Size	Output Shape	PramC	Output Shape	PramC	Output Shape	PramC
Input_Layer	–	[21 × 21 × 30 × 1]	0	[21 × 21 × 15 × 1]	0	[21 × 21 × 15 × 1]	0
Conv3DA	[3 × 3 × 7@8]	[19 × 19 × 24 × 8]	512	[19 × 19 × 9 × 8]	512	[19 × 19 × 9 × 8]	512
Conv3DB	[3 × 3 × 5@16]	[17 × 17 × 20 × 16]	5776	[17 × 17 × 5 × 16]	5776	[17 × 17 × 5 × 16]	5776
Conv3DDilated	[3 × 3 × 3@32]	[13 × 13 × 18 × 32]	13856	[13 × 13 × 3 × 32]	13856	[13 × 13 × 3 × 32]	13856
Reshape	–	[13 × 13 × 576]	0	[13 × 13 × 96]	0	[13 × 13 × 96]	0
Conv2DDilate	[3 × 3@64]	[9 × 9 × 64]	331840	[9 × 9 × 64]	55360	[9 × 9 × 64]	55360
FlattenA	–	5184	0	5184	0	5184	0
DenseA	–	256	1327360	256	1327360	256	1327360
DropoutA	0.4	256	0	256	0	256	0
DenseB	–	128	32896	128	32896	128	32896
DropoutB	0.4	128	0	128	0	128	0
DenseC	–	16	2064	9	1152	16	2064
Total Parameters		1714304		1436912		1437824	

extracted features. To expand the receptive field and capture long-range dependencies, the third convolutional layer, Conv3DDilated, utilizes a dilated 3D convolution with a kernel size of $K_c = [3 \times 3 \times 3]$, $f_c = 32$ filters, and a dilation rate of $r_c = 2$, which enables efficient feature extraction without a proportional increase in parameters. The extracted feature maps are then reshaped to $[13 \times 13 \times C]$ and processed by Conv2DDilate, a dilated 2D convolutional layer with a kernel size of $[3 \times 3]$, dilation rate $r_d = 2$ and $f_d = 64$ filters. This layer enhances spatial representations and improves class separability. Finally, the feature maps are flattened and passed through FC layers (DenseA, DenseB, and DenseC), incorporating dropout layers to prevent overfitting. Softmax classification is applied at the final layer for HSI classification.

3.5 Experiments: Implementations and Results

The section presents a comprehensive evaluation of the proposed MDCNN model. It covers datasets, training configurations, and experimental settings. Key hyperparameters, including patch size (PS), training sample effectiveness, and dilation factor influence, are analyzed. Finally, the performance of our proposed methodology is compared with SOTA methods to demonstrate its effectiveness in HSI classification.

3.5.1 Dataset Description

The subsection describes the implementation of the MDCNN classification system. The model is developed using Python and the Keras deep learning library. Experiments are conducted on Google Colab with an Intel Xeon CPU @ 2.20GHz, an NVIDIA Tesla T4 GPU, and 25 GB of RAM. Model performance is evaluated on three publicly

Table 3.2: The distribution of color codes, name of the classes, training, and testing sample distributions are outlined for the IP, PU, and SA datasets.

Dataset	IP				PU				SA			
Class	Color	Name	Train	Test	Color	Name	Train	Test	Color	Name	Train	Test
C01		Alfalfa	14	32		Asphalt	1989	4642		Brocoli-gw1	603	1406
C02		Corn-NT	428	1000		Meadows	5594	13055		Brocoli-gw2	1118	2608
C03		Corn-MT	249	581		Gravel	630	1469		Fallow	593	1383
C04		Corn	71	166		Trees	919	2145		Fallow-RP	418	976
C05		Grass-P	145	338		Painted-MS	403	942		Fallow-S	803	1875
C06		Grass-T	219	511		Bare-S	1509	3520		Stubble	1188	2771
C07		Grass-PM	8	20		Bitumen	399	931		Celery	1074	2505
C08		Hay-W	143	335		SB-Bricks	1105	2577		Grapes-U	3381	4890
C09		Oats	6	14		Shadows	284	663		Soil-VD	1861	4342
C10		Soybean-NT	292	680						Corn-SGW	983	2295
C11		Soybean-MT	736	1719						Lettuce-4wk	320	748
C12		Soybean-C	178	415						Lettuce-5wk	578	1349
C13		Wheat	62	143						Lettuce-6wk	275	641
C14		Woods	379	886						Lettuce-7wk	321	749
C15		Buildings-GTD	116	270						Vinyard-U	2180	5088
C16		SS-Towers	28	65						Vinyard-VT	524	1265

available HSI datasets: IP, PU, and SA. These datasets differ in spatial dimensions, spectral ranges, and land cover classes. Section 1.3 and Table 1.1 summarize their sensor specifications, wavelengths, spatial sizes, spectral bands, and classes. Table 3.2 presents the training and testing sample distribution, class names, and color codes.

3.5.2 Training Details and Experimental Settings

We have evaluated the MDCNN model by allocating 30% of the input samples for training and the rest for testing. Experiments on IP, PU, and SA datasets involved training for 100 epochs using the Adam optimizer with a mini-batch size of 256 for efficiency. The hyper-parameters are fine-tuned for accuracy, setting learning rate $lr = 0.001$ with exponential decay, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-6$ for stability. Xavier initialization ensured well-scaled kernel weights. Spatial patches of size $[21 \times 21 \times 15]$ are extracted for IP, while PU and SA used $[21 \times 21 \times 6]$ to maintain consistent pre-processing. The proposed MDCNN model is benchmarked against SOTA models, including SVM [144], 2D-CNN [145], 3D-CNN [146], R-3D-CNN [136], SSRN [113], LBP-DC-CNN [96], and HybridSN [110]. Classification effectiveness has been measured through OA, AA, and κ scores, complemented by qualitative insights from classification map visualizations. Computational complexity has been analyzed in terms of P_M. At the same time, efficiency has been assessed by recording Tr and Te. All experiments are repeated ten times to ensure the robustness of the proposed model. It reports the mean and standard deviation of classification results and further validates the reliability and consistency of the MDCNN model.

3.5.3 Hyperparameter Sensitivity Analysis

The MDCNN framework has been evaluated to optimize spectral-spatial feature extraction and computational efficiency. Key Hyperparameters such as spatial window size, training data ratio, and dilated convolutions were analyzed for their impact on classification accuracy and processing time. This analysis guided model refinement to achieve a balance between performance and efficiency on HSI datasets.

Effectiveness of Patch Size (PS): PS plays a crucial role in the classification performance of the MDCNN framework. Smaller patches provide limited spatial context, leading to lower accuracy. In contrast, excessively large patches introduce redundancy, increasing computational cost and the risk of overfitting. To determine the optimal configuration, six spatial window sizes—[15 × 15], [17 × 17], [19 × 19], [21 × 21], [23 × 23], and [25 × 25]—were evaluated using 30% of the samples for training. As shown in Table 3.3, the SOTA [21 × 21] consistently achieves the highest classification performance, with OA values of 99.80% for IP, 99.99% for PU, and 100% for the SA dataset. Larger SOTAs, such as [23 × 23] and [25 × 25], show a slight decline in accuracy due to increased redundancy. In contrast, smaller patches, like [15 × 15] and [17 × 17], fail to capture sufficient spatial information, resulting in lower OA scores. Ultimately, the [21 × 21] configuration effectively balances spatial context and computational efficiency, ensuring optimal performance for HSI classification.

Table 3.3: Hyperparameter sensitivity analysis of MDCNN: Impact of patch size and training sample size on classification performance (OA in %, AA in %, $\kappa \times 100$) for IP, PU, and SA datasets. Best results for each dataset are highlighted.

Dataset & Metrics		Patch Size						Training Amount				
		15x15	17x17	19x19	21x21	23x23	25x25	10%	20%	30%	40%	50%
IP	OA	99.45	99.57	99.55	**99.80**	99.75	99.70	97.97	99.39	**99.80**	99.78	99.72
	AA	99.52	98.82	99.67	**99.87**	99.77	99.21	95.17	99.39	**99.87**	99.86	99.84
	κ	99.38	99.51	99.49	**99.77**	99.71	99.67	97.67	99.30	99.77	**99.81**	99.76
PU	OA	99.95	99.96	99.97	**99.99**	99.96	99.98	99.70	99.96	**99.99**	**99.99**	**99.99**
	AA	99.88	99.90	99.93	**99.99**	99.91	99.95	99.17	99.91	**99.99**	**99.99**	99.95
	κ	99.94	99.95	99.96	99.97	99.96	**99.98**	99.60	99.95	99.97	99.95	**99.98**
SA	OA	99.98	99.98	**100**	**100**	**100**	**100**	99.99	99.99	**100**	**100**	99.99
	AA	99.98	99.98	**100**	**100**	**100**	**100**	99.97	99.97	**100**	**100**	99.99
	κ	99.98	99.98	**100**	**100**	**100**	**100**	99.98	99.99	**100**	**100**	99.99

Effectiveness of Training Sample Size The training sample size is critical for accurate and generalized MDCNN classification. A small dataset limits feature learning, reducing OA. Conversely, a large dataset increases computational costs and the risk of overfitting. To analyze this impact, model performance is evaluated using training proportions of 10%, 20%, 30%, 40%, and 50% of the total samples. As shown in Ta-

ble 3.3, the MDCNN framework achieves peak performance with 30% training data, yielding the highest OA, AA, and κ across most datasets. While 40% and 50% training sizes offer slight improvements, particularly for UP and SA, the gains come with increased computational overhead. In contrast, a 10% training sample significantly drops accuracy due to insufficient class representation. The findings confirm that 30% is an optimal balance between classification performance and computational efficiency.

Effectiveness of Dilated Convolution The performance of MDCNN depends on the placement and number of convolution layers configured as dilated convolution layers. Four different configurations are applied: (a) all convolution layers use dilation, (b) only the 3D convolution layers use dilation, (c) all layers except the first 3D convolution layer use dilation, and (d) only the last 3D convolution layer and the 2D convolution layer use dilation. The experiments utilize a $[21 \times 21]$ SOTA with 30% of the data for training. OA, AA, κ, and the number of trainable parameters are analyzed to assess the trade-off between model complexity and OA. Table 3.4 shows that the fully dilated configuration performs the worst due to overfitting. In contrast, a balanced mix improves accuracy. The best results are obtained when only the last 3D and 2D layers use dilated convolutions, optimizing the receptive field without excessive parameters. This configuration achieves 99.80% OA in IP, with similar trends in PU and SA, highlighting the importance of balancing dilation and complexity.

Table 3.4: Evaluation of MDCNN configurations with varying numbers of dilated convolution layers in terms of OA, AA, κ, and the number of trainable parameters for the IP, UP, and SA datasets. The best results for each dataset are highlighted.

Datasets	Evaluation Metrics	All Layers	All 3D Layers	Last 3 Layers	Last 2 Layers
IP	OA(%)	99.49	99.60	99.56	**99.80**
	AA(%)	97.50	99.49	98.41	**99.87**
	$\kappa \times 100$	99.42	99.55	99.50	**99.77**
	P_M	**796800**	1190016	1190016	1714304
PU	OA(%)	99.90	99.95	99.97	**99.99**
	AA(%)	99.94	99.89	99.91	**99.99**
	$\kappa \times 100$	99.88	99.94	**99.97**	**99.97**
	P_M	**519417**	912633	912633	1436921
SA	OA(%)	99.98	99.99	99.99	**100**
	AA(%)	99.95	99.99	99.99	**100**
	$\kappa \times 100$	99.98	99.98	99.99	**100**
	P_M	**520320**	913536	913536	1437824

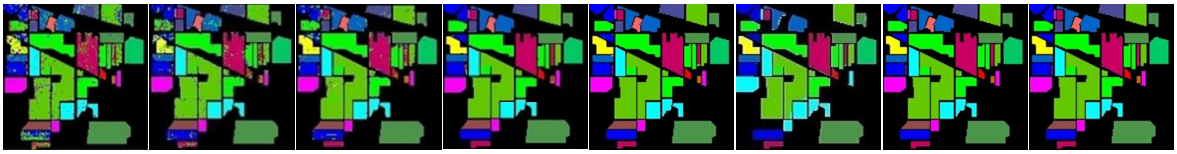
3.5.4 Comparison with State-of-the-Art (SOTA) Methods

We have demonstrated the efficacy of the proposed MDCNN model by benchmarking it against SOTA techniques. Experiments use three publicly available HSI datasets: IP,

PU, and SA. The model is trained on 30% of the input samples, while the remaining 70% is used for testing. We have also evaluated performance using three key metrics: OA, AA, and κ for a comprehensive assessment. Moreover, we have also analyzed the impact of model complexity by comparing Tr and Te. This highlights the balance between OA and computational efficiency.

Table 3.5: Classification Accuracy Comparison on IP Dataset with Best Results Highlighted in Terms of OA(%), AA (%) and Kappa Coefficient.

Class	SVM	2D-CNN	3D-CNN	SSRN	R3D-CNN	LBP-CNN	HybridSN	MD-CNN
C01	82.20	75.00	79.23	97.82	**100**	97.45	99.38	**100.00**
C02	73.82	81.40	88.60	99.17	**100.00**	98.58	99.58	99.62
C03	82.15	87.60	85.81	99.53	**100.00**	98.39	99.66	99.90
C04	77.12	62.04	90.53	97.79	**100.00**	97.92	99.88	**100.00**
C05	73.66	92.3	96.11	99.24	**100.00**	**100.00**	99.53	99.82
C06	93.40	99.21	98.43	99.51	**100.00**	97.99	99.96	**100.00**
C07	96.21	75.00	92.36	98.70	**100.00**	99.21	99.00	**100.00**
C08	85.72	100.00	98.51	99.85	**100.00**	**100.00**	**100.00**	**100.00**
C09	97.38	64.28	88.90	98.50	**100.00**	98.69	**100.00**	**100.00**
C10	71.01	82.79	87.72	98.74	**99.65**	99.21	99.56	99.57
C11	76.50	91.27	91.42	99.30	99.31	97.87	**99.84**	99.79
C12	83.91	82.89	90.04	98.43	**98.85**	96.63	99.52	99.42
C13	83.56	99.32	99.02	**100.00**	**100.00**	99.12	99.86	**100.00**
C14	98.63	98.87	97.95	99.31	99.73	99.37	**100.00**	99.98
C15	94.21	86.29	82.57	99.2	96.46	98.66	99.85	**100.00**
C16	69.93	**100.00**	98.51	97.82	96.42	97.78	98.46	**100.00**
OA	85.30±2.81	89.48±0.15	91.10±0.42	99.19±0.26	99.50±0.3	98.52±0.05	99.75±0.11	**99.80±0.03**
AA	79.03±2.65	86.14±0.82	91.58±0.15	98.93±0.59	99.42±0.03	98.68±0.06	99.63±0.15	**99.87±0.03**
$\kappa \times$ **100**	83.10±3.15	87.96±0.51	89.98±0.50	99.07±0.30	99.48±0.2	98.29±0.07	99.71±0.13	**99.77±0.04**



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

Figure 3.5: The classification outcome Map for IP data: (a) SVM (b) 2D-CNN (c) 3D-CNN (d) SSRN (e) R-3D-CNN (f) LBP-DC-CNN (g) HybridSN and (h) MDCNN.

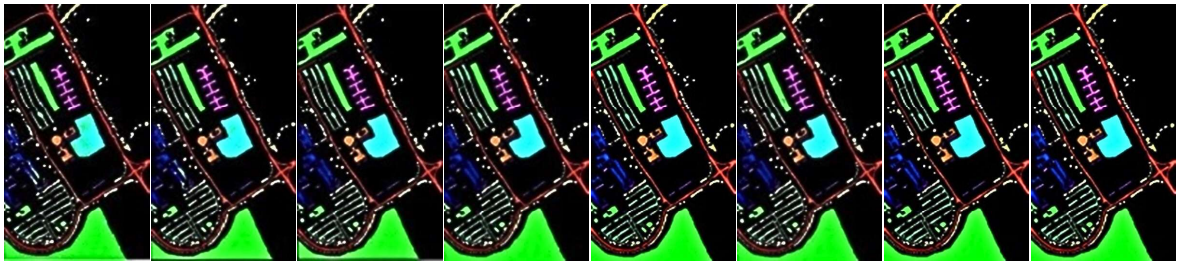
IP Dataset: The IP dataset results in Table 3.5 show that hybrid DL models outperform SVM. MDCNN achieves the highest OA of 99.80%. HybridSN follows at 99.75%. R-3D-CNN reaches 99.50%, while SSRN performs well at 99.19%. Traditional CNNs, such as 2D-CNN and 3D-CNN, show lower accuracy. This highlights the importance of spectral-spatial feature extraction in HSI classification. The classification maps in Figure 3.5 support these findings. Hybrid DL models, such as HybridSN and MDCNN, show better spatial consistency. SVM and 2D-CNN struggle with misclassifications and

noise. SSRN and R-3D-CNN preserve fine spatial details. In general, DL models using spectral-spatial correlations improve accuracy.

As shown in Fig. 3.5, Corn-NT (, C02) and Corn-MT (, C03) show heavy overlap in SVM, 2D-CNN, and 3D-CNN due to limited spectral-spatial learning. SSRN and R3D-CNN reduce this but still leave boundary noise. HybridSN and MD-CNN achieves cleaner separation by capturing both local and global spatial-spectral features, preserving even small classes like Oats (, C09).

Table 3.6: Classification Accuracy Comparison on PU Dataset with Best Results Highlighted in Terms of OA(%), AA (%) and Kappa Coefficient.

Class	SVM	2D-CNN	3D-CNN	SSRN	R3D-CNN	LBP-CNN	HybridSN	MD-CNN
C01	94.72	98.51	98.40	**100.00**	99.16	**100.00**	**100.00**	**100.00**
C02	97.15	99.54	96.91	99.87	99.48	**100.00**	**100.00**	**100.00**
C03	82.73	84.62	97.05	**100.00**	99.98	**100.00**	**100.00**	**100.00**
C04	96.82	98.04	98.84	**100.00**	98.86	99.89	99.84	**100.00**
C05	99.71	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
C06	90.48	97.12	99.32	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
C07	87.73	95.05	98.92	**100.00**	99.91	**100.00**	**100.00**	99.92
C08	88.29	96.39	98.33	99.34	99.76	**100.00**	99.98	**100.00**
C09	99.88	99.69	99.92	**100.00**	**100.00**	98.94	99.88	99.82
OA	94.34 ± 0.18	97.86 ± 0.20	96.53 ± 0.08	99.90 ± 0.0	99.54 ± 0.14	99.87 ± 0.3	99.98 ± 0.01	**99.99 ± 0.01**
AA	92.98 ± 0.41	96.55 ± 0.03	97.57 ± 1.31	98.91 ± 0.0	99.36 ± 0.15	99.97 ± 02	**99.97 ± 0.01**	**99.97 ± 0.03**
$\kappa \times$ **100**	92.50 ± 0.70	97.16 ± 0.51	95.51 ± 0.21	99.87 ± 0.0	98.68 ± 0.09	99.84 ± 0.2	99.98 ± 0.01	**99.99 ± 0.01**



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

Figure 3.6: The classification outcome Map for PU data : (a) SVM (b) 2D-CNN (c) 3D-CNN (d) SSRN (e) R-3D-CNN (f) LBP-DC-CNN (g) HybridSN and (h) MDCNN.

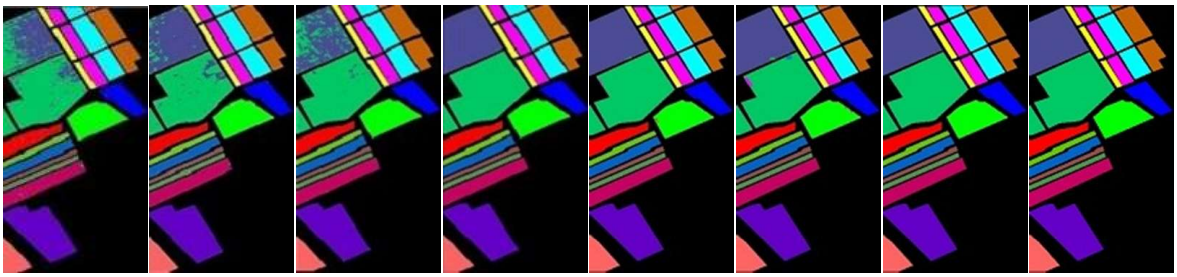
PU Dataset: The PU dataset results in Table 3.6 show the superiority of MDCNN over the ML and DL models. SVM records the lowest performance. 2D-CNN, 3D-CNN, and SSRN show improvements as compare to SVM. SSRN and R3D-CNN benefit from spectral-spatial feature extraction. MDCNN achieves the highest OA of 99.99%, AA of 99.97%, and κ of 99.99. It also attains 100% accuracy in most PU land cover classes, surpassing HybridSN, which lacks morphological feature descriptors. Figure 3.5 shows the classification maps. MDCNN exhibits superior spatial coherence and

reduced misclassification. These results confirm that morphological and multi-scale convolutional features enhance HSI classification accuracy.

As shown in Fig. 3.6, Meadows (, C02) and Gravel (, C03) exhibit significant noise in SVM and 2D-CNN due to weak spectral–spatial learning. SSRN and R3D-CNN reduce these distortions but still leave boundary inconsistencies. Painted-metal sheets (, C05) and Bitumen (, C07) also suffer misclassification caused by spectral similarity. Our MDCNN overcomes these limitations by leveraging multi-dimensional features, producing sharper boundaries and the most reliable classification map.

Table 3.7: Classification Accuracy Comparison on SA Dataset with Best Results Highlighted in Terms of OA(%), AA (%) and Kappa Coefficient.

Class	SVM	2D-CNN	3D-CNN	SSRN	R3D-CNN	LBP-CNN	HybridSN	MD-CNN
C01	99.6	**100**	98.41	**100**	99.83	99.75	**100**	**100**
C02	99.82	99.96	**100**	**100**	99.91	99.35	**100**	**100**
C03	99.26	99.63	99.23	**100**	99.66	**100**	**100**	**100**
C04	99.4	99.28	99.9	99.89	97.37	99.88	**100**	**100**
C05	99.42	99.2	99.43	**100**	99.5	99.76	**100**	**100**
C06	**100**	**100**	99.55	**100**	100	99.65	**100**	**100**
C07	99.83	**100**	99.72	**100**	99.53	99.4	**100**	**100**
C08	85.25	93.62	89.75	**100**	99.97	98.85	**100**	**100**
C09	99.71	100	99.81	**100**	**100**	99.98	**100**	**100**
C10	97.03	98.82	98.36	99.91	99.9	99.7	**100**	**100**
C11	98.24	99.73	98.12	**100**	**100**	**100**	**100**	**100**
C12	99.46	**100**	98.96	**100**	99.65	99.96	**100**	**100**
C13	98.77	**100**	98.93	**100**	**100**	99.81	**100**	**100**
C14	97.03	99.86	98.6	**100**	98.44	99.69	**100**	**100**
C15	72.71	91.52	79.31	99.96	**100**	99.61	**100**	**100**
C16	99.41	99.92	94.51	**100**	**100**	**100**	**100**	**100**
OA	92.95 ± 0.34	97.38 ± 0.02	93.96 ± 0.15	99.98 ± 0.10	99.80 ± 0.2	99.54 ± 0.05	**100 ± 0.00**	**100 ± 0.00**
AA	94.60 ± 2.28	98.84 ± 0.06	97.01 ± 0.63	99.97 ± 0.00	99.61 ± 0.2	99.50 ± 0.05	**100 ± 0.00**	**100 ± 0.00**
$\kappa \times$ **100**	92.11 ± 0.18	97.08 ± 0.10	93.32 ± 0.50	99.97 ± 0.10	99.72 ± 0.2	99.71 ± 0.05	**100 ± 0.00**	**100 ± 0.00**

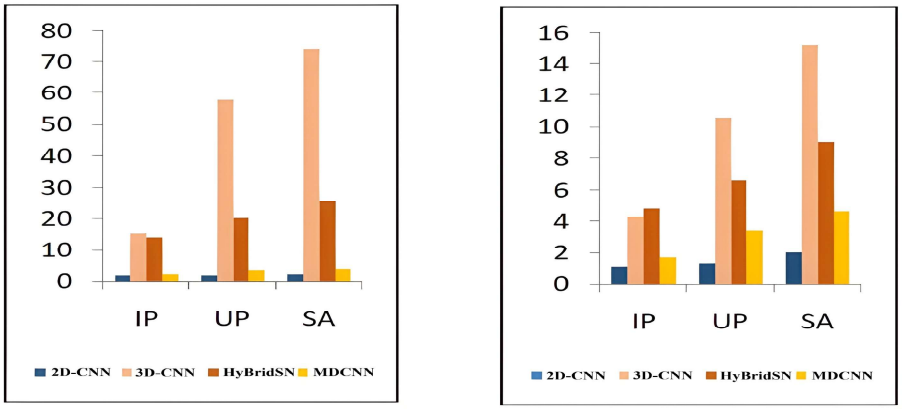


(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.7: The classification outcome Map for SA data : (a) SVM (b) 2D-CNN (c) 3D-CNN (d) SSRN (e) R-3D-CNN (f) LBP-DC-CNN (g) HybridSN and (h) MDCNN.

SA Dataset: Table 3.7 shows that MDCNN achieves 100% accuracy across all SA dataset classes. In contrast, SVM struggles with complex land cover types, with OA

values of 85.25% for "Grapes-Untrained" and 72.71% for "Vineyard-Untrained." Traditional CNN models, such as 2D-CNN and 3D-CNN, also show lower accuracy, particularly for these challenging classes. Among DL models, MDCNN outperforms SSRN, R3D-CNN, and LBP-DC-CNN by leveraging hierarchical 3D-2D dilated CNN layers with morphological descriptors. HybridSN achieves comparable accuracy, but MDCNN enhances spatial feature representation. Figure 3.7 illustrates the classification maps, highlighting spatial consistency of MDCNN and reduced misclassification. These results confirm the advantage of integrating spectral-spatial and morphological features for HSI classification, as reflected in OA, AA, and κ. As shown in Fig. 3.7, Vineyard-untrained (, C15) shows heavy misclassification in SVM and 3D-CNN, while Fallow (, C04; , C05) often suffers boundary noise in earlier models. Our MDCNN achieves cleaner separability, reducing overlaps in spectrally similar crops and producing the most reliable classification map.



(a) (b)

Figure 3.8: Execution time of different models: (a) Training time (Tr) in minutes and (b) Testing time (Te) in seconds.

Computational Efficiency: Figure 3.8 illustrates the computational efficiency of MDCNN in both training and testing compared to 3D-CNN and HybridSN. MDCNN completes training in 2.15 minutes for IP, 3.54 minutes for PU, and 3.81 minutes for SA. In contrast, 3D-CNN requires 15.2, 58, and 74 minutes, respectively. Similarly, MDCNN achieves faster testing times of 1.7s (IP), 3.4s (PU), and 4.6s (SA), outperforming both 3D-CNN and HybridSN. This efficiency results from hybrid 3D-2D convolutional design of our proposed model. It also balances model complexity while reducing computational cost. Unlike 3D-CNN, which relies entirely on computationally expensive 3D convolutions, MDCNN integrates 2D convolutions to enhance efficiency without

compromising feature extraction. Additionally, MDCNN employs dilated 3D and dilated 2D convolution layers, expanding the receptive field while minimizing trainable parameters. This design reduces overfitting and optimizes computational performance. In general observation, MDCNN achieves slightly better accuracy than HybridSN while significantly reducing training and testing time.

3.6 Chapter Summary

This chapter presented the development and evaluation of the proposed MDCNN, a novel HSI classification model that integrates spectral, spatial, and morphological features to enhance accuracy. It addressed key challenges in HSI classification, including high dimensionality, feature extraction complexity, computational cost, and overfitting. To overcome these, MDCNN incorporates morphological operations for spatial feature extraction and a hybrid CNN architecture that combines 3D and 2D convolutions. Furthermore, dilated convolutional layers expand the receptive field while minimizing parameters, improving efficiency.

Extensive experiments on three benchmark datasets (IP, PU, and SA) confirmed the superiority of MDCNN over SOTA methods. On the IP dataset, MDCNN achieved the highest OA of **99.80%**, AA of **99.87%**, and κ of **99.77**, outperforming HybridSN (99.75% OA) and R3D-CNN (99.50% OA). On the PU dataset, MDCNN obtained an OA of **99.99%**, AA of **99.97%**, and κ of **99.99**, surpassing HybridSN (99.98% OA) and far exceeding SVM (94.34% OA). On the SA dataset, MDCNN reached **100% OA, AA, and** κ, achieving perfect classification, while 2D-CNN and 3D-CNN recorded only 97.38% and 93.96% OA, respectively.

In terms of efficiency, MDCNN reduced training time to **2.15 min (IP)**, **3.54 min (PU)**, and **3.81 min (SA)**, compared to 3D-CNN's 15.2, 58, and 74 minutes. Testing time was also shorter at **1.7s (IP)**, **3.4s (PU)**, and **4.6s (SA)**, confirming MDCNN's ability to balance accuracy and cost.

Overall, MDCNN achieves SOTA accuracy with significantly reduced training and testing times by combining 3D and 2D convolutions with morphological feature extraction. A limitation is its reliance on local spatial features, restricting global context capture. Future research will explore transformer-based architectures (e.g., ViT) for global modeling and optimization strategies like pruning, quantization, and knowledge distillation to build lightweight, real-time HSI classifiers.